# SAS® Human Capital Management 5.2

## Administrator's Guide

**Second Edition**

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Human Capital Management 5.2: Administrator's Guide, Second Edition*. Cary, NC: SAS Institute Inc.

**SAS® Human Capital Management 5.2: Administrator's Guide, Second Edition**

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, February 2012

# Contents

*About This Book* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *ix*
*What's New in SAS Human Capital Management 5.2* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *xi*
*Accessibility* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *xiii*
*Recommended Reading* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *xv*

PART 1 The Administration Application 1

***Chapter 1 • About SAS Human Capital Management Administration* . . . . . . . . . . . . . . . . . . . . . . . 3**
What Is SAS Human Capital Management? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
SAS Human Capital Management Administration . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

***Chapter 2 • Managing the Data Sources* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5**
Opening the Administration Application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
Refreshing the Cache . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
Default Folder Locations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
Working with Data Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
Working with Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
Working with Cubes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Working with Information Maps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
Working with HR Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
Working with Formats . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

***Chapter 3 • Customizing the Display* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45**
Customizing an Employee Profile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
Customizing Geographic Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
Customizing Organization Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
Working with Templates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
Select the General Search Default Columns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

***Chapter 4 • Securing Objects and Tables* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65**
Security in SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
Importing Users . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
Securing Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
Searching for Identities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
Securing Table Rows . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
Hierarchical Filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
Securing Table Columns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83

***Chapter 5 • Configuring SAS Human Capital Management* . . . . . . . . . . . . . . . . . . . . . . . . . . 87**
The Diagnostic Utility . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
About the Configuration Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
System Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
Application Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
Custom Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
The SAS_DEFAULT_PROPERTIES Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99

***Chapter 6 • Customizing the Employee Profile Templates* . . . . . . . . . . . . . . . . . . . . . . . . . . 101**

Overview . . . . . . . . . . 101
A Look at the Available Templates . . . . . . . . . . 102
Customizing a Template . . . . . . . . . . 104
Making a Template Available . . . . . . . . . . 112

**Chapter 7 • Forecasting in SAS Human Capital Management . . . . . . . . . . 115**
About Forecasting in SAS Human Capital Management . . . . . . . . . . 115
Preparing the Data . . . . . . . . . . 117
Create a Stored Process Definition . . . . . . . . . . 120
Customize HCM Forecasting . . . . . . . . . . 124

**Chapter 8 • Retention Analysis . . . . . . . . . . 129**
Introduction to Retention Analysis . . . . . . . . . . 129
Extracting the Data . . . . . . . . . . 131
Generating the Scoring Table . . . . . . . . . . 138
Working with the Results . . . . . . . . . . 146

**Chapter 9 • SAS for Workforce Planning & Budgeting . . . . . . . . . . 149**
About SAS for Workforce Planning & Budgeting . . . . . . . . . . 149
Administering SAS for Workforce Planning & Budgeting . . . . . . . . . . 150
Creating Planning Measures . . . . . . . . . . 150
Creating a Form Set with Supplemental Schedules . . . . . . . . . . 156
Managing Plans . . . . . . . . . . 158

PART 2 Data Administration 159

**Chapter 10 • Introduction to Data Administration . . . . . . . . . . 161**
The Mission of Data Administration . . . . . . . . . . 161
New Data Administration Features in SAS Solutions Services . . . . . . . . . . 162
Survey of Tasks . . . . . . . . . . 164
Server Configuration . . . . . . . . . . 166
Documentation Conventions . . . . . . . . . . 166
Related Documentation . . . . . . . . . . 167

**Chapter 11 • Setting Up the SAS Data Integration Studio Environment . . . . . . . . . . 169**
Overview of Setup Tasks . . . . . . . . . . 169
Access Settings for the Data-Tier Server . . . . . . . . . . 169
Copy Jobs . . . . . . . . . . 170
Send Notifications from Jobs . . . . . . . . . . 170

**Chapter 12 • Using SAS Data Integration Studio to Supply Data to Solutions . . . . . . . . . . 171**
Overview of the Main Data Pathway . . . . . . . . . . 171
Data Encodings . . . . . . . . . . 172
Moving Data from Its Source to the Staging Tables . . . . . . . . . . 173
Loading the Detail Data Store Tables from the Staging Tables . . . . . . . . . . 174
Moving Data from the Detail Data Store to the SDM . . . . . . . . . . 177
Moving Data from the Detail Data Store to the HCM Data Mart . . . . . . . . . . 178
Overview of Other Data Pathways . . . . . . . . . . 178
Extending the Detail Data Store . . . . . . . . . . 178

**Chapter 13 • Loading Language Codes and Data Locale Codes . . . . . . . . . . 181**
Overview of Languages and Data Locales . . . . . . . . . . 181
Loading the Staging Table for Language and Locale Data . . . . . . . . . . 181
Data Locales with Predefined Text . . . . . . . . . . 182

Loading the Detail Data Store Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183
Loading Data Locale Codes into the SDM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .183

**Chapter 14 • Loading Users and User Groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185**
Overview of Users and Groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .185
Ways to Load User and Group Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .185

**Chapter 15 • Creating a Dimension . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .187**
Dimension Types, Dimensions, Hierarchies, and Members . . . . . . . . . . . . . . . . . . . . .187
Ways to Create a Dimension . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
Using the Create Dimension Transformation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .188
Starting from a Staging Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .189

**Chapter 16 • Loading Members and Hierarchies into a Dimension . . . . . . . . . . . . . . . . . . . . . .193**
Ways to Modify the Content of a Dimension . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .193
Moving Member and Hierarchy Data from Its Source to the Staging Tables . . . . . . . 194
Loading Members and Hierarchies from the Staging Tables to the Detail Data Store 201
Moving Member and Hierarchy Data from the Detail Data Store to the SDM . . . . . . 202

**Chapter 17 • Registering Member Properties So That They Are Loaded into the SDM . . . . . .205**
Overview of Member Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
Member Properties That Are Preregistered . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
Defining New Member Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
Registering Member Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
Using Member Properties That You Have Registered . . . . . . . . . . . . . . . . . . . . . . . . . .208

**Chapter 18 • Exporting and Promoting Members and Hierarchies . . . . . . . . . . . . . . . . . . . . .209**
Overview of Exporting Members and Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . .209
Using a Job to Export Members and Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . . .210
Using the Export Dimension Wizard to Export Members and Hierarchies . . . . . . . . .212
Details of the Result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
Possible Obstacles to Exporting a Dimension . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 213

**Chapter 19 • Adding a Dimension Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .215**
Overview of Adding a Dimension Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .215
Run the Job That Creates a New Dimension Type in the Staging Tables . . . . . . . . . . 216
Write Jobs to Load the New Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
Create the Jobs That Load the New Detail Data Store Tables . . . . . . . . . . . . . . . . . . .220
Customize the Job That Loads the Detail Data Store Primary Member Table . . . . . . 221
Customize the Job That Loads the Detail Data Store Hierarchy Identification Table . 227
Customize the Job That Loads the Detail Data Store Hierarchy Structure Table . . . . 236
Customize the Job That Loads the Detail Data Store Secondary Member Table . . . . 247
Loading New Dimension Types into the SDM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251
Creating Dimensions in a New Dimension Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .251
Loading Members and Hierarchies into a Dimension That
Belongs to a New Dimension Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251

**Chapter 20 • Loading Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253**
Overview: Measures and Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .253
Moving Measure Data from Its Source to the Detail Data Store . . . . . . . . . . . . . . . . . . 253
Moving Measure Data from the Detail Data Store to the SDM . . . . . . . . . . . . . . . . . . .255

**Chapter 21 • Loading Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257**
Overview: Measures and Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .257
Data Pathways for Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257
Preparing a Source Table of Metric Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .258

Preparing Jobs to Load Metric Data . . . . . . . . . . 259

**Chapter 22 • Creating a Stored Process from a SAS Data Integration Studio Job . . . . . . . . . 263**
Creating a Stored Process from a SAS Data Integration Studio Job . . . . . . . . . . 263

**Chapter 23 • Loading Data for SAS Human Capital Management . . . . . . . . . . 265**
Overview of Data Needs for SAS Human Capital Management . . . . . . . . . . 266
The Locale for SAS Human Capital Management . . . . . . . . . . 266
Competency Tables . . . . . . . . . . 266
Moving SAS Human Capital Management Data from Its Source to the Staging Tables . . . . . . . . . . 267
Loading HCM Data from the Staging Tables to the Detail Data Store . . . . . . . . . . 271
Moving Data from the Detail Data Store to the SDM . . . . . . . . . . 273
Auxiliary Files and Information Sources . . . . . . . . . . 274
Moving Data from the Detail Data Store to the HCM Data Mart . . . . . . . . . . 279
Loading HCM Metrics into a Metric Table . . . . . . . . . . 291
Creating the HCM Information Maps . . . . . . . . . . 293

**Chapter 24 • Modifying the Data Model for SAS Human Capital Management . . . . . . . . . . 295**
Overview of Modifying the HCM Data Model . . . . . . . . . . 295
Adding a Column . . . . . . . . . . 295
Changing the Character Length of a Column . . . . . . . . . . 298
Adding a Table . . . . . . . . . . 299
Adding a Cube . . . . . . . . . . 300

**Chapter 25 • Macro Variables in the PREBUILD.SAS Macro File . . . . . . . . . . 303**
Overview of SAS Human Capital Management Macro Variables . . . . . . . . . . 304
Families of Table-Acronym Macro Variables . . . . . . . . . . 304
Individual Macro Variables . . . . . . . . . . 313

**Chapter 26 • Internal Formats . . . . . . . . . . 329**
Introduction . . . . . . . . . . 329
IACTION . . . . . . . . . . 329
ICHURN . . . . . . . . . . 330
IEEOCL . . . . . . . . . . 330
IEMPSTA . . . . . . . . . . 331
IEXEMPT . . . . . . . . . . 331
IONPAYRL . . . . . . . . . . 331
IPAYPER . . . . . . . . . . 332
IREGTMP . . . . . . . . . . 332
ISTECLS . . . . . . . . . . 332
ITERM . . . . . . . . . . 333

PART 3 System Administration 335

**Chapter 27 • Introduction to System Administration . . . . . . . . . . 337**
Overview of SAS Solutions Services . . . . . . . . . . 337
What's Changed in SAS Solutions Services . . . . . . . . . . 338
Required Skills . . . . . . . . . . 338
Documentation Conventions . . . . . . . . . . 339
Additional Documentation . . . . . . . . . . 339

**Chapter 28 • Post-Configuration Steps . . . . . . . . . . 341**
Overview . . . . . . . . . . 342

General Modifications . . . . . 343
SAS Human Capital Management Modifications . . . . . 346
KPI Viewer Modifications . . . . . 350
Secure Your Installation . . . . . 350
(Optional) Verify Using Sample Data . . . . . 353
Create the Site's Users and Groups . . . . . 358
Load Production Data . . . . . 358
Localization . . . . . 359
Add-On Configuration . . . . . 361

***Chapter 29 • Assigning Groups and Roles*** . . . . . **363**
Overview of Users, Groups, and Roles . . . . . 363
SAS Solutions Services Users, Groups, and Roles . . . . . 364
SAS Human Capital Management Groups and Roles . . . . . 367
Registering Users . . . . . 370

***Chapter 30 • Portal and Content Administration*** . . . . . **373**
Portal Administration . . . . . 373

***Chapter 31 • Administering the Middle Tier*** . . . . . **377**
Overview . . . . . 377
Oracle WebLogic Server Modifications . . . . . 378
IBM WebSphere Server Modifications . . . . . 379

***Chapter 32 • Viewing and Configuring the Log Files*** . . . . . **381**
Log File Locations and Configuration Files . . . . . 381
Dynamically Configuring Logging Levels . . . . . 382
Additional Log Files . . . . . 382

***Chapter 33 • Administering MySQL Server*** . . . . . **383**
MySQL Overview . . . . . 383
Windows: MySQL Installation and Configuration . . . . . 383
UNIX: MySQL Installation and Configuration . . . . . 384

PART 4 The Data Model 385

***Chapter 34 • Introduction to the Data Model*** . . . . . **387**
Introduction . . . . . 387

***Chapter 35 • Descriptions of DDS Tables*** . . . . . **389**
Detail Data Store Table Descriptions . . . . . 389

***Chapter 36 • Descriptions of DDS Table Columns*** . . . . . **401**
Descriptions of Detail Data Store Table Columns . . . . . 401

***Chapter 37 • More Information about DDS Tables*** . . . . . **509**
More Information about Detail Data Store Tables . . . . . 509

***Chapter 38 • Solutions Data Mart Jobs*** . . . . . **571**
Solutions Data Mart Jobs . . . . . 571

***Chapter 39 • HCM Data Mart Jobs*** . . . . . **573**
HCM Data Mart Jobs . . . . . 573

***Chapter 40 • HCM Data Mart Tables*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **591**
HCM Data Mart Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 591

***Chapter 41 • Descriptions of HCM Data Mart Table Columns*** . . . . . . . . . . . . . . . . . . . . . . . . . **597**
Descriptions of HCM Data Mart Table Columns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 597

PART 5 Reference Material 671

***Appendix 1 • Object Security: List of Objects*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **673**
About Object Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 673
Employee Profile Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 673
Geographical Analysis Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 674
Organization Analysis Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 675
General Search Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 675
Administrator Options: Data Tab . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 676
Administrator Objects: Customize Tab . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 678
Administrator Objects: Security Tab . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 679
Administrator Objects: Configuration Tab . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 681
Home Page Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 681
Custom Objects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 682

***Appendix 2 • Metrics in SAS Human Capital Management*** . . . . . . . . . . . . . . . . . . . . . . . . . . . **683**
Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 683
Organizational Effectiveness . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 684
Human Resources Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 686
Compensation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 687
Benefits . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 689
Separations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 691
Staffing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 695
Training and Development . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 701

***Appendix 3 • The Public API*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **705**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 706
Hierarchy for all Packages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 707
API Index . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 708
Packages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 713
Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 715
Classes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 716
Interface PublicAPIInterface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 727
Serialized Form . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 734

***Appendix 4 • MySQL Reserved Words*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **735**

***Appendix 5 • Configuring the SAS Environment Files*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **737**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 737
Deploying the SAS Environment Files to an HTTP Server . . . . . . . . . . . . . . . . . . . . . . . . . 738
Defining Additional SAS Environments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 739
Defining Additional Solutions Environments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 739

***Appendix 6 • Default Port Usage*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **743**

***Index*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **747**

# About This Book

## Audience

This book helps you use SAS Human Capital Management as an administrator. For an overview of SAS Human Capital Management, see "What Is SAS Human Capital Management?" on page 3.

For related information that you might find useful in administering SAS Human Capital Management, see Recommended Reading and the administration guides for the SAS Intelligence Platform, available at **support.sas.com/93administration**.

# What's New in SAS Human Capital Management 5.2

## Overview

SAS Human Capital Management 5.2 includes new features to help you analyze your workforce, measure performance, and produce reports based on regulatory or organizational needs.

## New and Enhanced Features

The following features are new or enhanced in SAS Human Capital Management 5.2:

- enhanced employee profiles, including multiple profiles, profile templates, external actions, and search capabilities
- enhanced geographic analysis, with better performance and filtering support
- built-in content management, including the ability to manage permissions and register for alerts
- tighter integration with SAS BI Dashboard, including a new provider for SAS Human Capital Management metrics
- tighter integration with SAS Strategy Management, SAS Web Report Studio, and the SAS Information Delivery Portal
- built-in support for single-variable forecasting
- support for workforce planning and budgeting (with SAS Financial Management)
- support for retention analysis using predictive analytics
- an enhanced Administration application, including the following new or enhanced features:
    - consolidated security administration (object-level security, row-level security, and column-level security)
    - employee profile management, which includes the ability to create and assign custom profiles
    - the ability to create cubes and information maps
    - the ability to create and manage measures
    - configuration management
- a public API for customizing employee profile templates

- a diagnostic tool for SAS Human Capital Management, with these features:
  - reports on critical configuration elements, SAS server connections, database connections, and applications that are running on the managed servers
  - accessibility from the Administration application or from the command line

## Changes to Stored Processes

The following features have changed in SAS Human Capital Management:

- Stored processes cannot be executed within a Microsoft Office application. They can be executed from the workspace, from a shortcut link, from SAS Web Report Studio, or from the action menu in the Employee Browser or a geographic analysis.

# Accessibility

## Introduction

SAS Human Capital Management includes the following accessibility and compatibility features that improve usability of the product for users with disabilities. These features are related to accessibility standards for electronic information technology that were adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended.

If you have questions or concerns about the accessibility of SAS products, send an e-mail message to **`accessibility@sas.com`**

## Keyboard Navigation

### *Standard Keyboard Navigation*

SAS Human Capital Management can be navigated by using the keyboard. The following table includes some guidelines:

| Task | Keyboard Control |
|---|---|
| Move forward through controls | TAB |
| Move backward through controls | SHIFT+TAB |
| Display the contents of a drop-down list | ALT+down arrow |
| Display a menu that has focus | Down arrow |
| Activate a button or menu selection when it has focus | ENTER |
| Open a context menu | SHIFT+F1 |

### *Shortcut Keys Added for Faster Navigation*

SAS Human Capital Management enables the following keyboard shortcuts for faster navigation for some wizards, such as the New Cube wizard:

| Task | Keyboard Control |
|---|---|
| Select **Next** | ALT+right arrow |
| Select **Previous** | ALT+left arrow |
| Select **Finish** | ALT+up arrow |
| Select **Cancel** | ALT+down arrow |

## Using SAS Human Capital Management with JAWS

### *Tables Used for Screen Layout*

When tables are used for screen layout in some dialog boxes, JAWS reads the table dimensional information. To ensure that all fields are read, set your JAWS reader to read one row at a time and use the standard JAWS keystrokes for reading a table.

### *The Edit Field*

When using the keyboard to navigate to the Edit field, JAWS repeats the last valid item that it was able to read before a header or a footer.

## High Contrast and Custom Color Styles

There is intermittent support for high contrast and custom color styles in SAS Human Capital Management.

# Recommended Reading

Here is the recommended reading list for this title:

- *SAS Human Capital Management: User's Guide*
- *SAS Solutions Services: Customization Guide*

For a complete list of SAS publications, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

# Part 1

# The Administration Application

*Chapter 1*
***About SAS Human Capital Management Administration*** . . . . . . . . . . . *3*

*Chapter 2*
***Managing the Data Sources*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *5*

*Chapter 3*
***Customizing the Display*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *45*

*Chapter 4*
***Securing Objects and Tables*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *65*

*Chapter 5*
***Configuring SAS Human Capital Management*** . . . . . . . . . . . . . . . . . . . . *87*

*Chapter 6*
***Customizing the Employee Profile Templates*** . . . . . . . . . . . . . . . . . . *101*

*Chapter 7*
***Forecasting in SAS Human Capital Management*** . . . . . . . . . . . . . . . . *115*

*Chapter 8*
***Retention Analysis*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *129*

*Chapter 9*
***SAS for Workforce Planning & Budgeting*** . . . . . . . . . . . . . . . . . . . . . . *149*

# *Chapter 1* About SAS Human Capital Management Administration

**What Is SAS Human Capital Management?** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**SAS Human Capital Management Administration** . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

## What Is SAS Human Capital Management?

SAS Human Capital Management software integrates workforce data into a single source of information, enabling you to analyze your workforce, measure performance, and produce reports based on regulatory or organizational needs. Using SAS Human Capital Management provides the following benefits:

- **You can analyze the workforce and simulate organizational changes.** Predefined analytics such as organization analysis display organizational hierarchies in a list of graphical views while surfacing critical data about the work force. You can also simulate organizational structure changes for workforce planning and modeling by moving groups or people to see the effect.
- **You can measure and improve workforce productivity.** For the analysis of key indicators, SAS Human Capital Management provides an extensive set of prepackaged metrics as well as the ability for users to create their own measures. Combined with the SAS BI Dashboard (part of the SAS Intelligence Platform), SAS Human Capital Management enables you to view the status of key metrics, such as revenue per employee, relative to goals.
- **You can minimize risk by changing likely outcomes.** SAS Human Capital Management provides advanced analytics and easy-to-use interfaces so business users can identify and minimize risks by predicting workforce changes and analyzing associated costs. Forecasting or analytic expertise is not needed.
- **You can budget for future workforce needs.** When integrated with SAS Financial Management, SAS Human Capital Management makes detailed employee information available for planning and budgeting within a structured workflow.

## SAS Human Capital Management Administration

Users with the HCM Administrator role can log on to SAS Human Capital Management and perform these tasks:

- administer data sources
  - import, view, copy, and export tables
  - modify table and column attributes
  - add hierarchy mappings
  - create and rebuild cubes and information maps
  - create and manage measures for metric analysis
  - create and manage planning measures
  - create and edit formats
- customize the user interface
  - create and customize employee profiles
  - set default values for geographic analysis
  - set default values for organization analysis
  - set default values for the general search
- manage security
  - manage object security, which determines the actions that users can perform
  - manage table security, by creating and assigning row-level filters
  - assign column permissions
- manage the SAS Human Capital Management configuration, and view and edit configuration properties

In addition, administrators and consultants can perform these tasks:

- generate single-variable forecasting data
  - prepare and run ETL jobs that generate the forecasting data
  - define stored processes to display the forecasting results
- prepare data for retention analysis
  - prepare the data tables and properties files
  - run the code that generates the scoring table
  - create reports for viewing the results
- use SAS for Workforce Planning & Budgeting to create planning measures (in the Administration application), and create form sets that managers can use for entering budget data (with SAS Financial Management)

# *Chapter 2*
# Managing the Data Sources

**Opening the Administration Application** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **6**
Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
**Refreshing the Cache** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **7**
**Default Folder Locations** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **7**
**Working with Data Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **7**
About Data Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
About Importing and Exporting Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
Add a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
Perform a Custom Import . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
Modify Table Attributes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
Modify Column Attributes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
Modify the hcmtitles and hcmlabels Properties Files . . . . . . . . . . . . . . . . . . . . . . . . 16
Copy a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
View a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
Create an Information Map from a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
Export a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
Delete a Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
Generate Search Symbols . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
**Working with Hierarchies** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **22**
What Is a Hierarchy? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
What Is a Hierarchy Mapping? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
View Available Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
Add a Hierarchy Mapping . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
View Hierarchy Mappings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
**Working with Cubes** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **25**
Using Cubes in SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . . . . 25
Create a Cube . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Create a New Dimension for a Cube . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
Managing Cubes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
Refresh or Rebuild Multiple Cubes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
**Working with Information Maps** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **33**
**Working with HR Measures** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **33**
About HR Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
Create an HR Measure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
Manage HR Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
**Working with Formats** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **35**
About Display Formats . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

Add a Format . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
Assign a Format to a Column . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
Manage Formats . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
Predefined Formats . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

# Opening the Administration Application

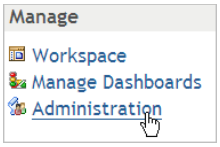
## Introduction

In the SAS Human Capital Management Administration application, you can manage and customize SAS Human Capital Management, including data sources, employee profiles, application default values, security, and configuration settings.

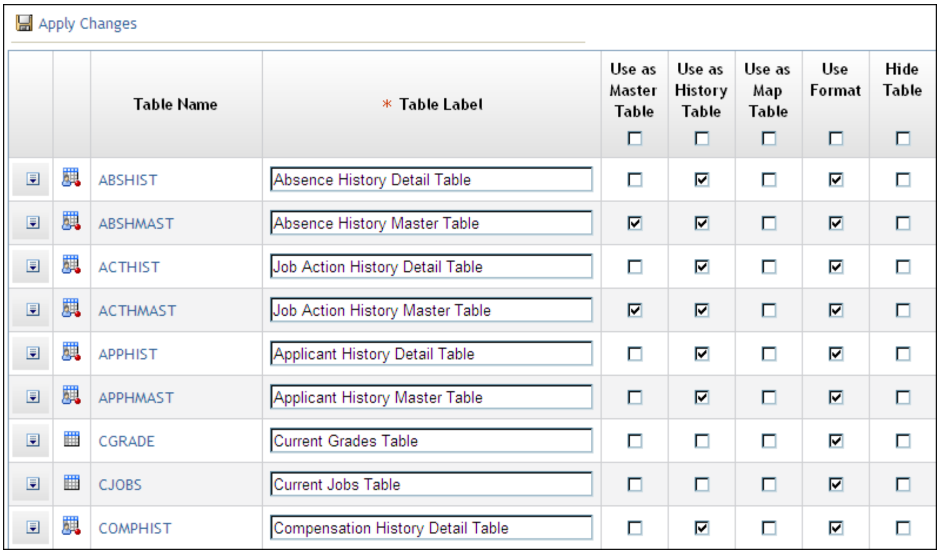
To administer SAS Human Capital Management:

1. Log on to SAS Human Capital Management as a user with the HCM Administrator role.

2. Click the **Administration** task:



The Administration application appears.



| | | Table Name | * Table Label | Use as Master Table<br>☐ | Use as History Table<br>☐ | Use as Map Table<br>☐ | Use Format<br>☐ | Hide Table<br>☐ |
|---|---|---|---|---|---|---|---|---|
| | | ABSHIST | Absence History Detail Table | ☐ | ☑ | ☐ | ☑ | ☐ |
| | | ABSHMAST | Absence History Master Table | ☑ | ☑ | ☐ | ☑ | ☐ |
| | | ACTHIST | Job Action History Detail Table | ☐ | ☑ | ☐ | ☑ | ☐ |
| | | ACTHMAST | Job Action History Master Table | ☑ | ☑ | ☐ | ☑ | ☐ |
| | | APPHIST | Applicant History Detail Table | ☐ | ☑ | ☐ | ☑ | ☐ |
| | | APPHMAST | Applicant History Master Table | ☐ | ☑ | ☐ | ☑ | ☐ |
| | | CGRADE | Current Grades Table | ☐ | ☐ | ☐ | ☑ | ☐ |
| | | CJOBS | Current Jobs Table | ☐ | ☐ | ☐ | ☑ | ☐ |
| | | COMPHIST | Compensation History Detail Table | ☐ | ☑ | ☐ | ☑ | ☐ |

*Note:* If you are using Internet Explorer 8 with SAS Human Capital Management, you need to select compatibility mode. You can enable compatibility mode in Internet Explorer 8 by selecting **Tools** ⇨ **Compatibility View** in your Internet Explorer browser.

3. Select an area to modify:

   - On the **Data** tab, you can manage tables, columns, hierarchies and hierarchy mappings, cubes, information maps, measures, and formats.

   - On the **Customize** tab, you can customize SAS Human Capital Management by creating employee profiles, assigning and customizing templates, and setting default properties for geographic analysis, organization analysis, and the general search.

- On the **Security** tab, you can control security for objects, tables, and columns.
- The **Configuration** tab displays information about the current configuration of SAS Human Capital Management. You can set some of the configuration properties. Other properties are Read-Only.

## Refreshing the Cache

After you make changes in the Administration application, SAS Human Capital Management often continues to use cached values rather than the new values that you supply. Using cached values makes the application run faster; it also prevents data or the user interface from changing while someone is using the application.

When you restart the Web application server, the cache is refreshed and your changes are available.

To make changes available without restarting the Web application server, make your changes for a session and then click **Refresh Cache** in the toolbar of the Administration application. (Do not refresh the cache after every change.) An e-mail message is sent to the administrator confirming that the cache has been refreshed.

## Default Folder Locations

SAS Human Capital Management uses the following default folder locations:

- information maps: **SAS Folders** ⇨ **Products** ⇨ **SAS Human Capital Management** ⇨ **Data Sources** ⇨ **Information Maps**
- cubes: **SAS Folders** ⇨ **Products** ⇨ **SAS Human Capital Management** ⇨ **Data Sources** ⇨ **Cubes**
- stored processes: **SAS Folders** ⇨ **Products** ⇨ **SAS Human Capital Management** ⇨ **Reports**
- ETL jobs for imported tables: **SAS Folders** ⇨ **Products** ⇨ **SAS Human Capital Management**

## Working with Data Tables

### *About Data Tables*

#### *Types of Tables*

On the **Data** tab of the HCM Administration application, you can add a table or modify its attributes, such as the table description or its use as a master table, a history table, or a map table. You can add a column to a table or modify column attributes, such as the column label or format.

The HCM database contains these types of tables:

- Detail tables contain specific data, such as absence histories, open positions, or current grades.
- Master tables have been optimized for query and reporting. They contain data from one or more detail tables and can also contain columns with calculated values.
- Summary tables contain summarized data from the master tables, with less historical detail.
- Cubes contain data from the master and summary tables that has been reorganized into a multidimensional structure. In SAS Web Report Studio, an HCM cube can be viewed by means of an information map.
- Map tables contain data for image maps that can be used in a geographic analysis.

Both detail tables and master tables can also be history tables. A history table can contain multiple records per employee. In the standard HCM tables, the _LASTREC column identifies the current record for each employee. When a user views data from one of the fixed categories in the employee browser, the records are filtered to return only the current record for an employee.

For information about the jobs that load the standard HCM tables, see "Introduction to Data Administration" on page 161.

### Master Tables

A master table contains fields that identify the hierarchical structure of the data, such as levels of management or division and department levels.

The following master tables are provided on installation of SAS Human Capital Management. If necessary, you can add other master tables.

***Table 2.1*** *Master Tables in SAS Human Capital Management*

| Table | Description |
|---|---|
| Employee Master (EMPMAST) | Contains the most current information for each employee, including job action history, position history, jobs, pay grades, workgroup, employee general, and (optionally) open positions.<br>Sites can decide to include only active employees. By default, all employees are included. Including only active employees would affect any tables using EMPMAST for terminated employees. |
| Job Action History Master (ACTHMAST) | Contains all employee actions, denormalized by including workgroup, jobs, pay grade, position history, current employee. |
| Absence History Master (ABSHMAST) | Contains all employee absence records, denormalized by including workgroup, jobs, pay grade, position history, and current employee. |
| Applicant History Master (APPHMAST) | Contains all applicant tracked records, denormalized by including workgroup, jobs, pay grade, position history, current employee. |
| Open Position Master (OPOSMAST) | Contains all open position records, denormalized by including workgroup, position history, and jobs. |
| Termination Master (TERMMAST) | Contains all termination records from the Job Action History Master Table. |

| Table | Description |
|---|---|
| Time in Position (TIP) | Contains records of the date an employee's position changed and calculates the time spent in a position. |

## Summary Tables

The following summary tables are provided by default.

*Table 2.2 Summary Tables in SAS Human Capital Management*

| Summary Table | Description |
|---|---|
| Salary History (SALHIST) | Contains all salary change information from the Job Action History Master table. |
| Salary History Summary (SALHSUM) | Contains the total compensation history based on salary changes. Uses the Compensation, Job Action History, Position, Jobs, Workgroup, Grades, and Current Employee General tables. |
| Open Positions Summary (OPOSSUM) | Summarizes the open position information using the Open Position History, Position History, Workgroup, and Jobs tables. |
| Headcount Summary (HEADSUM) | Summarizes the headcount over time. Time is defined by month or year, beginning or end based on settings in the %PREBUILD macro. Information uses data from the Job Action History, Position History, Jobs, Workgroup, Pay Grade, and Current Employee General tables. |
| Churn (CHURN) | Records internal movement. The %PREBUILD macro contains criteria for identifying the characteristics to define for CHURN. The CHURN table is based on the Job Action History Master table. |

## Map Tables

Map tables are used in geographic analysis. The following map tables are provided on installation of SAS Human Capital Management:

| Table | Description |
|---|---|
| HRV_GBL | World map |
| HRV_US | United States map |
| HRV_STATE | State maps |

## About Importing and Exporting Tables

### Location of Table Data for Import

If the file being imported does not reside on the data tier, you must specify a universal naming convention (UNC) path to the file. You must also have a network login and Read access to the file.

### Effect on Data Types

Be aware that CSV files do not attach data types to columns, and exporting or importing a table as a CSV file can result in some columns being given the wrong data type (for example, character data that consists solely of numbers might be treated as numeric).

We recommend that you export and import tables in Microsoft Excel workbook format rather than as CSV files. For greater control over tables and columns, use an ETL process.

### Exporting Tables with Formatted Columns

When you export a table, you must decide how you want to handle formatted columns. For example, by default the JOB_GROUP_CD column has a user-defined display format associated with it. In the sample data, a code of **2S** is associated with a formatted value of **Sales and Marketing**. When you export the table, do you want to export the code (**2S**) or the formatted value (**Sales and Marketing**)?

If a column is formatted, and **Use Format** is enabled for the table and column, then the display format is exported, rather than the code for that column.

To export a column's unformatted value, follow these steps:

1. Temporarily disable formatting for the table:
   a. On the **Data** tab, select **Tables** from the navigation tree.
   b. Clear the **Use Format** check box for the table.
   c. Click **Apply Changes**.
   d. Click **Refresh Cache**.

   See “Modify Table Attributes” on page 14.
2. Export the table.

   For details, see “Export a Table” on page 21 or “View a Table” on page 18.
3. Re-enable formatting for the table.
   a. Open the table attributes again (as in Step 1) and select the **Use Format** check box.
   b. Click **Apply Changes**.
   c. Click **Refresh Cache**.

For more information about table attributes, see “Export a Table” on page 21.

### Importing SAS Data Sets with Formatted Columns

If you import a SAS data set, and any columns are associated with a display format, the formatted values are imported, rather than the underlying codes. The

SAS_HCMFORMATS table and the SAS formats catalog are not updated, and the SAS_HCMMETACOLUMN table does not contain a format attribute for the column.

If you want the table to include the underlying codes instead of the formatted values, remove the formatting before you import the SAS data set. After the import, you can create the appropriate formats and assign them to the table columns.

### Warning Messages

When you export a file to Microsoft Excel, you might get a warning message that the file is in a different format than the one that is specified by the file extension. The message appears because the content is an XML stream rather than native Excel format. Click **Yes** to open the file. In Microsoft Excel, when you save the file, save it in Excel Workbook format.

## Add a Table

The **Add Table** feature makes a table available to SAS Human Capital Management. The table can be one that already exists in the HCM database, or it can be an external source such as a Microsoft Excel workbook or CSV file.

When you add a table, its information is added to the SAS_HCMMETATABLE and SAS_HCMMETACOLUMN tables, and it is available as a source table in SAS Human Capital Management. For example, the table can be mapped to a hierarchy, or it can be displayed in an employee profile.

To add a table, click the **Data** tab in the Administration application. Then follow these steps:

*Note:* For information about the **View Code** button, see "Perform a Custom Import" on page 13.

1. Click **New Table**.
2. To add a table that already exists in the HCM database:
    a. Select the **Add Existing** radio button.
    b. Select a table from the drop-down list.
3. To add a table from an external source:
    a. Select the **External** radio button.
    b. From the **Type** drop-down list, select the source type.

       For more information about source type considerations, see "About Importing and Exporting Tables" on page 10.

       *Note:* Available options can vary depending on your operating system. Importing a table using the Microsoft Excel 4 or Microsoft Excel 5 type can result in truncation of column names to 8 characters.
    c. Provide a path to the external file in the **Source** field.

       If the file does not reside on the data tier, you must specify a universal naming convention (UNC) path to the file. You must also have a network login and Read access to the file.
    d. In the **Table Name** field, type a name for this table.

       The maximum length for the name is 32 characters.
    e. To replace an existing table, select **Replace table if the same name exists**.

If you replace an existing table, the table attributes and column attributes are replaced for this table. The row-level security settings and filters are deleted. If you specify **Allow security access for HCM roles**, new settings and filters are created.

f. To create an ETL job that you can use to load this table on a regular basis, select **Create an ETL job**.

If you are replacing an existing table, checking this box replaces a previous ETL job, if one existed. For the location of the ETL jobs, see "Default Folder Locations" on page 7.

4. Provide the following information in the **General Settings** section:

**Table Label**
(Optional) Enter a descriptive label. This value appears in the user interface when users must select a table. If the label is empty, the table name is used.

The maximum length for the label is 100 characters.

**Use as a master table**
Select if this is a master table, which can contain fields from multiple detail tables as well as calculated fields.

**Use as a master history table**
Select if this is a history table, which can contain multiple records per employee.

If a table contains multiple records per employee, it must contain a _LASTREC column and it must be identified as a history table. Otherwise, results can be unexpected, including a configuration exception.

**Create a map table**
Select if this is a map table, which contains data for geographic analysis.

**Use format**
Select to specify that a column display format, if specified, should be applied to the table.

*Note:* When you first add a table, it has no column display formats. You must modify the table's column attributes. See "Modify Column Attributes" on page 14.

**Hide table**
Select to hide the table in the user interface. You might want to hide a table that provides supporting data to another, visible table. For example, you might want to hide a table that provides values for a lookup field for another table.

5. Provide the following information in the **Security Options** section:

**Allow security access for HCM roles**
Select to create filters for each of the HCM roles. As created, these filters allow full access to the table. For more information, see "About Row-Level Security" on page 74.

*Note:* To log on to SAS Human Capital Management, users must have one of the HCM roles.

**Register table in metadata repository**
Select to add the table to the HCMData library in the metadata repository.

Only registered tables are available in the user interface of SAS Human Capital Management.

If you are replacing a table that was previously registered, be sure to select this box in order to pick up any changes to the table, such as additional columns.

6. Provide the following information in the **Other Options** section:

   **Build Information Map**

   Select to create an information map from the table. All columns are included. The information map is written to the default folder location in the workspace. (See “Default Folder Locations” on page 7.)

   If you entered a label for the table, the information map name is ***label*_MAP**. Otherwise, it is ***table*_MAP**. If an information map with this name already exists, it is replaced.

7. Click **OK**.

The table is added to the SAS_HCMMETATABLE table, which contains information about table attributes, and its columns are added to the SAS_HCMMETACOLUMN table, which contains information about column attributes. For more information about the SAS_HCMMETATABLE table, see “Modify Table Attributes” on page 14, and for more information about the SAS_HCMMETACOLUMN table, see “Modify Column Attributes” on page 14.

*Note:* If you add a table to the HCM database, you must use the **Add Table** feature so that the table is registered in SAS Human Capital Management.

If the table does not appear in the list of tables, click **SAS Log** to view the log from the **Add Table** operation.

## *Perform a Custom Import*

From the Add Table dialog box, you can view, modify, and run the code that is generated for adding the table. Follow these steps:

1. Complete all the fields in the dialog box (including the table name and source, the **General Settings**, **Security Options**, and **Other Options**), so that the code reflects your selections.
2. Click **View Code**.
3. In the dialog box that appears, edit the code.
4. Click **Run** to execute the code.
5. Click **Close**.
6. Click **Cancel** to exit the Add Table dialog box.

   ***CAUTION:***

   Do not click **OK**.

7. Click **SAS Log** to view the log from the operation.

Customizing the table import code can be useful, but it can also have unwanted side effects if you are not careful. In particular:

- If you modify the generated code and remove the checks for duplicate tables, your code might add more tables with the same name.
- If you execute your custom code, close the SAS Code window, and then click **OK** instead of **Cancel** on the Add Table dialog box, the application tries to add the table again. If it succeeds, it replaces the table that you just added.
- If a custom import fails, you must manually perform any necessary cleanup tasks. For example, you should remove any related entries in the SAS_HCMMETATABLE and SAS_HCMMETACOLUMN tables. If you requested that row-level filters be

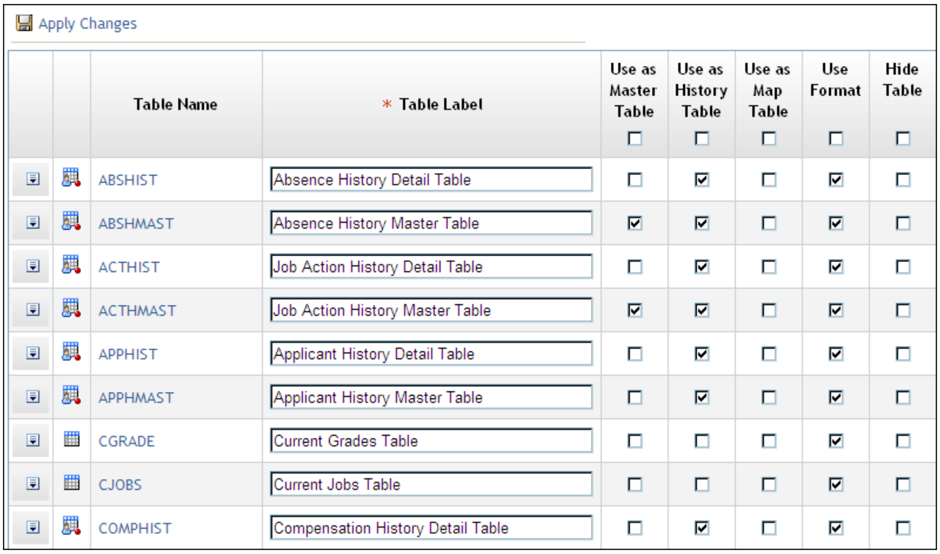
created, you might need to delete those filters. If you requested that an ETL job or an information map be generated, you might need to remove those objects. If you added tables with the same name by mistake, you will need to remove them manually.

## Modify Table Attributes

On the **Data** tab, you can modify attributes for a table, such as its description and its use as a master or a history table. These attributes are stored in the SAS_HCMMETATABLE table in the HCM database.

To modify table attributes, click the **Data** tab in the Administration application. Then follow these steps:

1. In the navigation tree at the left, select **Tables**.



| | | Table Name | * Table Label | Use as Master Table<br>□ | Use as History Table<br>□ | Use as Map Table<br>□ | Use Format<br>□ | Hide Table<br>□ |
|---|---|---|---|---|---|---|---|---|
| ▾ | ▦ | ABSHIST | Absence History Detail Table | □ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | ABSHMAST | Absence History Master Table | ☑ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | ACTHIST | Job Action History Detail Table | □ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | ACTHMAST | Job Action History Master Table | ☑ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | APPHIST | Applicant History Detail Table | □ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | APPHMAST | Applicant History Master Table | □ | ☑ | □ | ☑ | □ |
| ▾ | ▦ | CGRADE | Current Grades Table | □ | □ | □ | ☑ | □ |
| ▾ | ▦ | CJOBS | Current Jobs Table | □ | □ | □ | ☑ | □ |
| ▾ | ▦ | COMPHIST | Compensation History Detail Table | □ | ☑ | □ | ☑ | □ |

History tables are marked with the History Table symbol ▦. All other tables are marked with the Normal Table symbol ▦.

2. Modify one or more of the table attributes. For information about these attributes, see "Modify Column Attributes" on page 14.

3. Click **Apply Changes**.

On the table properties page, you can also copy a table, view table contents, build an information map from a table, export a table, delete a table, or generate symbols for all the columns in a table. These tasks are available from the action menu ▾.

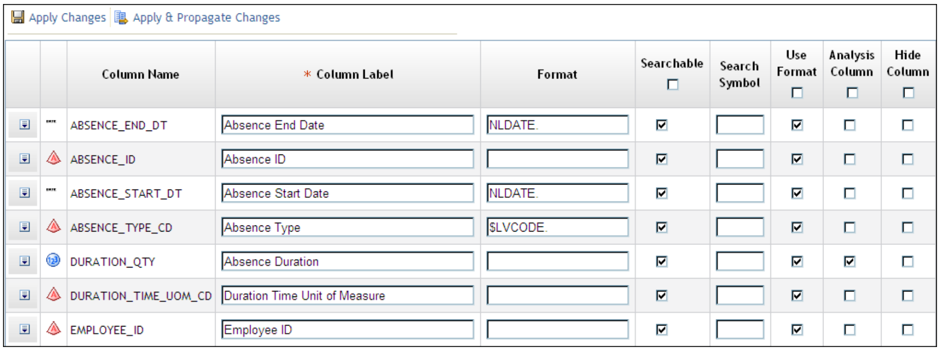
## Modify Column Attributes

In addition to modifying table attributes, you can modify column attributes for a table. These attributes are stored in the SAS_HCMMETACOLUMN table in the HCM database.

To modify column attributes, click the **Data** tab in the Administration application. Then follow these steps:

1. In the navigation tree at the left, select the table that you want to modify.

   A list of columns is displayed.



| | | Column Name | * Column Label | Format | Searchable □ | Search Symbol | Use Format □ | Analysis Column □ | Hide Column □ |
|---|---|---|---|---|---|---|---|---|---|
| ⊡ | "" | ABSENCE_END_DT | Absence End Date | NLDATE. | ☑ | | ☑ | □ | □ |
| ⊡ | △ | ABSENCE_ID | Absence ID | | ☑ | | ☑ | □ | □ |
| ⊡ | "" | ABSENCE_START_DT | Absence Start Date | NLDATE. | ☑ | | ☑ | □ | □ |
| ⊡ | △ | ABSENCE_TYPE_CD | Absence Type | $LVCODE. | ☑ | | ☑ | □ | □ |
| ⊡ | ① | DURATION_QTY | Absence Duration | | ☑ | | ☑ | ☑ | □ |
| ⊡ | △ | DURATION_TIME_UOM_CD | Duration Time Unit of Measure | | ☑ | | ☑ | □ | □ |
| ⊡ | △ | EMPLOYEE_ID | Employee ID | | ☑ | | ☑ | □ | □ |

2. Modify one or more of the following attributes:

   *Note:* Use the scroll bars at the side and foot of the page as necessary. To apply an attribute to all columns in this table, select the check box under the attribute heading. To deselect an attribute for all columns, clear the check box under the attribute heading.

| Attribute | Description |
|---|---|
| **Column Name** | Name for this column. |
| **Column Label** | A label to apply to this column in organizational analysis, geographic analysis, the employee browser, or a report.<br>Some items in the user interface (such as some statistics) are not associated with a specific table column. The labels for those items are taken from the hcmlabels.properties file. |
| **Format** | The display format for the column. To add a custom format, select **Formats** in the navigation tree on the **Data** tab.<br>In order for a table's columns to use the display format, the table's **Use Format** property must also be selected.<br>Custom date display formats are supported. If the currency display format is not appropriate for the way the data is stored in the database, you can change it in this attribute. However, currency conversions are not supported. |
| **Searchable** | Select this check box to make a column eligible for searches in the general search, the Employee Browser, or a geographic analysis. |
| **Search Symbol** | This column contains a column abbreviation that can be used in the general search. The symbol must be unique to this table (it cannot match another search symbol or another column name). It can have a maximum length of 3 alphanumeric characters. |
| **Use Format** | Select this check box to apply the display format, if one exists, to this column. |

| Attribute | Description |
|---|---|
| **Analysis Column** | Select this check box to make the column available for analysis.<br>For a geographic analysis or a cube, this selection applies only to numeric values (data types of NUM, DATA, or CURRENCY).<br>For a geographic analysis, selecting **Analysis Column** for a character column makes it available for display in table view. |
| **Hide Column** | Select this check box to hide the column from display. |
| *Statistics* | Select a check box to include the corresponding statistic. The selection applies only if **Analysis Column** is selected and the column is numeric. The following statistics are available:<br>• **Count**: number of nonmissing values<br>• **Min**: minimum value<br>• **Max**: maximum value<br>• **Sum**: sum of values<br>• **Number Missing**: number of missing values<br>• **Uncorrected Sum Sq**: uncorrected sum of squares<br>• **Avg**: arithmetic mean or average of values<br>• **Range**: range of values |

3. To apply your changes only to this table, click **Apply Changes**.

   To apply these attribute values to columns with the same name in all HCM tables, click **Apply & Propagate Changes**. All column attributes—not just the attributes that you changed—are applied to all instances of this column.

   *Note:* Search symbols are not propagated to other tables, to avoid possible duplication. (You might have already defined the identical search symbol for a different column in another table.)

## *Modify the hcmtitles and hcmlabels Properties Files*

In addition to the labels that you can modify on the **Data** tab, SAS Human Capital Management maintains two property files: hcmlabels.properties and hcmtitles.properties.

The hcmlabels.properties file contains labels for tables, columns, statistics, metric factors, OLAP cubes, OLAP hierarchies, employee profile category tables, miscellaneous labels for reports, information maps, and the forecasting stored processes. During the installation process, these labels are used for seeding of the SAS_ tables (including the SAS_HCMMETATABLE and SAS_HCMMETACOLUMN tables). After that point, changes that you make to the table or column properties on the **Data** tab are used rather than the values in the properties files, with the following exceptions.

Column formats and labels in OLAP cubes are extracted from SAS_HCMMETACOLUMN when the cube is built or rebuilt (both the physical cube and the cube metadata). Any other labels that are related to OLAP cubes, such as hierarchy captions, dimension captions, and cube labels, are extracted from the properties files when the cube is built or rebuilt.

*Note:* The Dimension captions use the labels of the associated columns from the properties files rather than the labels from SAS_HCMMETACOLUMN. If you change column labels on the **Data** tab, and they are used in cube dimensions, you should propagate those changes to the hcmlabels.properties file.

To modify the hcmlabels.properties file:

1. Change your current directory to `!SASROOT\hrds\sasmisc` (Windows).
2. Open the appropriate version of the file for editing.

   The sasmisc folder contains localized versions of the file, in the form hcmlabels_*locale*.properties.

   *Note:* We recommend that you make a backup copy of the file before proceeding.
3. Modify the values to the right of the equal sign.
4. Save the file.

The hcmtitles.properties file contains the titles of the standard stored process reports that are available with SAS Human Capital Management. The titles can be modified to suit customer needs, as follows:

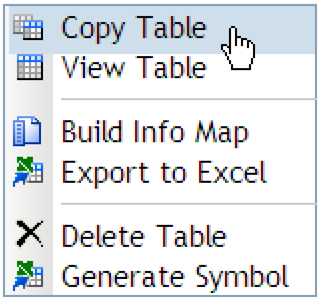
1. Change your current directory to `!SASROOT\hrds\sasmisc`.
2. Open the appropriate version of the file for editing. The sasmisc folder contains localized versions of the file, in the form hcmtitles_*locale*.properties.

   *Note:* We recommend that you make a backup copy of the file before proceeding.
3. Modify the values to the right of the equal sign.
4. Save the file.

## Copy a Table

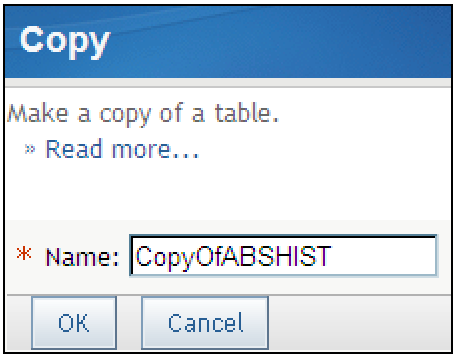
On the **Data** tab of the Administration application, you can copy a table as a backup or for some other purpose. Follow these steps:

1. From the navigation tree, select **Tables**.
2. Click the action menu at the left of the table and select **Copy Table**.



3. In the Copy dialog box, type a name for the table.

   The name must conform to naming restrictions for MySQL tables.



Copy

Make a copy of a table.
» Read more…

* Name: CopyOfABSHIST

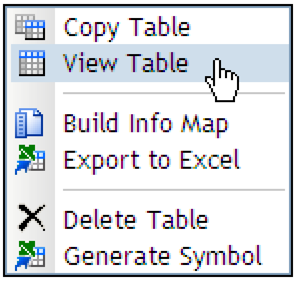
4. Click **OK**.

The table is copied to the HCM database. It is not registered in the metadata repository, and it is not made available to SAS Human Capital Management. To perform those tasks, see "Add a Table" on page 11.

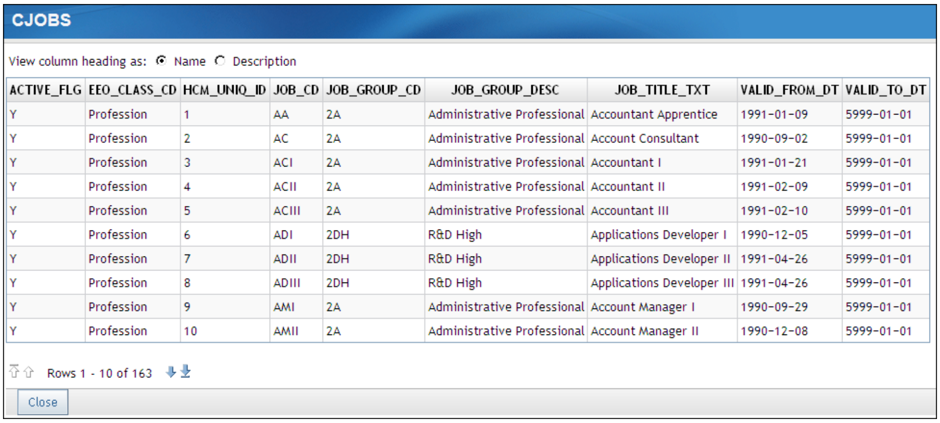
## *View a Table*

To view table data, click the **Data** tab in the Administration application. Then follow these steps:

1. Click **Refresh Cache** to ensure that you are viewing current data.
2. From the navigation tree, select **Tables**.
3. Click the action menu ▾ at the left of the table and select **View Table**.



A separate browser window displays the table data.



CJOBS

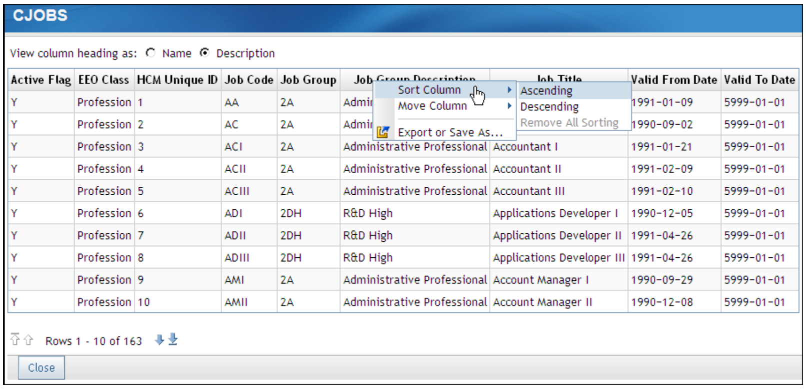
View column heading as: ● Name ○ Description

| ACTIVE_FLG | EEO_CLASS_CD | HCM_UNIQ_ID | JOB_CD | JOB_GROUP_CD | JOB_GROUP_DESC | JOB_TITLE_TXT | VALID_FROM_DT | VALID_TO_DT |
|---|---|---|---|---|---|---|---|---|
| Y | Profession | 1 | AA | 2A | Administrative Professional | Accountant Apprentice | 1991-01-09 | 5999-01-01 |
| Y | Profession | 2 | AC | 2A | Administrative Professional | Account Consultant | 1990-09-02 | 5999-01-01 |
| Y | Profession | 3 | ACI | 2A | Administrative Professional | Accountant I | 1991-01-21 | 5999-01-01 |
| Y | Profession | 4 | ACII | 2A | Administrative Professional | Accountant II | 1991-02-09 | 5999-01-01 |
| Y | Profession | 5 | ACIII | 2A | Administrative Professional | Accountant III | 1991-02-10 | 5999-01-01 |
| Y | Profession | 6 | ADI | 2DH | R&D High | Applications Developer I | 1990-12-05 | 5999-01-01 |
| Y | Profession | 7 | ADII | 2DH | R&D High | Applications Developer II | 1991-04-26 | 5999-01-01 |
| Y | Profession | 8 | ADIII | 2DH | R&D High | Applications Developer III | 1991-04-26 | 5999-01-01 |
| Y | Profession | 9 | AMI | 2A | Administrative Professional | Account Manager I | 1990-09-29 | 5999-01-01 |
| Y | Profession | 10 | AMII | 2A | Administrative Professional | Account Manager II | 1990-12-08 | 5999-01-01 |

Rows 1 - 10 of 163

4. By default, the table column names are displayed as headings. To view the column labels instead, select **Description** from the radio buttons at the top of the screen.
5. To scroll through the table rows, use the scroll buttons at the foot of the page.

6. To sort the data by means of a table column, right-click the column heading and select **Sort Column ⇨ Ascending** or **Sort Column ⇨ Descending**.



CJOBS

View column heading as: ○ Name ● Description

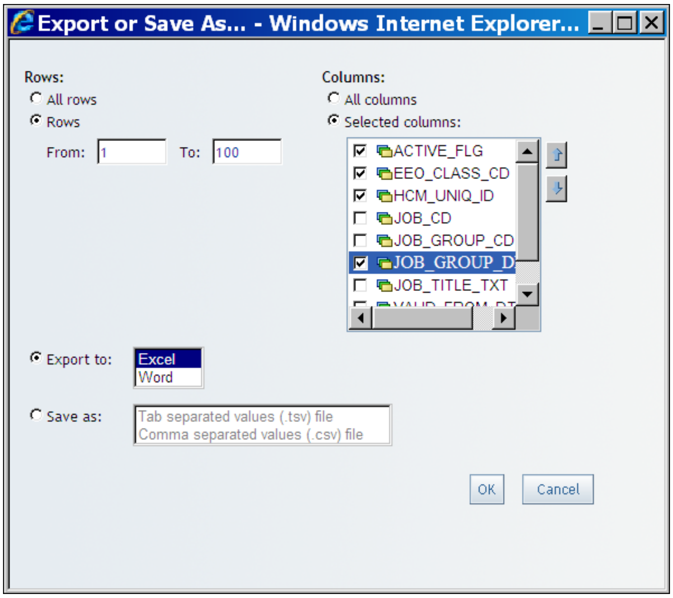
| Active Flag | EEO Class | HCM Unique ID | Job Code | Job Group | Job Group Description | Job Title | Valid From Date | Valid To Date |
|---|---|---|---|---|---|---|---|---|
| Y | Profession | 1 | AA | 2A | Admi |  | 1991-01-09 | 5999-01-01 |
| Y | Profession | 2 | AC | 2A | Admi |  | 1990-09-02 | 5999-01-01 |
| Y | Profession | 3 | ACI | 2A | Administrative Professional | Accountant I | 1991-01-21 | 5999-01-01 |
| Y | Profession | 4 | ACII | 2A | Administrative Professional | Accountant II | 1991-02-09 | 5999-01-01 |
| Y | Profession | 5 | ACIII | 2A | Administrative Professional | Accountant III | 1991-02-10 | 5999-01-01 |
| Y | Profession | 6 | ADI | 2DH | R&D High | Applications Developer I | 1990-12-05 | 5999-01-01 |
| Y | Profession | 7 | ADII | 2DH | R&D High | Applications Developer II | 1991-04-26 | 5999-01-01 |
| Y | Profession | 8 | ADIII | 2DH | R&D High | Applications Developer III | 1991-04-26 | 5999-01-01 |
| Y | Profession | 9 | AMI | 2A | Administrative Professional | Account Manager I | 1990-09-29 | 5999-01-01 |
| Y | Profession | 10 | AMII | 2A | Administrative Professional | Account Manager II | 1990-12-08 | 5999-01-01 |

Rows 1 - 10 of 163

To restore the original row order, select **Sort Column ⇨ Remove All Sorting**.

7. To move a column one position to the left or the right, right-click the column heading and select **Move Column ⇨ Left** or **Move Column ⇨ Right**.

8. To export the table data, right-click a column heading and select **Export or Save As**. In the dialog box that appears, make the following selections:



Export or Save As... - Windows Internet Explorer...

Rows:
○ All rows
● Rows
From: 1 To: 100

Columns:
○ All columns
● Selected columns:
☑ ACTIVE_FLG
☑ EEO_CLASS_CD
☑ HCM_UNIQ_ID
☐ JOB_CD
☐ JOB_GROUP_CD
☑ JOB_GROUP_D
☐ JOB_TITLE_TXT

● Export to: Excel
Word

○ Save as: Tab separated values (.tsv) file
Comma separated values (.csv) file

**Rows**

To export all table rows, select the **All rows** radio button.

To export a subset of table rows, select the **Rows** radio button. Enter starting and ending row values in the **From** and **To** boxes.

**Columns**

To export all columns from this table, select the **All columns** radio button.

To export a subset of table columns, select the **Selected columns** radio button. From the list, select the check box for each column that you want to export. Use the up and down buttons to modify the column order in the output.

*Note:* If the selected columns for export are too many, you might receive an error using Internet Explorer. This can occur if the selected columns exceed the browser's URL limit of 2083 characters.

**Export to**
Select this radio button to export the data to a Microsoft Excel worksheet or Microsoft Word document.

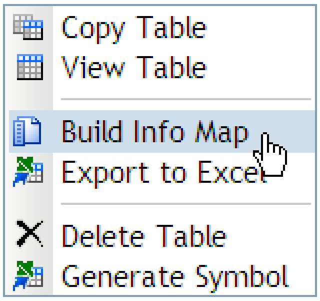
**Save as**
Select this radio button to save the data as a tab-separated values (TSV) file or as a comma-separated values (CSV) file.

*Note:* Be aware that if you are viewing formatted values in the table, then formatted values are exported. For more information about exporting table data, see "About Importing and Exporting Tables" on page 10.

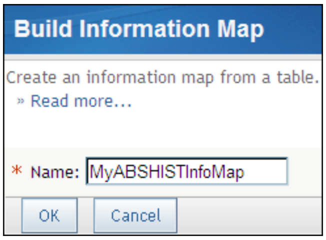
## Create an Information Map from a Table

To create an information map from a table, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigation tree, select **Tables**.
2. Click the action menu at the left of the table and select **Build Info Map**.



3. On the Build Information Map dialog box, enter a name for the map and click **OK**.



Build Information Map

Create an information map from a table.
» Read more...

* Name: MyABSHISTInfoMap

*Note:* Take care in naming the information map. If the folder already contains an information map with this name, it will be replaced, and any dashboards or reports that are based on the old information map will now use the new version. If the new version references a different data source, the dashboards and reports might not work correctly.

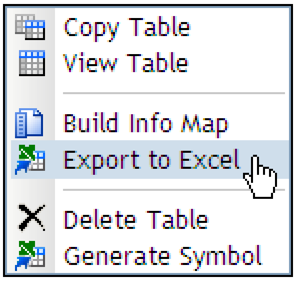
All columns are included in the information map, which is written to the default folder location in the workspace.(See "Default Folder Locations" on page 7.)

*Note:* Users must have both ReadMetadata and Read permission for an information map in order to access its data in SAS Web Report Studio. If the information map is built on a cube, users must also have Read permission for the cube.

## *Export a Table*

On the **Data** tab of the Administration application, you can export table data to a Microsoft Excel worksheet. Follow these steps:

1. Click **Refresh Cache** to ensure that you are exporting current data.
2. From the navigation tree, select **Tables**.
3. Click the action menu ▾ at the left of the table and select **Export to Excel**.



From the File Download dialog box, you can choose whether to view the file in Microsoft Excel or save the data to a file.

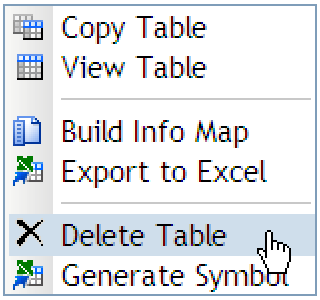
All rows are exported (subject to security provisions), and the column names are used as headings. (To export a subset of table rows, see "View a Table" on page 18.)

For more information about exporting tables, see "About Importing and Exporting Tables" on page 10.

## *Delete a Table*

To remove a table so that it is no longer available to SAS Human Capital Management, click the **Data** tab in the Administration application. Then follow these steps:

1. If you previously mapped the table to a hierarchy, remove the mapping. See "View Hierarchy Mappings" on page 24.
2. In the navigation tree, select **Tables**.
3. Click the action menu ▾ at the left of the table and select **Delete Table**.



The table entry is removed from the SAS_HCMMETATABLE table, and its columns are removed from the SAS_HCMMETACOLUMN table. The table definition is not removed from the metadata repository, and the table itself is not removed from the HCM database.

### *Generate Search Symbols*

Search symbols are abbreviations for columns that can be used in the general Search. For example, if the symbol for EMPLOYEE_ID was EI, you could use a search string such as **`EI=2973`**, instead of spelling out the column name.

You can enter a search symbol manually. See "Modify Column Attributes" on page 14.

Alternatively, SAS Human Capital Management can generate one or more search symbols. Generated symbols are taken from the column label and use the current language and encoding for that label.

To generate symbols for all columns in a table, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigation tree, select **Tables**.
2. Click the action menu ▾ at the left of the table and select **Generate Symbol**.

   When you look at the **Search Symbol** attribute for the table, you will see symbols for each column.

   *Note:* The symbols are immediately written to the database. There is no need to click **Apply Changes**.

To create a symbol for a single column in a table:

1. From the navigation tree, select the table name.
2. Click the action menu ▾ at the left of the column and select **Generate Symbol**.
3. To apply your changes to this table, click **Apply Changes**.

   *Note:* Search symbols are not propagated to other tables, to avoid possible duplication. (You might have already defined the identical search symbol for a different column in another table.)

## Working with Hierarchies

### *What Is a Hierarchy?*

A dimension is a set of elements (members) of a particular type. It can have one or more hierarchies, each of which includes some or all of the members of a dimension. Most hierarchies are tree structures that consist of parent-child relationships, although it is possible to have a flat hierarchy. The sample data contains two hierarchies within the ORG dimension: INTORG_HR, which represents the departmental structure of the organization; and INTORG_MGR, which represents the management structure of the organization.

### *What Is a Hierarchy Mapping?*

Hierarchy members are identified by their codes. For example, in the INTORG_HR hierarchy, the code represents a department within the organization. In the Employee Browser, when a user selects a department, the Employee Browser displays a list of all the employees that belong to that department.

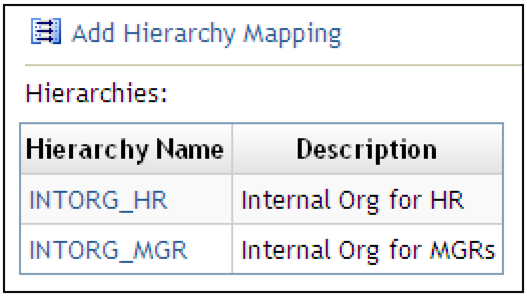
The records are selected by matching the hierarchy code to values in an information table, such as the Employee Master table. The hierarchy mapping specifies which column of the information table contains the matching codes—in this case, the department code.

Hierarchy mappings are used in the Employee Browser and in an organization analysis.

## *View Available Hierarchies*

To view the hierarchies that are available for use by the Employee Browser and in an organization analysis, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigational tree, select **Hierarchies**. The list of available hierarchies appears.



Hierarchies:

| Hierarchy Name | Description |
|---|---|
| INTORG_HR | Internal Org for HR |
| INTORG_MGR | Internal Org for MGRs |

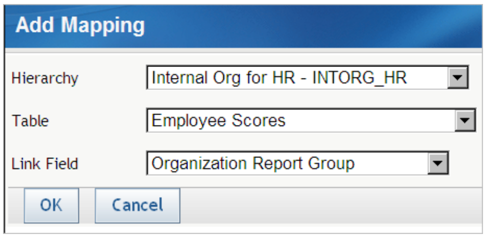
2. To view information about a hierarchy, click its name in the list.
3. To add a hierarchy mapping, click **Add Hierarchy Mapping**.

   For details, see "Add a Hierarchy Mapping" on page 23.

## *Add a Hierarchy Mapping*

A hierarchy mapping creates a relationship between a hierarchy and a table in the HCM database. To add a hierarchy mapping, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigation tree, select **Hierarchies**.
2. Click **Add Hierarchy Mapping**.
3. From the drop-down lists in the Add Mapping dialog box, select a hierarchy, table, and link field:



Hierarchy: Internal Org for HR - INTORG_HR

Table: Employee Scores

Link Field: Organization Report Group

**Hierarchy**
Select a hierarchy for this mapping.

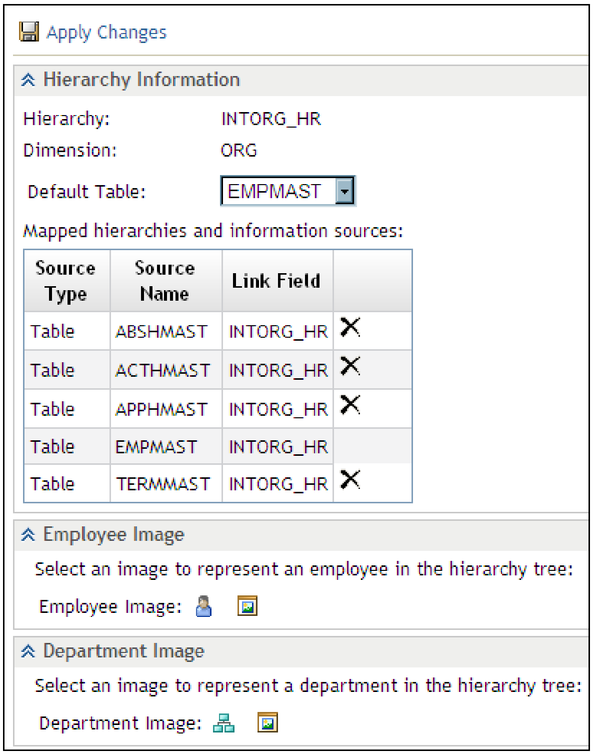
**Table**
Select an information table for this mapping.

**Link Field**
Select the table column whose values match the hierarchy codes.

## View Hierarchy Mappings

To view the mappings for a hierarchy, and to select the default hierarchy mapping, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigational tree, select **Hierarchies** ⇨ ***hierarchy-name***. The hierarchy information is displayed.



Apply Changes

Hierarchy Information

Hierarchy: INTORG_HR
Dimension: ORG

Default Table: EMPMAST

Mapped hierarchies and information sources:

| Source Type | Source Name | Link Field | |
|---|---|---|---|
| Table | ABSHMAST | INTORG_HR | ✕ |
| Table | ACTHMAST | INTORG_HR | ✕ |
| Table | APPHMAST | INTORG_HR | ✕ |
| Table | EMPMAST | INTORG_HR | |
| Table | TERMMAST | INTORG_HR | ✕ |

Employee Image

Select an image to represent an employee in the hierarchy tree:

Employee Image:

Department Image

Select an image to represent a department in the hierarchy tree:

Department Image:

2. To select a hierarchy mapping as the default, select the source table from the **Default Table** drop-down list.

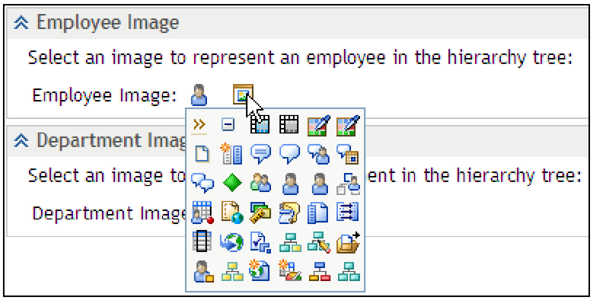
   When a user creates an organization analysis and selects this hierarchy, the default mapping determines which table is presented as the default. However, the user can override the default and select a different table from the available mappings.

   *Note:* The default mapping applies only to organization analysis. To select defaults for an employee profile, see "Create an Employee Profile" on page 48.

3. To delete a hierarchy mapping, click the black delete button ✕ at the right of the mapping.

   *Note:* You cannot modify a hierarchy mapping, but you can delete it and then re-create it.

4. To select an image to represent employees in an organization analysis, click the **Employee Image** button:



Employee Image

Select an image to represent an employee in the hierarchy tree:

Employee Image:

Department Image

Select an image to represent a department in the hierarchy tree:

Department Image

From the pop-up display of images, make a selection.

5. To select an image to represent departments in an organization analysis, click the **Department Image** button and make a selection from the pop-up display.

   The employee and department images are applied to new and existing organization charts.

6. After you have modified this page, click **Apply Changes**.

# Working with Cubes

## Using Cubes in SAS Human Capital Management

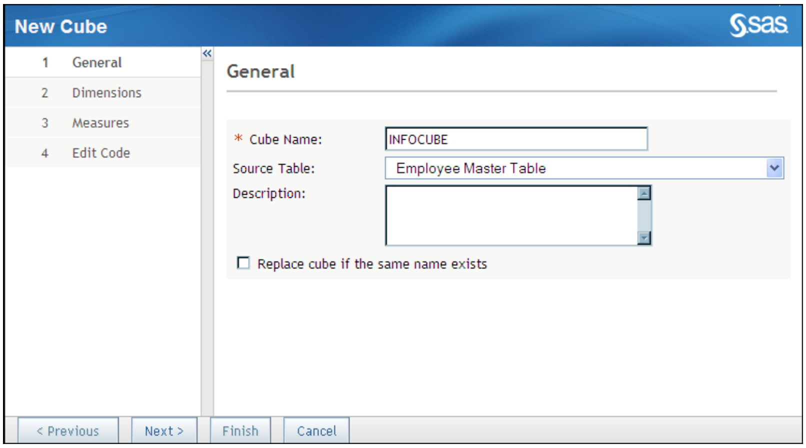
A cube is a set of data that is organized and structured in a hierarchical, multidimensional arrangement that provides more efficient access to data than traditional relational databases. SAS cubes are designed to offer fast data access and efficient data storage. In many cases, a basic cube without additional aggregations can be smaller than the input data, because the process of creating the cube consolidates records. A good rule of thumb is, the larger your input data, the greater the storage gain by loading data into a cube.

In SAS Human Capital Management, you can create cubes from hierarchical data that is stored in the HCM tables, and you can refresh or rebuild cubes. Cubes can be used as input for information maps and can be explored directly in SAS Web Report Studio.

## Create a Cube

To create a cube, click the **Data** tab in the Administration application. Then follow these steps:

1. Click **New Cube** to open the New Cube wizard.



* Cube Name: INFOCUBE

Source Table: Employee Master Table

Description:

☐ Replace cube if the same name exists

2. Provide general information for your cube.

   a. Type a cube name, such as **`INFOCUBE`**.

      *Note:* If you want to be able to run a SAS Data Integration Studio job to refresh the new cube, then the cube name needs to be all upper case.

   b. Specify a source table for your cube, such as the Employee Master Table.

   c. (Optional) Provide a description to associate with your cube.

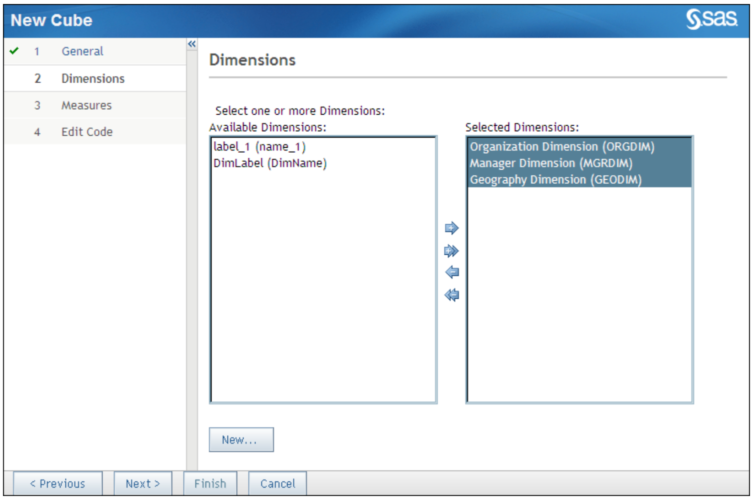
   d. To replace a preexisting cube with the same name, select **Replace cube if the same name exists**.

   e. Click **Next**.

      *Note:* Column-level security is supported in a cube. Row-level security is not.

3. Select the dimensions for your cube.

   A dimension is a data element that serves as a category for each item in a data set. Examples of dimensions include the employee's organization, manager, or location.



Select one or more Dimensions:

Available Dimensions:
label_1 (name_1)
DimLabel (DimName)

Selected Dimensions:
Organization Dimension (ORGDIM)
Manager Dimension (MGRDIM)
Geography Dimension (GEODIM)

   a. To include a dimension in a cube, move it from the **Available Dimensions** list to the **Selected Dimensions**list using the arrows.

      The default dimensions are as follows:

      - **Organization Dimension (ORGDIM)**: The cube includes a dimension and hierarchy with the same name as the dimension (ORGDIM), with levels INTORG_HR5 through INTORG_HR1.

- **Manager Dimension (MGRDIM)**: The cube includes a dimension and hierarchy named MGRDIM, with levels INTORG_MGR5 through INTORG_MGR1.

- **Geography Dimension (GEODIM)**: The cube includes a dimension and hierarchy named GEODIM, with these levels: COUNTRY_CD, STATE_REGION_CD, CITY_NM.

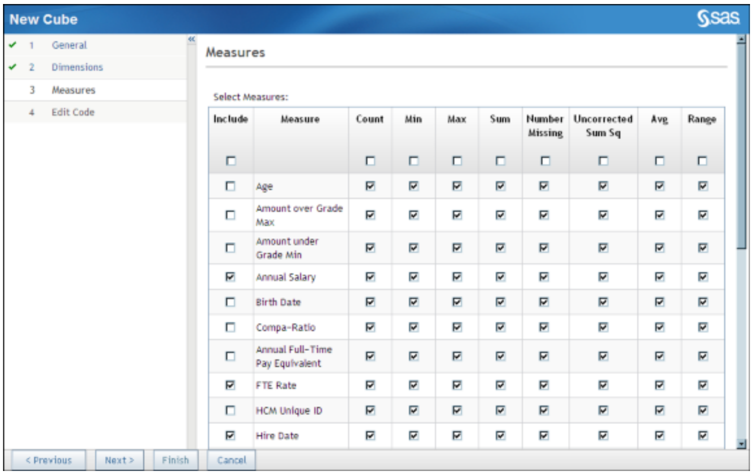
The ORGDIM, MGRDIM, and GEODIM dimensions will display whether they are available in the source table. If one or more of these dimensions are selected, and the source table does not have the columns associated with them, they will not be included in the new cube.

To create additional dimensions for the cube, click **New**. (See "Create a New Dimension for a Cube" on page 29.)

*Note:* For information about maximum sizes, refer to the documentation for PROC OLAP at `support.sas.com/documentation/cdl_main/index.html`.

b. Click **Next**.

4. Select measures to be displayed for your cube. The selection is based on the numeric columns (including date and currency columns) that are marked as analysis columns in the attributes for this table.



Measures

Select Measures:

| Include | Measure | Count | Min | Max | Sum | Number Missing | Uncorrected Sum Sq | Avg | Range |
|---|---|---|---|---|---|---|---|---|---|
| □ | | □ | □ | □ | □ | □ | □ | □ | □ |
| □ | Age | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | Amount over Grade Max | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | Amount under Grade Min | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| ☑ | Annual Salary | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | Birth Date | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | Compa-Ratio | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | Annual Full-Time Pay Equivalent | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| ☑ | FTE Rate | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| □ | HCM Unique ID | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| ☑ | Hire Date | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |

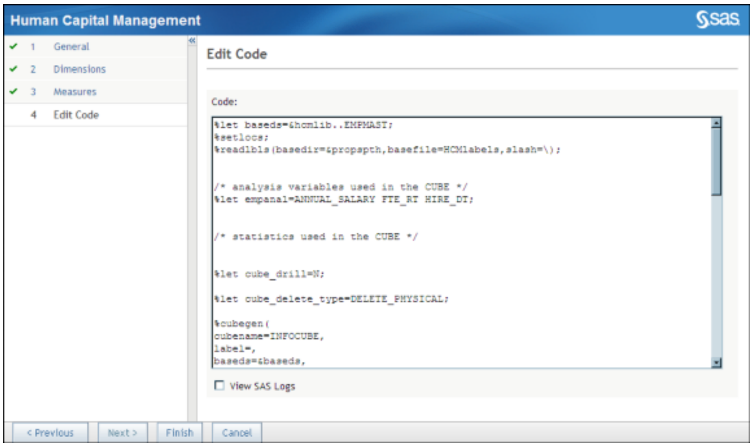
a. Select the **Include** check box for each measure that you want to include in your cube.

b. For each included measure, select the check box for each statistic option that you want a user to be able to choose for that measure. Select a check box under a statistic heading to select or deselect that statistic's check boxes for each available measure.

*Note:* If you select a statistic that was not selected as one of the column attributes, then the column attributes are updated as well.

c. Click **Next**.

5. Based on your selections, the New Cube wizard generates SAS code to create your cube.



Code:

```
%let baseds=&hcmlib..EMPMAST;
%setlocs;
%readlbls(basedir=&propspth,basefile=HCMlabels,slash=\);


/* analysis variables used in the CUBE */
%let empanal=ANNUAL_SALARY FTE_RT HIRE_DT;


/* statistics used in the CUBE */


%let cube_drill=N;

%let cube_delete_type=DELETE_PHYSICAL;

%cubegen(
cubename=INFOCUBE,
label=,
baseds=&baseds,
```

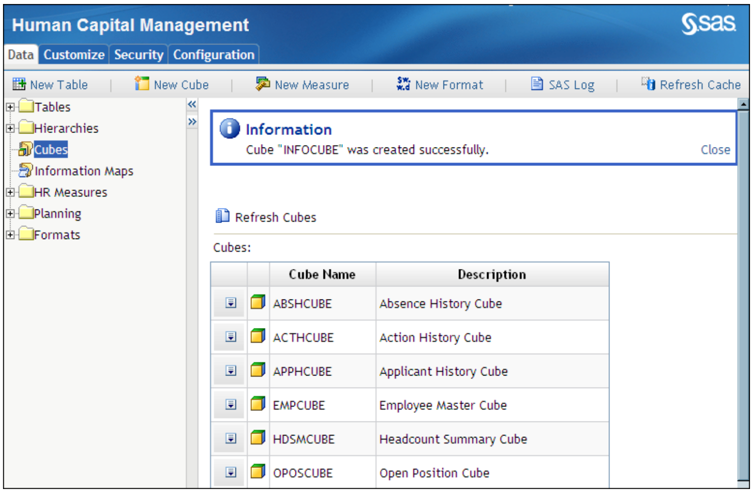
☐ View SAS Logs

a. If your site uses double-byte character sets (DBCS), you must modify the SAS code so that the names and labels for the cube and any dimensions, measures, hierarchies, and columns do not use DBCS (DBCS are not supported in the creation of cubes).

   (Optional) You can make additional customizations to the SAS code for your cube. However, additional modifications are not recommended unless you are familiar with SAS programming.

b. Select the **View SAS Logs** check box to open a pop-up dialog box that contains a SAS log of the cube's creation. Check this log to make sure that the cube was built successfully.

6. Click **Finish** to create your cube. If your cube is created successfully, a success message appears at the top of the data tab, and the cube appears in the list when you select **Cubes** in the navigation tree.



Information

Cube "INFOCUBE" was created successfully.

Cubes:

| Cube Name | Description |
|---|---|
| ABSHCUBE | Absence History Cube |
| ACTHCUBE | Action History Cube |
| APPHCUBE | Applicant History Cube |
| EMPCUBE | Employee Master Cube |
| HDSMCUBE | Headcount Summary Cube |
| OPOSCUBE | Open Position Cube |

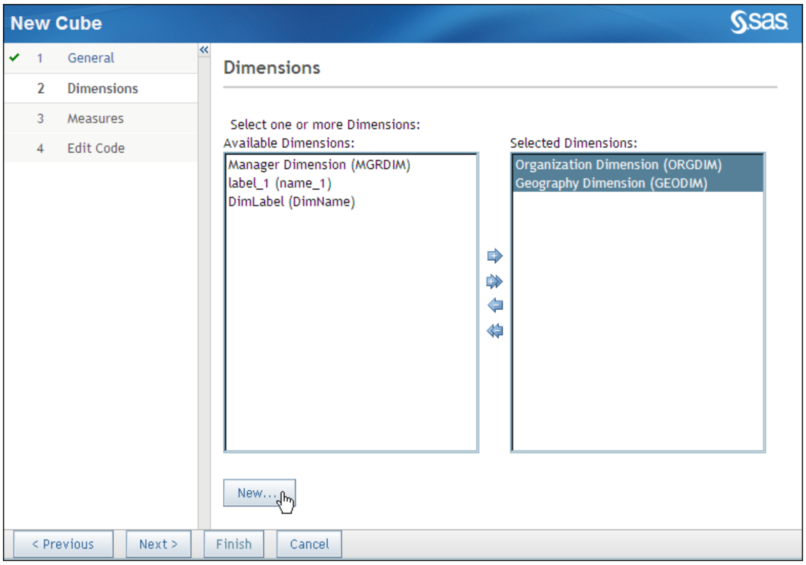
Your cube is written to the default folder location in the workspace. (See "Default Folder Locations" on page 7.)

7. Rather than opening a cube directly, we recommend that you create an information map from the cube in a shared folder. You (or your end users) can then create a SAS report based on that information map. For more information about creating an information map from a cube, see "Managing Cubes" on page 30.

### *Create a New Dimension for a Cube*

While you are creating a cube in the New Cube wizard, you can create additional dimensions, as follows:

1. After you have provided general cube information, select **New** on the Dimensions page of the New Cube wizard.



Select one or more Dimensions:

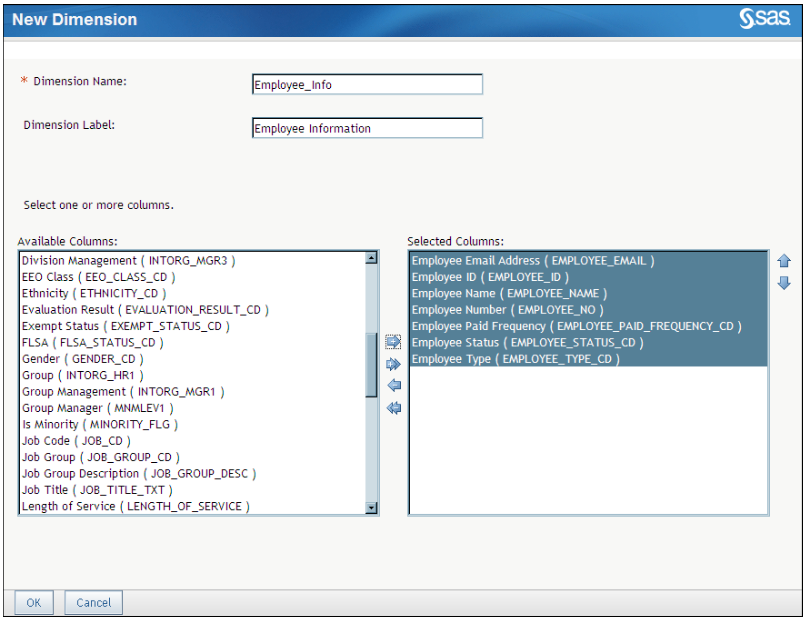
Available Dimensions:

Manager Dimension (MGRDIM)
label_1 (name_1)
DimLabel (DimName)

Selected Dimensions:

Organization Dimension (ORGDIM)
Geography Dimension (GEODIM)

2. Provide information for the new dimension.



* Dimension Name: Employee_Info

Dimension Label: Employee Information

Select one or more columns.

Available Columns:

Division Management ( INTORG_MGR3 )
EEO Class ( EEO_CLASS_CD )
Ethnicity ( ETHNICITY_CD )
Evaluation Result ( EVALUATION_RESULT_CD )
Exempt Status ( EXEMPT_STATUS_CD )
FLSA ( FLSA_STATUS_CD )
Gender ( GENDER_CD )
Group ( INTORG_HR1 )
Group Management ( INTORG_MGR1 )
Group Manager ( MNMLEV1 )
Is Minority ( MINORITY_FLG )
Job Code ( JOB_CD )
Job Group ( JOB_GROUP_CD )
Job Group Description ( JOB_GROUP_DESC )
Job Title ( JOB_TITLE_TXT )
Length of Service ( LENGTH_OF_SERVICE )

Selected Columns:

Employee Email Address ( EMPLOYEE_EMAIL )
Employee ID ( EMPLOYEE_ID )
Employee Name ( EMPLOYEE_NAME )
Employee Number ( EMPLOYEE_NO )
Employee Paid Frequency ( EMPLOYEE_PAID_FREQUENCY_CD )
Employee Status ( EMPLOYEE_STATUS_CD )
Employee Type ( EMPLOYEE_TYPE_CD )

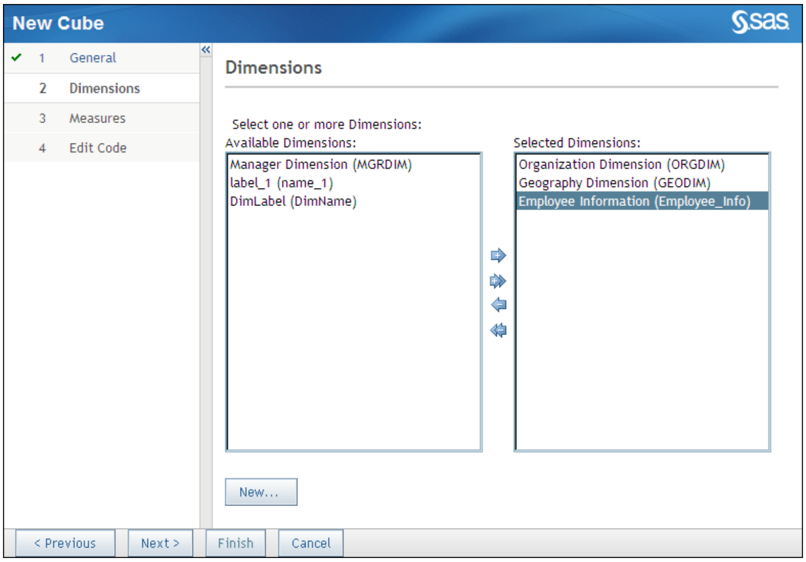
a. Provide a name for your dimension.

b. Provide a label for your dimension.

c. Select the columns that you want to use to construct your dimension. The available columns are from the character columns in the table that is being used for the cube.

d. Click **OK**.

3. Return to the New Cube wizard, select the dimension that you created from the list of available dimensions, and use the arrows to move it to the **Selected Dimensions** list.



Dimensions

Select one or more Dimensions:

Available Dimensions:

Manager Dimension (MGRDIM)
label_1 (name_1)
DimLabel (DimName)

Selected Dimensions:

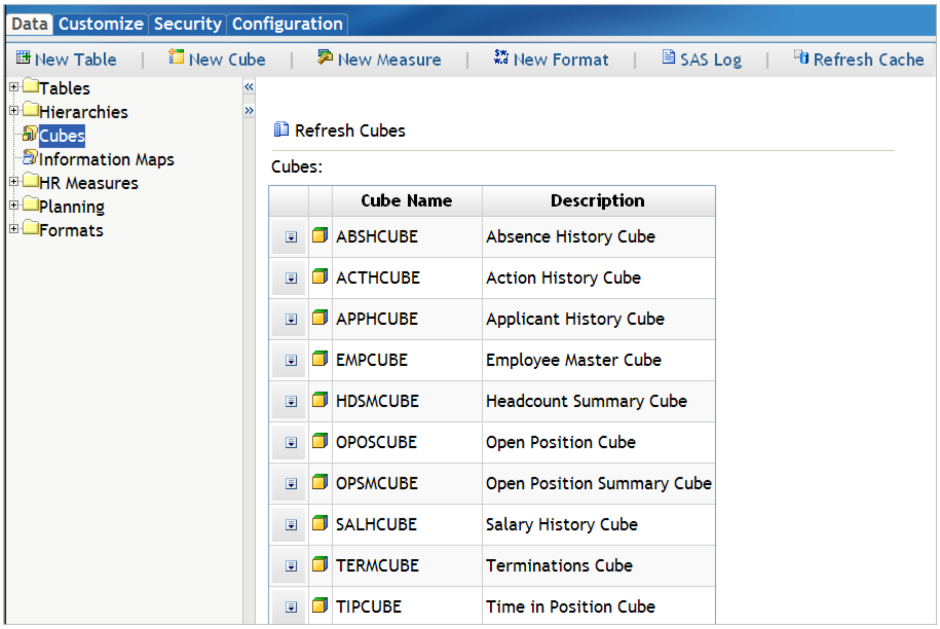
Organization Dimension (ORGDIM)
Geography Dimension (GEODIM)
Employee Information (Employee_Info)

4. Complete the remaining steps in the New Cube wizard to create your cube.

*Note:* The dimensions that you create are available for use in other cubes that are created from the same table.

## Managing Cubes

On the **Data** tab of the Administration application, select **Cubes** from the navigation tree. A list of available cubes is displayed.



Cubes:

| | Cube Name | Description |
|---|---|---|
| | ABSHCUBE | Absence History Cube |
| | ACTHCUBE | Action History Cube |
| | APPHCUBE | Applicant History Cube |
| | EMPCUBE | Employee Master Cube |
| | HDSMCUBE | Headcount Summary Cube |
| | OPOSCUBE | Open Position Cube |
| | OPSMCUBE | Open Position Summary Cube |
| | SALHCUBE | Salary History Cube |
| | TERMCUBE | Terminations Cube |
| | TIPCUBE | Time in Position Cube |

Select from the available choices:

- To view the cube data, select **View** from the action menu ▾ to the left of the cube name.

  The cube opens. (There are some caveats about opening a cube this way. See "Create a Cube" on page 25.)

- To rebuild a cube, select **Rebuild cube** from the action menu .

  When you rebuild a cube, both the physical cube (the data) and the cube structure (the metadata) are deleted and re-created. However, any existing cube permissions are saved and reapplied to the new metadata.

- To refresh a cube, select **Refresh** from the action menu.

  When you refresh a cube, the physical cube is deleted and a new cube is created with current data, but the cube structure in the metadata repository is not changed, and cube permissions are not changed.

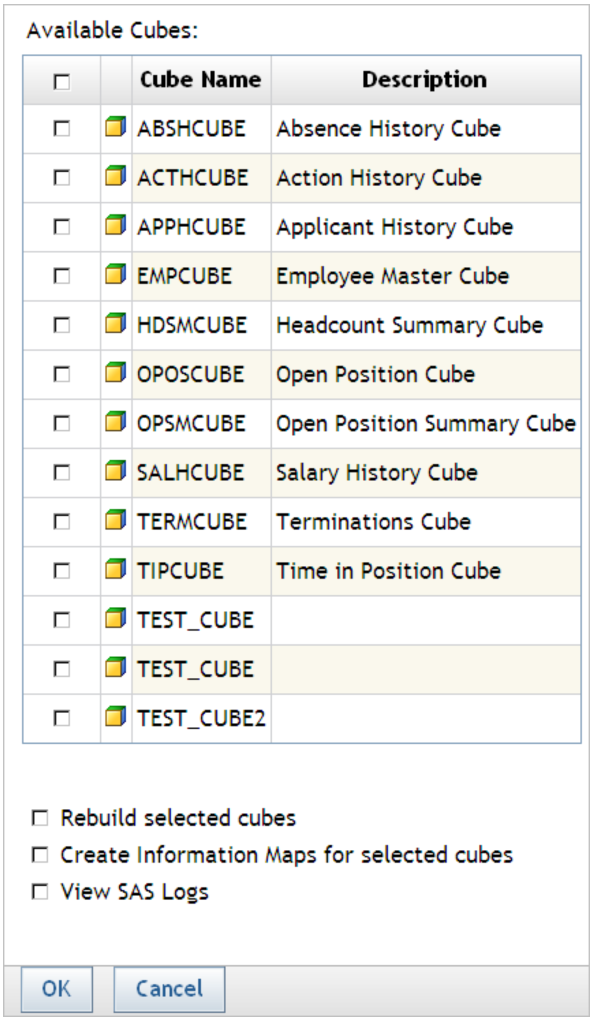
- To create an information map from a cube, select **Build Information Maps** from the action menu.

  An information map, named ***cube-name*_MAP**, is created in the default location for information maps.

- To delete a cube, select **Delete** from the action menu.

### Refresh or Rebuild Multiple Cubes

To refresh or rebuild multiple cubes, click **Refresh Cubes**. The dialog box that appears displays a list of available cubes.



Available Cubes:

| □ | Cube Name | Description |
|---|---|---|
| □ | ABSHCUBE | Absence History Cube |
| □ | ACTHCUBE | Action History Cube |
| □ | APPHCUBE | Applicant History Cube |
| □ | EMPCUBE | Employee Master Cube |
| □ | HDSMCUBE | Headcount Summary Cube |
| □ | OPOSCUBE | Open Position Cube |
| □ | OPSMCUBE | Open Position Summary Cube |
| □ | SALHCUBE | Salary History Cube |
| □ | TERMCUBE | Terminations Cube |
| □ | TIPCUBE | Time in Position Cube |
| □ | TEST_CUBE | |
| □ | TEST_CUBE | |
| □ | TEST_CUBE2 | |

□ Rebuild selected cubes
□ Create Information Maps for selected cubes
□ View SAS Logs

Follow these steps:

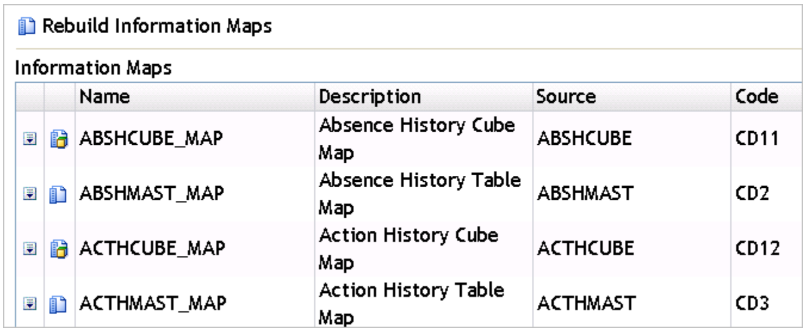
1. Select the check box next to each cube that you want to affect.
2. Select **Rebuild selected cubes** if you want to delete both the physical cube and the metadata.

   If you do not select this check box, the cube is refreshed instead of being rebuilt. For an explanation of the difference between refreshing and rebuilding a cube, see "Managing Cubes" on page 30.
3. If you select **Create Information Maps for selected cubes**, then an information map is created for each cube that you checked.
4. If you select **View SAS Logs**, a separate window is opened to display the log file. When you click **OK**, the log file displays the operation's progress. This feature can be helpful if you are rebuilding a large number of cubes.
5. Click **OK** to begin the operation.

# Working with Information Maps

A number of information maps are included with SAS Human Capital Management. Those information maps are generated as part of the installation and configuration steps. In the Administration application, you can view available information maps, open them in SAS Web Report Studio, and rebuild the maps. You can also generate new information maps from tables that you import into SAS Human Capital Management, or from cubes.

On the **Data** tab, select **Information Maps** from the navigation tree to view the information maps that have been generated, as in this example:



Rebuild Information Maps

Information Maps

| Name | Description | Source | Code |
|---|---|---|---|
| ABSHCUBE_MAP | Absence History Cube Map | ABSHCUBE | CD11 |
| ABSHMAST_MAP | Absence History Table Map | ABSHMAST | CD2 |
| ACTHCUBE_MAP | Action History Cube Map | ACTHCUBE | CD12 |
| ACTHMAST_MAP | Action History Table Map | ACTHMAST | CD3 |

To rebuild a single information map, click the action menu ▾ at the left of the map name and select **Rebuild**.

To rebuild all the information maps, click **Rebuild Information Maps**.

To open the information map in SAS Web Report Studio, click the action menu ▾ at the left of the information map and select **View**.

*Note:* Users need Read and ReadMetadata permissions for folders that contain information maps. Typically these permissions are assigned to the HCM Solution Users group.

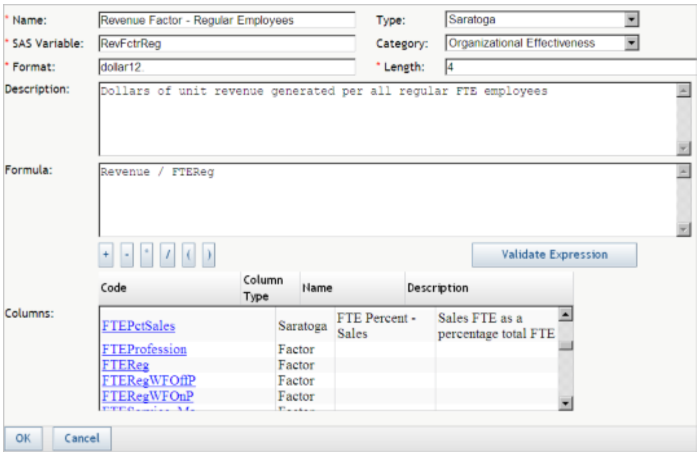
# Working with HR Measures

## *About HR Measures*

SAS Human Capital Management provides an extensive collection of predefined measures that are designed to be used with data provided by the Saratoga Institute. (See "Metrics in SAS Human Capital Management" on page 683.) In addition, HCM administrators can define a set of custom measures, based on the Saratoga measures or created to suit other criteria.

## *Create an HR Measure*

On the **Data** tab of the Administration application, you can create HR measures and modify existing measures. To create a measure, click **New Measure** in the toolbar. In the New Measure dialog box, respond to these prompts:



* Name: Revenue Factor - Regular Employees

Type: Saratoga

* SAS Variable: RevFctrReg

Category: Organizational Effectiveness

* Format: dollar12.

* Length: 4

Description: Dollars of unit revenue generated per all regular FTE employees

Formula: Revenue / FTEReg

Columns:

| Code | Column Type | Name | Description |
|---|---|---|---|
| FTEPctSales | Saratoga | FTE Percent - Sales | Sales FTE as a percentage total FTE |
| FTEProfession | Factor | | |
| FTEReg | Factor | | |
| FTERegWFOffP | Factor | | |
| FTERegWFOnP | Factor | | |

**Name**
Specify a name to identify this measure, such as **Revenue Factor**. This name is used when you are creating a scorecard or dashboard.

**Type**
Select the measure type: **Saratoga** for a Saratoga measure, or **HCM** for a custom measure.

**SAS Variable**
When you create a measure, it becomes a SAS variable. Specify a name for this variable, using SAS naming conventions. The name must be unique among the defined measures.

**Category**
Select a category for this measure. The category determines this measure's grouping on the Measures page. It also is used when a user creates a model for the SAS BI Dashboard: if the model uses the SAS Human Capital Management metrics, users select a category of measures to display.

If the measure does not fit in any of the standard categories, select **Custom Metrics**.

**Format**
Specify the display format for this measure. You can specify any numeric format from the formats catalog, which includes custom HCM formats.

**Length**
Specify the length of this measure, in bytes.

**Description**
Enter a description for this measure, which will be displayed on the Measures page.

**Formula**
Build the formula by selecting from the **Columns** list and from the operator buttons (**+**, **-**, *****, **/**, **(**, and **)**).

The **Columns** list includes measures (standard measures and those defined at a site) and factors. Factors are precalculated values that are stored in the HCM database. They might be considered as the basic building blocks of the measures.

To select a measure or factor, click its code (SAS variable name) in the list.

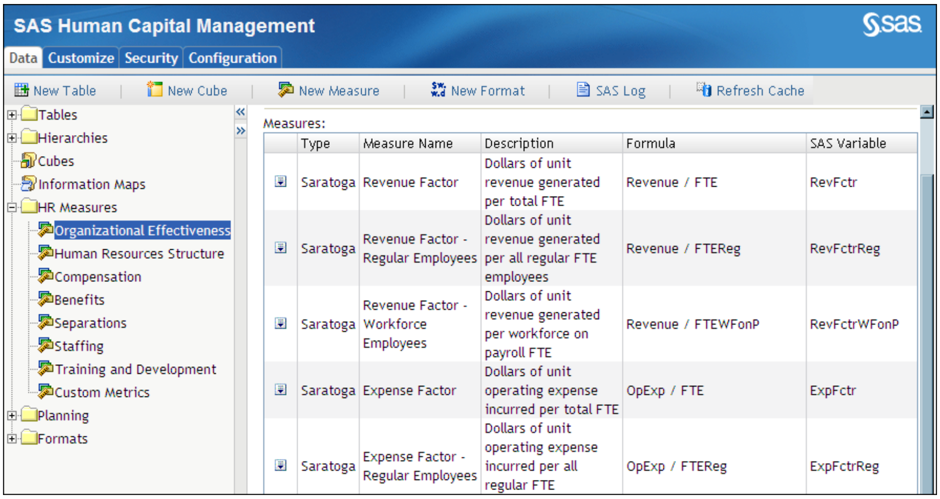
**Validate Expression**
Click this button to validate the expression syntax. (This function checks only the open and closed parentheses in the expression.)

Click **OK** to save the measure.

### *Manage HR Measures*

To display all the measures that are defined, select **HR Measures** from the navigation tree on the **Data** tab.

To display the measures in a particular category, select **HR Measures** ⇨ ***category-name***. The list of corresponding measures is displayed.



| | Type | Measure Name | Description | Formula | SAS Variable |
|---|---|---|---|---|---|
| ▾ | Saratoga | Revenue Factor | Dollars of unit revenue generated per total FTE | Revenue / FTE | RevFctr |
| ▾ | Saratoga | Revenue Factor - Regular Employees | Dollars of unit revenue generated per all regular FTE employees | Revenue / FTEReg | RevFctrReg |
| ▾ | Saratoga | Revenue Factor - Workforce Employees | Dollars of unit revenue generated per workforce on payroll FTE | Revenue / FTEWFonP | RevFctrWFonP |
| ▾ | Saratoga | Expense Factor | Dollars of unit operating expense incurred per total FTE | OpExp / FTE | ExpFctr |
| ▾ | Saratoga | Expense Factor - Regular Employees | Dollars of unit operating expense incurred per all regular FTE | OpExp / FTEReg | ExpFctrReg |

For information about the fields in this display, see "Create an HR Measure" on page 33.

- To edit a measure, click the action menu ▾ at the left of the measure and select **Edit**.
- To delete a measure, click the action menu ▾ at the left of the measure and select **Delete**.
- To run an ETL job that calculates values for the measures by year, click **Run Job**.

  Typically, the ETL job is run on a regular schedule. You can also run this job from the Administration application. The measure values are calculated in this order:

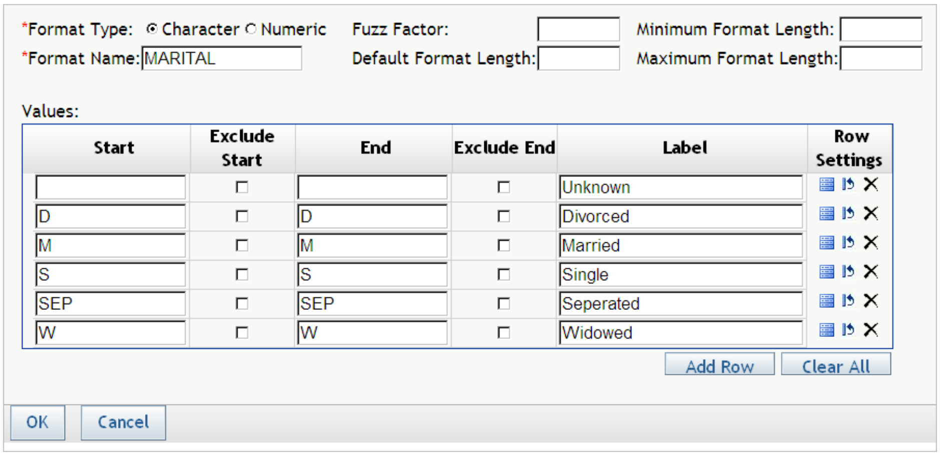
  1. Factors (in the order in which the factors were defined)
  2. Saratoga measures (in the order in which the measures were defined)
  3. HCM measures (in the order in which the measures were defined)

  (The order is important because frequently a measure combines the values from two or more measures.)

## Working with Formats

### *About Display Formats*

In its data displays, SAS Human Capital Management uses the standard display formats, which determine attributes such as the length of a character string, the way a date is displayed, or the format of numeric items, including currencies. In addition, SAS Human Capital Management uses a number of custom formats, which associate values with formatted text strings. For example, here is the definition of the MARITAL format, a character format that describes possible marital status:



*Format Type: ● Character ○ Numeric Fuzz Factor: Minimum Format Length:

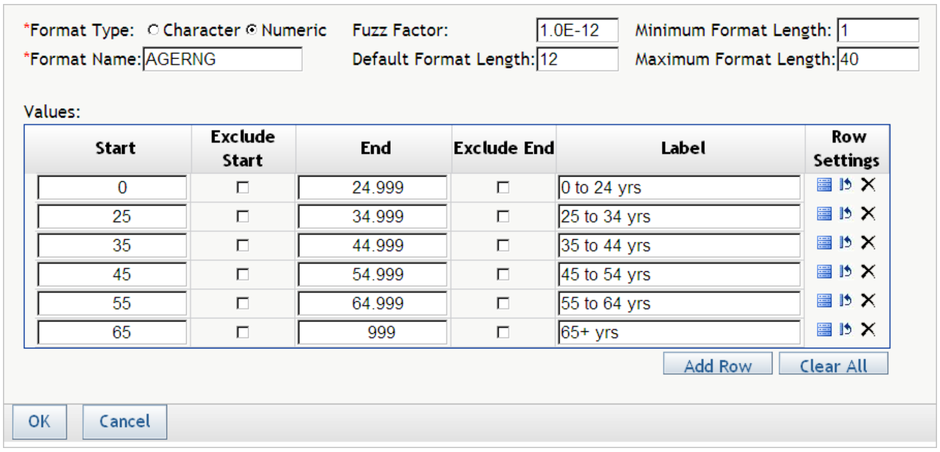
*Format Name: MARITAL Default Format Length: Maximum Format Length:

Values:

| Start | Exclude Start | End | Exclude End | Label | Row Settings |
|---|---|---|---|---|---|
| | □ | | □ | Unknown | ✕ |
| D | □ | D | □ | Divorced | ✕ |
| M | □ | M | □ | Married | ✕ |
| S | □ | S | □ | Single | ✕ |
| SEP | □ | SEP | □ | Seperated | ✕ |
| W | □ | W | □ | Widowed | ✕ |

In the database, marital status is stored using values such as **D**, **M**, **W**, and **S**. On a page, these values are displayed with the matching strings: **Divorced**, **Married**. **Widowed**, and **Single**.

Here is the definition of the AGERNG format for numeric data that represents age ranges:



*Format Type: ○ Character ● Numeric Fuzz Factor: 1.0E-12 Minimum Format Length: 1

*Format Name: AGERNG Default Format Length: 12 Maximum Format Length: 40

Values:

| Start | Exclude Start | End | Exclude End | Label | Row Settings |
|---|---|---|---|---|---|
| 0 | □ | 24.999 | □ | 0 to 24 yrs | ✕ |
| 25 | □ | 34.999 | □ | 25 to 34 yrs | ✕ |
| 35 | □ | 44.999 | □ | 35 to 44 yrs | ✕ |
| 45 | □ | 54.999 | □ | 45 to 54 yrs | ✕ |
| 55 | □ | 64.999 | □ | 55 to 64 yrs | ✕ |
| 65 | □ | 999 | □ | 65+ yrs | ✕ |

One feature of the format definition dialog box is the **Exclude Start** and **Exclude End** check boxes. In the AGERNG definition, the **0 to 24** format includes values between **0** and **24.999**. You could also have used an **End** value of **25** and selected **Exclude End**, to include all values from 0 up to but not including 25. For character data, the starting and ending values are typically identical.

*Note:* The fuzz factor and the **Exclude Start** and **Exclude End** flags are not supported in searches. Fuzz Factor is not supported in SAS Human Capital Management 5.2.

## *Add a Format*

To add a format to the list of available formats, click the **Data** tab in the Administration application. Then follow these steps:

1. From the toolbar, select **New Format**.
2. Provide the following information:

   **Format Type**

   Select the radio button for **Character** or **Numeric** data.

**Minimum Format Length**
Enter the minimum length of a label that represents a format value.

**Format Name**
Type a unique name for this format.

**Default Format Length**
Enter the default length of a label that represents a format value.

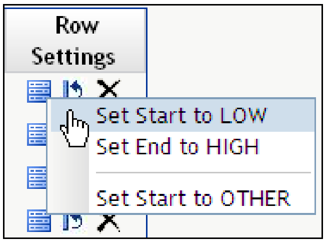
**Maximum Format Length**
Enter the maximum length of a label that represents a format value.

3. Enter the values and labels that define this format:

   **Start**, **End**
   The starting and ending values that describe the range. To exclude the starting value, select the **Exclude Start** check box. To exclude the ending value, select the **Exclude End** check box. (See “About Display Formats” on page 35.)

   **Label**
   Enter a string to display for this range.

4. To insert a keyword in place of a value, click the Rows button. A pop-up menu is displayed.

   Select one of the following keywords:

   - **LOW** represents the lowest value in the data.
   - **HIGH** represents the highest value in the data.
   - **OTHER** represents any value that does not fit into another range, including missing values.

5. To clear the contents of a row, click the Clear value button.
6. To delete a row, click the Delete row button ✕.
7. To add a new row, click **Add Row**.
8. To clear the contents of all rows, click **Clear All**.
9. When you finish defining the format, click **OK**.

The format is written to the HCM database but not to the formats catalog. To add it to the formats catalog, click **Rebuild Formats Catalog** in the toolbar.

## *Assign a Format to a Column*

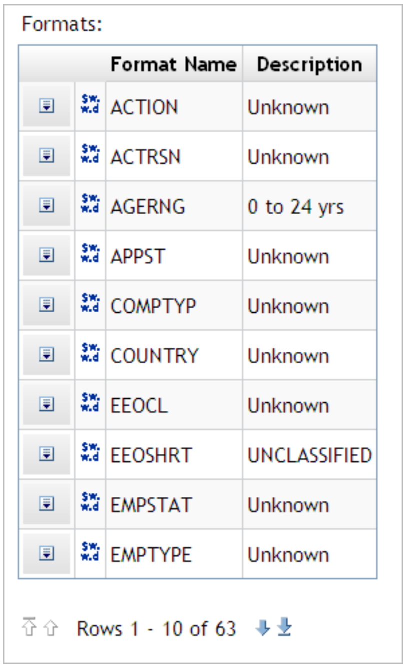
Formats are column attributes. To assign a format, see “Modify Column Attributes” on page 14.

## Manage Formats

To manage formats, click the **Data** tab in the Administration application. Then follow these steps:

1. From the navigation tree, select **Formats**.

   The list of formats is displayed. Use the up and down arrows to page through the list.



Formats:

| Format Name | Description |
|---|---|
| ACTION | Unknown |
| ACTRSN | Unknown |
| AGERNG | 0 to 24 yrs |
| APPST | Unknown |
| COMPTYP | Unknown |
| COUNTRY | Unknown |
| EEOCL | Unknown |
| EEOSHRT | UNCLASSIFIED |
| EMPSTAT | Unknown |
| EMPTYPE | Unknown |

Rows 1 - 10 of 63

   The **Description** field displays the label for the first range of values in the format definition.

2. To delete a format, click the action menu and select **Delete Format**.
3. To view the definition for a format, click the action menu and select **Properties**.
4. Modify the format definition.

   For information about the fields in a format definition, see "Add a Format" on page 36.

5. To save your changes, click **OK**.
6. Update the formats catalog by clicking **Rebuild Formats Catalog**.

## Predefined Formats

### Display Formats

The following table contains predefined display formats in SAS Human Capital Management. For more information, see the SAS_HCM_FORMATS table in the HCM database. The following formats are not currently being used: EEOSHRT, INSTNM, MAJOR, MSAMSA, MSASTATE, RNGEFMT.

*Note:* Do not modify formats in SAS Human Capital Management that are maintained through ETL jobs. Unless otherwise specified, the following display formats are maintained through ETL jobs.

| Format Name | Description |
|---|---|
| ACADEMICCREDIT | Describes credit status for a course. For example, the values "A" and "C" correspond to the labels "Adult Credits" and "Continuing Education". |
| ACADEMICHONORS | Describes an academic honor. For example, "MCL" and "SCL" correspond to "Magna Cum Laude" and "Summa Cum Laude". |
| ACTION | Describes an action. For example, "HIRE" and "PRO" correspond to "New Hire" and "Promotion". |
| ACTRSN | Describes a reason for an action. For example, "New" and "MER" and correspond to "New Position" and "Merit". |
| AGERNG | Describes an age range, such as "0 to 24 yrs" or "65+ yrs". This format is not maintained with an ETL job. |
| APPST | Describes application status. For example, "H" and "REJ" correspond to "Hired" and "Rejected". |
| ATTENDANCESTATUS | Describes attendance status. For example, "A" and "G" correspond to "Attending" and "Graduated". |
| AYN | Contains a yes/no format that is used in retention analysis. |
| COMPTYP | Describes compensation type. For example, "BNUS" and "CMSN" correspond to "Bonus" and "Commission". |
| COUNTRY | Describes the name of a country. For example, "BOL" and "US" correspond to "BOLIVIA" and "UNITED STATES". |
| COURSELEVEL | Describes the level of a course, such as "Remedial" or "General". |
| DEGREECONCENTRATION | Describes a degree concentration. For example, "ACC" and "CSC" correspond to "Accounting" and "Computer Science". |
| DEGREEOPTION | Describes a degree option, such as "Computer Engineering" or "Electrical Engineering". |
| DEGREEPROGRAM | Describes the type of degree program. For example, "BS" and "JD" correspond to "Bachelor of Science" and "Juris Doctor". |
| DEGREETYPE | Describes the type of degree program. For example, "BS" and "JD" correspond to "Bachelor of Science" and "Juris Doctor". |
| EDUVALUESYSTEM | Describes the type of educational value system, such as a 4.0 system for grades or a class rank. |
| EDUVALUETYPE | Describes the type of the educational value system. For example, "CR" and "GPA" correspond to "Class Rank" and "Grade Point Average". |
| EEOCL | Describes an EEO classification. For example, "Tech" and "Prof" correspond to "Technicians" and "Professionals". |
| EMPSTAT | Describes an employee's status. For example, "A" and "M" correspond to "Active" and "Medical". |

| Format Name | Description |
|---|---|
| EMPTYPE | Describes employee type. For example, "RFT" and "RPT" correspond to "Regular Full-Time" and "Regular Part-Time". |
| ENROLLSTATUS | Describes enrollment status. For example, "A" and "G" correspond to "Attending" and "Graduated". |
| ETHNIC | Describes ethnicity. For example, "H" and "W" correspond to "Hispanic" and "Caucasian". |
| EVALRES | Describes an evaluation response, such as "Constantly Exceeds Expectations" or "Seldom Meets Expectations". This format is not maintained with an ETL job. |
| EXEMPT | Describes exempt status. For example, "E" and "N" correspond to "Exempt" and "Non-Exempt". |
| FICE | Describes an educational institution code. For example, "002077" and "002918" correspond to "Johns Hopkins University" and "Davidson College". |
| FLSA | Describes the FLSA status. For example, "N" and "Y" correspond to "Non-Exempt" and "Exempt". |
| GENDER | Describes gender. "F" and "M" correspond to "Female" and "Male". |
| GRADUATINGDEGREE | Describes the type of degree. For example, "C" and "D" correspond to "Certification" and "Degree". |
| GRECTYP | Describes a salary grade. |
| GRP25FM | Describes a manager hierarchy level for the organization. For example, it could describe the highest level manager in an organization, such as "MGR of ACME Computers, Inc.". |
| GRP24FM | Describes a manager hierarchy level for the organization. For example, it could describe the manager levels below the manager level specified by the GRP25FM format, such as "MGR of U.S." or "MGR of Mexico". |
| GRP23FM | Describes a manager hierarchy level for the organization. For example, it could describe the manager levels below the manager levels specified by the GRP24FM format, such as "MGR of Sales" or "MGR of Administration". |
| GRP22FM | Describes a manager hierarchy level for the organization. For example, it could describe the manager levels below the manager levels specified by the GRP23FM format, such as "MGR of Facilities" or "MGR of Payroll". |
| GRP21FM | Describes a manager hierarchy level for the organization. For example, it could describe the manager levels below the manager levels specified by the GRP22FM format, such as "MGR of Public Relations" or "MGR of Benefits". |
| GRP15FM | Describes an organization hierarchy level for the organization. For example, it could describe the highest group level of an organization, such as "ACME Computers, Inc.". |

| Format Name | Description |
|---|---|
| GRP14FM | Describes an organization hierarchy level for the organization. For example, it could describe the group levels below the highest group level of an organization specified in the GRP15FM format, such as "U.S." or "Mexico". |
| GRP13FM | Describes an organization hierarchy level for the organization. For example, it could describe the group levels below the group levels of an organization specified in the GRP14FM format, such as "Sales" or "Administration". |
| GRP12FM | Describes an organization hierarchy level for the organization. For example, it could describe the group levels below the group levels of an organization specified in the GRP13FM format, such as "Facilities" or "Payroll". |
| GRP11FM | Describes an organization hierarchy level for the organization. For example, it could describe the group levels below the group levels of an organization specified in the GRP12FM format, such as "Auditing" or "Benefits". |
| HONORSPROGRAM | Describes an honors program, such as "University Honors" or "University Scholars". |
| INTORG | Describes an institutional organization, such as "Contracts" or "Book Sales". |
| IORGS | Describes an institutional organization, such as "Contracts" or "Book Sales". |
| JOBGRP | Describes a job group, such as "Skilled Laborers" or "Administrative Professional". |
| LANG | Describes a language, such as "English". |
| LOS | Describes a length of service range, such as "1+ to 3 yrs" or "10+ yrs". This format is not maintained with an ETL job. |
| LVCODE | Describes a leave of absence code, such as "Death in Family" or "Jury Duty". |
| MARITAL | Describes marital status, such as "Married" or "Single". |
| MONEY | Describes a currency denomination, such as "Schilling" or "US Dollar". |
| OTHERHONORS | Describes additional honorary designations, such as "Phi Beta Kappa (General Academics)" or "Tau Beta Pi (Engineering Excellence)". |
| PAYPER | Describes a pay schedule, such as "Hourly" or "Yearly". |
| POSTY | Describes a position type, such as "Permanent Full-Time" or "Temporary Part-Time". |
| PSTAT | Describes a position status code, such as "Inactive". |
| RECSRC | Describes a recruitment source code, such as "Chicago Tribune" or "Minneapolis Star". |
| REGTEMP | Describes employment status, such as "Regular Full-Time" or "Temporary Part-Time". |
| REJRSN | Describes a reason for rejection, such as "More Qualified Candidate" or "Not Qualified". |

| Format Name | Description |
|---|---|
| SCHOOLDEPTTYPE | Describes a school department type, such as "Engineering" or "Music". |
| SCHOOLNAMETYPE | Describes the type of school from a governmental perspective, such as "Private School" or "Public School". |
| SCHOOLTYPE | Describes the type of school, such as "High School" or "University". |
| STATE | Describes a state, such as "North Carolina" or "New York". |
| UNION | Describes a union type, such as "Distribution Clerk's Union" or "Payroll Clerks Union". |
| VETERAN | Describes veteran status, such as "Veteran" or "Unknown". |
| VTGROUP | Categorizes predicted termination probabilities (0=low, 1=moderate, 2=high) for retention analysis. |
| YESNO | Describes an answer such as "Yes" or "No". |

### *Internal Formats*

The following table contains predefined internal formats in SAS Human Capital Management. For more information, see the SAS_HCM_FORMATS table in the HCM database. The following formats are not currently being used: IMNSTAT, IEMPTYP, IETHNIC, IGENDER.

| Format Name | Description |
|---|---|
| IACTION | Maps certain personnel actions. If necessary, there can be more than one line mapping to the same keyword. This format is applied to the ACTION_TYPE_CD column. |
| ICHURN | Is required only for the Internal Churn report. The input data values covered by this format are the job action codes that represent an employee voluntarily leaving one position to take another position that is within the same organization, but in a different reporting group. There can be as many lines as necessary mapping to the keyword CHURN. This format is applied to the ACTION_TYPE_CD column. |
| IEEOCL | Maps EEO class codes. This format is applied to the EEO_CLASS_CD column. The internal keywords associated with this format are the EEO classifications that are used in Saratoga Institute data. This format is used for HCM Measures. |
| IEMPSTA | Is used to determine whether an employee is active or inactive. This format is applied to the EMPLOYEE_STATUS_CD column. It is required; at least one employee status code must be included in it. |
| IEXEMPT | Is used to classify employees as exempt or non-exempt according to the United States Fair Labor Standards Act. This format is applied to the EXEMPT_STATUS_CD column. This format is used for HCM Measures. |

| Format Name | Description |
|---|---|
| IONPYRL | Determines whether an employee is currently on the organization’s payroll. This format is applied to the ONPAYRL column. It is used in the computation of certain metrics for which Saratoga benchmarks exist. This format is used for HCM Measures. |
| IPAYPER | Is used to determine an employee’s normal pay period. It is used to calculate an employee’s total annual compensation. It is applied to the EMP_PAID_FREQ_CD column. It is required. |
| IREGTMP | Is used to determine whether an employee is a regular or temporary employee. There can be as many lines as necessary mapping to each keyword. This format is applied to the PERMANENCE_CD column. |
| ISTECLS | Is used to determine the Saratoga employee class. This format is applied to the STECLASS column. It is required in order to compute certain metrics for which Saratoga benchmarks exist. This format is used for HCM Measures. |
| ITERM | Determines the job action codes that indicate that an employee has left the organization, whether voluntarily or involuntarily. This format is applied to the ACTION_TYPE_CD column. |

# *Chapter 3*
# Customizing the Display

**Customizing an Employee Profile** . . . . . . . . . . **45**
Working with Employee Profiles . . . . . . . . . . 45
Assign an Employee Profile to a User . . . . . . . . . . 46
Contents of an Employee Profile . . . . . . . . . . 46
Create an Employee Profile . . . . . . . . . . 48
Customize the Employee List, Profile Header, and Search Criteria . . . . . . . . . . 50
Add Fixed Categories . . . . . . . . . . 51
Copy an Employee Profile . . . . . . . . . . 52
Define an External Action . . . . . . . . . . 52
Select the Default Employee Profile . . . . . . . . . . 55
Select a Default Category to Display for an Employee Profile . . . . . . . . . . 56
Delete a Profile . . . . . . . . . . 56

**Customizing Geographic Analysis** . . . . . . . . . . **56**
About Geographic Analysis . . . . . . . . . . 56
Drilling Down into an Image Map . . . . . . . . . . 56
Modify the Drill Level Hierarchy . . . . . . . . . . 58
Modify the Geographic Analysis Actions . . . . . . . . . . 59
Create Maps for Geographic Analysis . . . . . . . . . . 59

**Customizing Organization Analysis** . . . . . . . . . . **60**
About Organization Analysis . . . . . . . . . . 60
Modify the Organization Analysis Defaults . . . . . . . . . . 60

**Working with Templates** . . . . . . . . . . **61**
About Templates . . . . . . . . . . 61
Select Templates . . . . . . . . . . 61
Add a Template . . . . . . . . . . 62

**Select the General Search Default Columns** . . . . . . . . . . **63**

## Customizing an Employee Profile

### *Working with Employee Profiles*

An employee profile provides a view into the employee data. It determines the tables and columns that are displayed in the Employee Browser when a user views information about a single employee, views a list of employees in an organizational unit, or conducts an employee search.

In the HCM Administration application, you can customize employee profiles in the following ways:

- select the default employee profile
- create a new employee profile
- select or specify the following attributes of an employee profile:
  - default template settings
  - default information table settings
  - default hierarchy settings
  - actions
- select the columns to display in the following areas:
  - the employee list
  - the profile header
  - the search criteria
- select the fixed categories, and the columns for each category, to be displayed in an employee profile
- create and assign a template, which determines how the information is displayed (for example, using tabs or drop-down lists for selection)

In the Employee Browser, users can further customize their profiles (for example, by adding tables to the display, or by selecting the columns to be displayed as the result of a search). However, these customizations affect only the individual user's version of a profile, not the settings that you assign.

## Assign an Employee Profile to a User

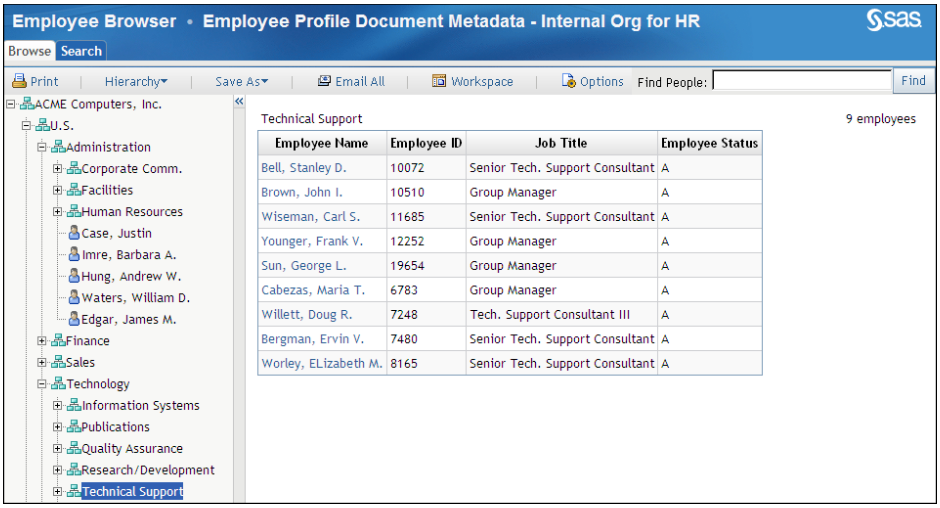
As an HCM Administrator, you assign an employee profile to a user or group by granting them ReadMetadata access to the profile and to the folder that contains the profile.

You might determine that some employees require multiple profiles that display different views of employee information. For example, some profiles might focus on compensation information, while other profiles contain demographic data. In that case, you would grant those users ReadMetadata access to more than one profile. The user can then select a profile from the Options page of the Employee Browser.

## Contents of an Employee Profile

### Employee List

In an employee profile, **Employee List** specifies the default summary columns that are displayed when a user selects a management or departmental node in the hierarchy or displays the results of a search. In this example, the user has selected the **Technical Support** node:



ACME Computers, Inc.
U.S.
Administration
Corporate Comm.
Facilities
Human Resources
Case, Justin
Imre, Barbara A.
Hung, Andrew W.
Waters, William D.
Edgar, James M.
Finance
Sales
Technology
Information Systems
Publications
Quality Assurance
Research/Development
Technical Support

Technical Support

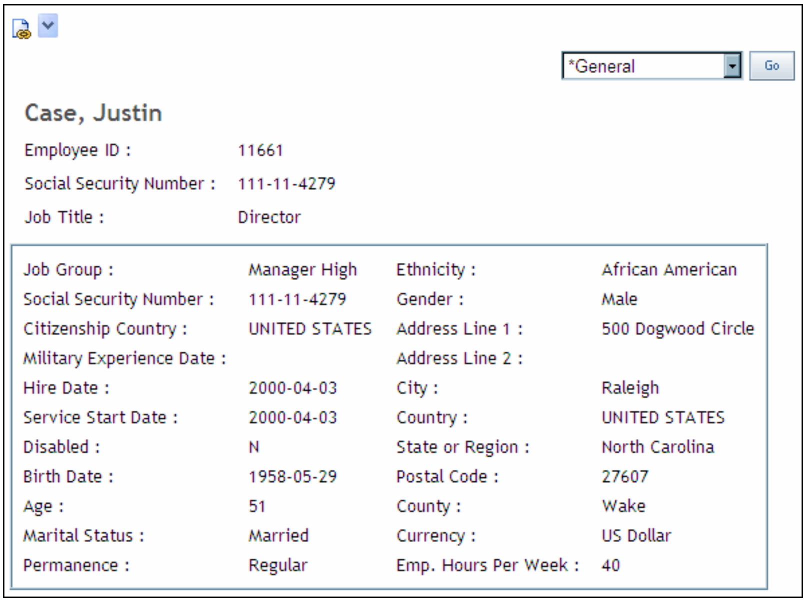
9 employees

| Employee Name | Employee ID | Job Title | Employee Status |
|---|---|---|---|
| Bell, Stanley D. | 10072 | Senior Tech. Support Consultant | A |
| Brown, John I. | 10510 | Group Manager | A |
| Wiseman, Carl S. | 11685 | Senior Tech. Support Consultant | A |
| Younger, Frank V. | 12252 | Group Manager | A |
| Sun, George L. | 19654 | Group Manager | A |
| Cabezas, Maria T. | 6783 | Group Manager | A |
| Willett, Doug R. | 7248 | Tech. Support Consultant III | A |
| Bergman, Ervin V. | 7480 | Senior Tech. Support Consultant | A |
| Worley, ELizabeth M. | 8165 | Senior Tech. Support Consultant | A |

In the Employee Browser, users can customize their own profiles and add or remove columns from the display.

## Profile Header

The **Profile Header** specifies the columns that are displayed in the header area of an employee profile. In this case, the header contains four columns: employee name, employee ID, Social Security number, and job title:



Case, Justin

Employee ID : 11661

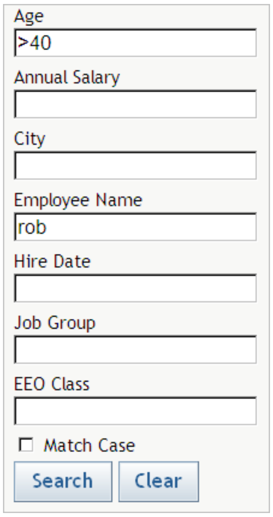
Social Security Number : 111-11-4279

Job Title : Director

| | | | |
|---|---|---|---|
| Job Group : | Manager High | Ethnicity : | African American |
| Social Security Number : | 111-11-4279 | Gender : | Male |
| Citizenship Country : | UNITED STATES | Address Line 1 : | 500 Dogwood Circle |
| Military Experience Date : | | Address Line 2 : | |
| Hire Date : | 2000-04-03 | City : | Raleigh |
| Service Start Date : | 2000-04-03 | Country : | UNITED STATES |
| Disabled : | N | State or Region : | North Carolina |
| Birth Date : | 1958-05-29 | Postal Code : | 27607 |
| Age : | 51 | County : | Wake |
| Marital Status : | Married | Currency : | US Dollar |
| Permanence : | Regular | Emp. Hours Per Week : | 40 |

## Employee Search

**Search** specifies the columns on which a user can search. This example from the Employee Browser shows several search columns:



Age
>40

Annual Salary

City

Employee Name
rob

Hire Date

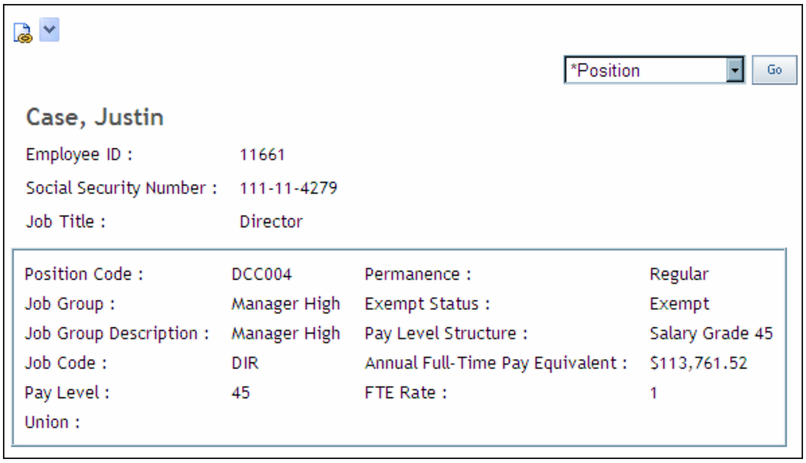
Job Group

EEO Class

□ Match Case

### The Fixed Categories

The fixed categories determine the columns that are displayed when a user views the information for a single employee. A profile must contain at least one fixed category that includes one or more columns. Categories are simply ways of grouping information for display. Some common categories are general information, position information, and compensation. In this example, the user is displaying a category containing information such as the employee's job group, permanence status, and pay level:



Case, Justin

| | |
|---|---|
| Employee ID : | 11661 |
| Social Security Number : | 111-11-4279 |
| Job Title : | Director |

| | | | |
|---|---|---|---|
| Position Code : | DCC004 | Permanence : | Regular |
| Job Group : | Manager High | Exempt Status : | Exempt |
| Job Group Description : | Manager High | Pay Level Structure : | Salary Grade 45 |
| Job Code : | DIR | Annual Full-Time Pay Equivalent : | $113,761.52 |
| Pay Level : | 45 | FTE Rate : | 1 |
| Union : | | | |

Users cannot modify the fixed categories. However, on the Options page of the Employee Browser, they can add other tables or information maps to the display.

## Create an Employee Profile

There are two approaches to creating an employee profile:

- You can copy an existing profile and customize it. (See "Copy an Employee Profile" on page 52.)
- You can create a new profile.

To create a new employee profile:

1. In the HCM Administration application, click the **Customize** tab.
2. Click **New Employee Profile**.

The New Employee Profile dialog box is displayed.

3. Enter a name and description for the profile.

   If you enter a description, it is displayed at the top of the Employee Profile, along with the hierarchy description. If the description is blank, the profile name is displayed instead. (When you edit a profile, you can modify the description, if necessary.)

4. From the **Save in** box, select a location in which to store the profile. The profile appears as a document in the workspace. You can move it or change its permissions if necessary.

   In order to access a profile, a user must have ReadMetadata access to the folder and to the profile itself. You can use this permission to restrict access to particular profiles. See "Assign an Employee Profile to a User" on page 46 for more information.

   Administrators must have ReadMetadata and WriteMetadata access to this folder and its contents.

   *Note:* Do not try to rename an employee profile (in the workspace or in SAS Management Console). Renaming an existing profile would prevent users from being able to access it. In addition, be aware that changing the profile description in the workspace does not affect the description that is displayed in the profile properties on this page.

5. From the **Employee details template** drop-down list, select a template for the profile. The template determines the way the information is displayed in the Employee Browser. For example, categories might be displayed in a drop-down list, or on a set of tabs.

6. Provide the following information in the **Default Information Table Settings** section:

   The **Table** selection determines the information table that is always used by this profile, regardless of which hierarchy is used. Only tables that have a hierarchy mapping are available in the drop-down list. A typical selection is the Employee Master table.

   The **Link attribute** and **Key attribute** are used to index into the information table. The key attribute must contain a unique identifier. In the Employee Browser, when a user clicks the link attribute (such as employee name), the key attribute value (such as employee ID) is used to select the correct record from the table.

7. Provide the following information in the **Default Hierarchy Settings** section:

   The **Employee browser hierarchy** determines the hierarchy that is displayed by default in the Employee Browser. The user can override this default with another hierarchy selection.

   If you select **View employee nodes in hierarchies**, then employees are displayed as separate nodes in the hierarchical tree. Clear the check box to display only nodes that represent entities such as managers, departments, and divisions, rather than individual employees.

   If you select **View all department nodes**, then users can view all nodes, even nodes without employees. (The department might not have any employees, or the user might be restricted from seeing employee information for those nodes.)

8. Click **OK**.

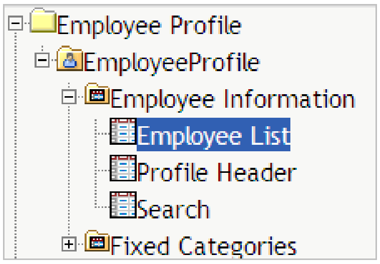
When you create an employee profile, no columns are selected for the profile header, search criteria, or fixed categories. You must edit the profile to assign these columns. Otherwise, no information will be displayed in the Employee Browser for this profile. You can also create actions (links) that can appear at the department level or at the employee detail level.

## *Customize the Employee List, Profile Header, and Search Criteria*

Initially, the employee list includes two required columns: the columns that you specified as key and link attributes for the profile. You can add other columns to the list. The profile header and search criteria have no columns associated with them. In order to view or search employee information, you must add columns to these profile components.

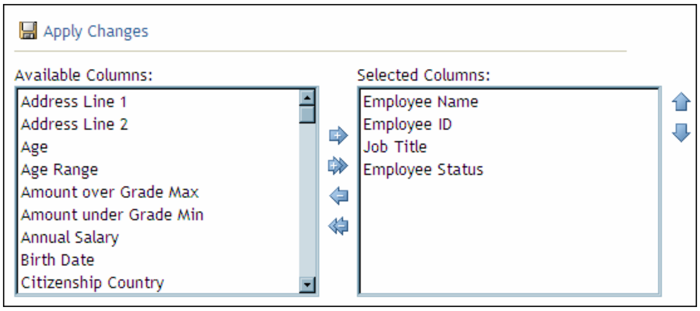
To add columns to a profile component:

1. In the Administration application, select the **Customize** tab.
2. In the navigation tree, find **Employee Profile** ⇨ ***profile-name*** ⇨ **Employee Information** and select the appropriate section: **Employee List**, **Profile Header**, or **Search**.



3. In the display area, select one or more columns for display.

   *Note:* The **Employee List** columns must include the profile's **Key Attribute** and **Link Attribute** columns. For descriptions of these attributes, see “Create an Employee Profile” on page 48.



Available Columns:
Address Line 1
Address Line 2
Age
Age Range
Amount over Grade Max
Amount under Grade Min
Annual Salary
Birth Date
Citizenship Country

Selected Columns:
Employee Name
Employee ID
Job Title
Employee Status

   a. To add columns to the display, select one or more columns from the **Available Columns** list and use the arrows to move them to the **Selected Columns** list.

      To remove columns from the display, move them back to the **Available Columns** list.

   b. To change the order in which the columns appear, use the up and down arrows.

4. Click **Apply Changes**.

   Be sure to apply changes before switching sections. Otherwise, your changes will be lost.

*Note:* As with all tabs of the Administration application, click **Refresh Cache** when you finish your modifications. (See "Refreshing the Cache" on page 7.)

## *Add Fixed Categories*

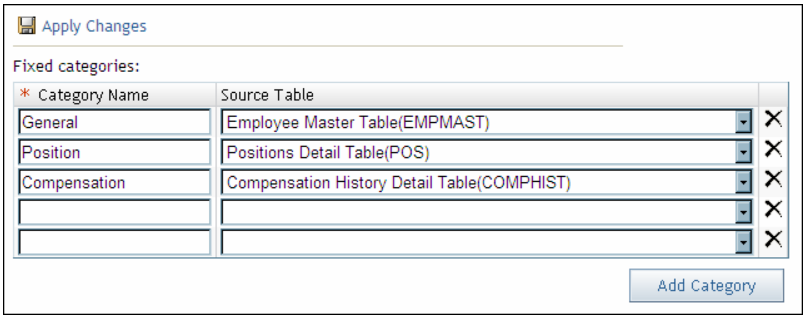
In an employee profile, the fixed categories determine the columns that are displayed when a user views employee information.

*Note:* When you first create an employee profile, it has one fixed category. The name of the default category is specified on the **Customize** tab of the Administration application. There are no columns associated with the default category. You need to add columns, as described below.

To add one or more fixed categories to an employee profile:

1. In the Administration application, click the **Customize** tab.
2. In the navigation tree, select **Employee Profile** ⇨ ***profile-name*** ⇨ **Fixed Categories**.
3. In the **Category Name** box, enter a name that describes the category.
4. From the **Source Table** drop-down list, select a table to be the source for the columns in this category.

   In this example, the HCM Administrator has added three fixed categories, using columns from three different source tables:



Apply Changes

Fixed categories:

| * Category Name | Source Table | |
|---|---|---|
| General | Employee Master Table(EMPMAST) | ✕ |
| Position | Positions Detail Table(POS) | ✕ |
| Compensation | Compensation History Detail Table(COMPHIST) | ✕ |
| | | ✕ |
| | | ✕ |

Add Category

5. To add more rows to the list, click **Add Category**.
6. To delete a category, click the Delete icon ✕ in the column for that category.
7. Click **Apply Changes**.

The category is added to the profile. However, it has no columns associated with it yet. To add columns to a category:

1. In the navigation tree, select **Employee Profile** ⇨ ***profile-name*** ⇨ **Fixed Categories** ⇨ ***category-name***.
2. Follow the instructions in "Customize the Employee List, Profile Header, and Search Criteria" on page 50 to add or remove columns from the category.

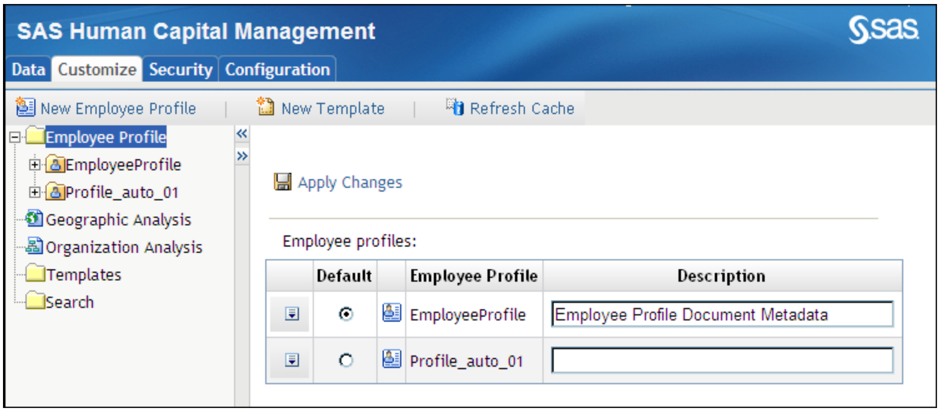
*Note:* On the Options page of the Employee Browser, users can add additional categories to their personal profiles. If you do not want some users to customize their profiles, you can use object security to deny access to the Options page. For more information, see "Securing Objects" on page 67.

## Copy an Employee Profile

Instead of creating a profile from scratch, you can copy an existing employee profile and customize it for your purposes, as follows:

1. In the navigation tree, select **Employee Profile**.

   The list of available profiles is displayed.



Employee profiles:

| | Default | Employee Profile | Description |
|---|---|---|---|
| ▾ | ● | EmployeeProfile | Employee Profile Document Metadata |
| ▾ | ○ | Profile_auto_01 | |

2. Click the action menu ▾ at the left of a profile and select **Copy**.
3. On the Copy dialog box, type a name and description for the new profile. (The description is optional.)
4. Select a folder in which to save the new profile, and click **Copy**.

## Define an External Action

In the Employee Browser or a geographic analysis, an action is a link to an external object such as a JSP page, an HTML page, or a stored process. When the user clicks the link, the action is performed (for example, the page is displayed or the stored process is executed). You can pass parameters to the action (such as employee ID or department code) to customize the user experience. By default, the session ID is also passed on the link query string, so that the target page or application is aware of the user who is making the request and can apply the necessary filters.

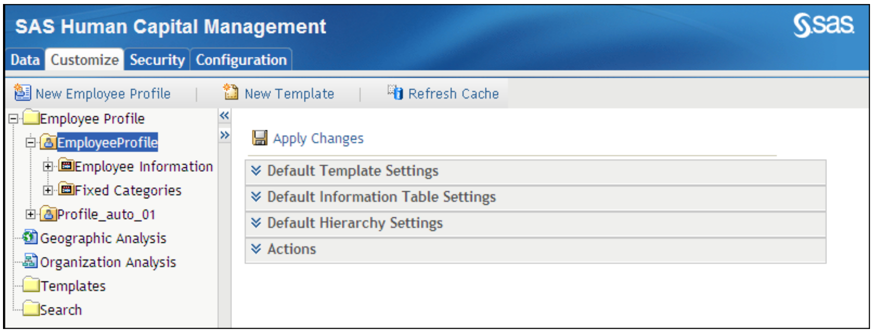
In an employee profile, you can define multiple actions, and you can specify whether they are available at the employee level, at the department level, or both.

For geographic analysis, you can define multiple actions that appear in map view mode for all geographic analyses.

Follow these steps to define an action:

1. In the Administration application, click the **Customize** tab.
2. In the navigation tree, select **Employee Profile** ⇨ ***employee-profile***.

   The matching profile is displayed:



3. Click » **Actions** to expand the **Actions** section.

4. Click **New**.

5. In the **Action Name** box, type a name for the action.

6. From the **Type** drop-down list, select the type of object that is the target of the hyperlink:

   - **External Link (JSP/HTML)**: a JavaServer page, HTML page, or other valid MIME type such as an image or a PDF file. In the Employee Browser or in a geographic analysis, clicking the link opens the target page in a separate browser window.

   - **Information Map**: a SAS Information Map. Clicking the link opens SAS Web Report Studio, with this map selected as the data source.

   - **Web Report**: a SAS Web Report Studio report. Clicking the link opens this report in SAS Web Report Studio.

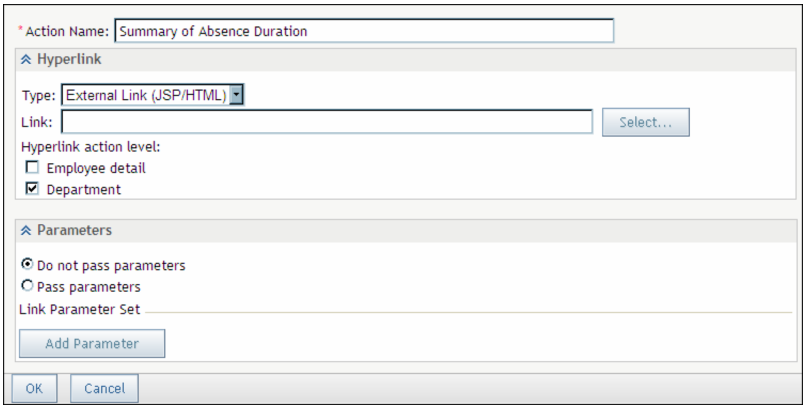
   - **Web Service**: a Web service such as Google.

   - **SAS Stored Process**: a stored process. Clicking the link causes the stored process to be executed.

   - **BI Dashboard**: a SAS BI dashboard object. Clicking the link opens the dashboard.

7. If the **Select** button is active, click **Select** to browse the **SAS Folders** for an information map, SAS Web Report Studio report, or stored process, or to select a dashboard.

   Otherwise, type in the complete path (including **http://**) to an HTML page, JSP, or Web service.


*Action Name: Summary of Absence Duration

Hyperlink

Type: External Link (JSP/HTML)

Link:

Hyperlink action level:

□ Employee detail

☑ Department

Parameters

● Do not pass parameters

○ Pass parameters

Link Parameter Set

By default, HCM information maps are stored at **Products ⇨ SAS Human Capital Management ⇨ Data Sources ⇨ Information Maps**. The SAS Human Capital

Management standard reports are stored at **Products** ⇨ **SAS Human Capital Management** ⇨ **Reports**.

8. Select the levels at which the action is available:

   - **Employee detail**
   - **Department**

   *Note:* These check boxes do not apply to geographic analysis.

9. If you do not want to pass any parameters on the link string, select **Do not pass parameters**.

10. To define parameters to be passed on the link query string, select **Pass parameters**.

    If you choose this option, the **Add Parameter** button is enabled.

11. (Optional) To add one or more parameters to an action, first select a source table from the **Table** drop-down list. The source table applies to all parameters for this action.

    Then follow these steps:

    a. Click **Add Parameter**.

    b. In the **Name** box, type a name for the parameter.

    c. From the **Column** drop-down list, select a column to supply the parameter value.

       When a user selects the action, the parameter value to match the currently selected department or employee is retrieved from the table.

    d. To delete a parameter, click the Delete button × next to the parameter.

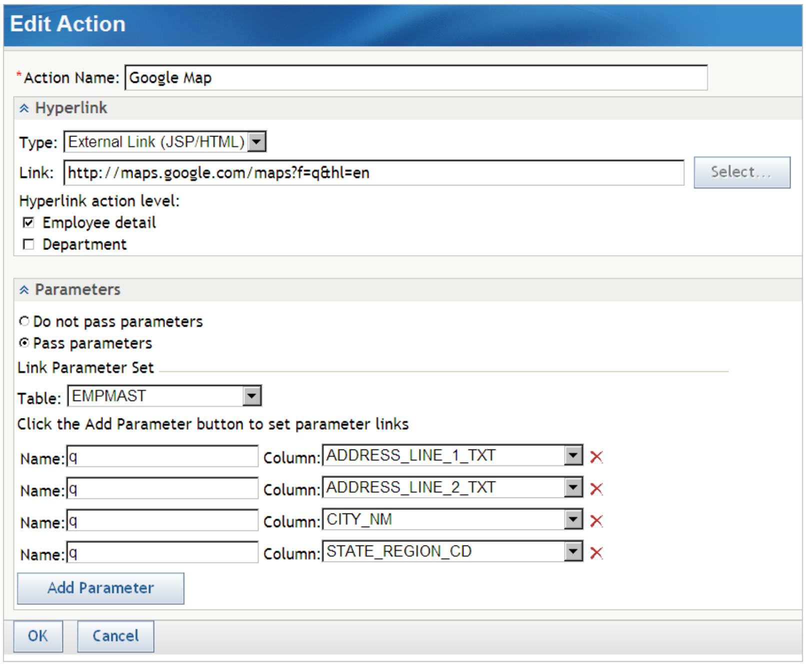
    e. When you have finished adding parameters, click **OK**.

12. To apply an action, move it from the **Available actions** list to the **Selected actions** list.

13. After you have completed your modifications, click **Apply Changes**.

    *Note:* If selected actions are not saved when you click **Apply Changes**, click the selected actions to highlight them before clicking **Apply Changes**.

Here is an example of an action that is a link to Google maps. The selected employee's address is passed in the parameters that Google maps requires:



Edit Action

*Action Name: Google Map

Hyperlink

Type: External Link (JSP/HTML)

Link: http://maps.google.com/maps?f=q&hl=en

Hyperlink action level:

☑ Employee detail

□ Department

Parameters

○ Do not pass parameters

● Pass parameters

Link Parameter Set

Table: EMPMAST

Click the Add Parameter button to set parameter links

| Name | Column |
|---|---|
| q | ADDRESS_LINE_1_TXT |
| q | ADDRESS_LINE_2_TXT |
| q | CITY_NM |
| q | STATE_REGION_CD |

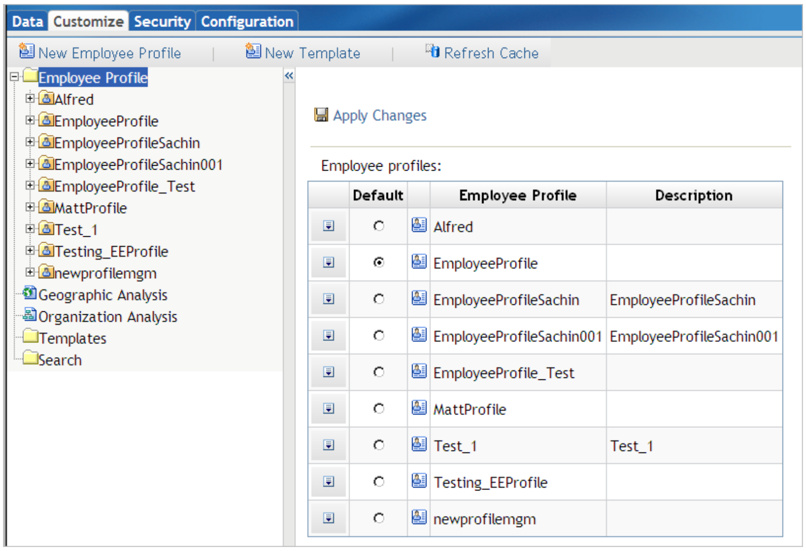
## Select the Default Employee Profile

The default employee profile is displayed when a user first opens the Employee Browser. The user can then select a different profile (if other profiles are available) on the Options page of the Employee Browser.

To select the default employee profile:

1. In the Administration application, select the **Customize** tab.
2. From the navigation tree, select **Employee Profile**.

   The list of available profiles appears.



Employee profiles:

| Default | Employee Profile | Description |
|---|---|---|
| ○ | Alfred | |
| ● | EmployeeProfile | |
| ○ | EmployeeProfileSachin | EmployeeProfileSachin |
| ○ | EmployeeProfileSachin001 | EmployeeProfileSachin001 |
| ○ | EmployeeProfile_Test | |
| ○ | MattProfile | |
| ○ | Test_1 | Test_1 |
| ○ | Testing_EEProfile | |
| ○ | newprofilemgm | |

3. Select the radio button for a profile under the **Default** column.
4. Click **Apply Changes**.

### Select a Default Category to Display for an Employee Profile

To specify a default category to display in the Employee Browser, perform the following steps in the Administration application:

1. Select the **Customize** tab.
2. Select **Employee Profile** ⇨ ***Employee Profile-name*** ⇨ **Fixed Categories**.
3. Select a category from the **Default Category** drop-down menu.

*Note:* After applying these changes and refreshing the cache, you need to log off and log in again to view this change.

### Delete a Profile

To delete an employee profile, click the action menu ▾ to the left of a profile and select **Delete**.

## Customizing Geographic Analysis

### About Geographic Analysis

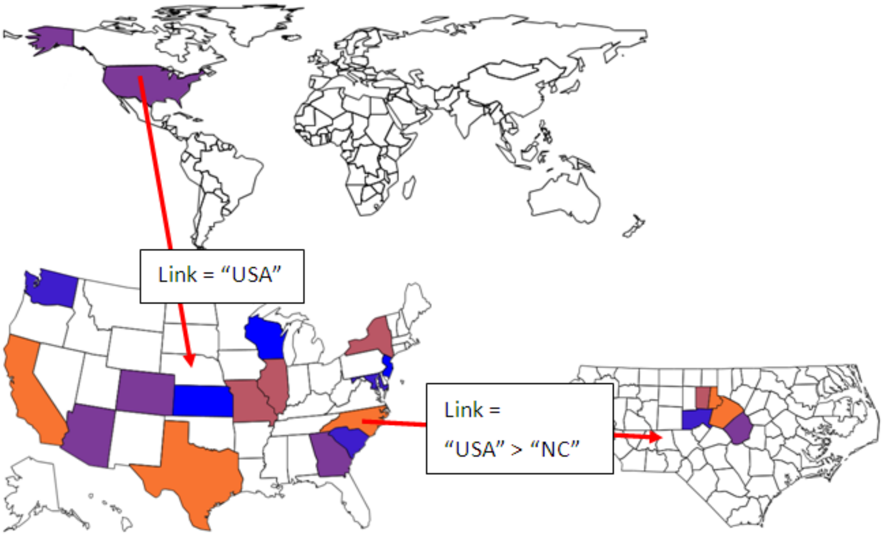
A geographic analysis displays employee information that is associated with an image map. Colored regions on the map indicate areas where employees are located. A legend on the display associates each color with a range of values that represent the number of employees. As the mouse pointer hovers over an area, a set of statistics is displayed for that area.

At any mapping level, the user can switch to a table view of the data, search the data, or print the map or table. A user can also modify the analysis options, such as the colors used and the data that is displayed at each level.

As an HCM Administrator, you define the drill level hierarchy and the actions that are available at each level.

### Drilling Down into an Image Map

The image maps that are used in geographic analysis have link fields that allow a user to drill down to the next level. For example, here are three maps that are available in the default installation—a map of the world, a map of the U.S., and state map—showing the link field values that were used to drill down to the next map:



In this example, the user clicked the outline of the United States on the world map to drill down to the USA country map. On the country map, the user clicked the outline of North Carolina to drill down to the map for that state.

A geographic analysis requires two types of tables:

- map tables
- an information table

Map tables contain data for displaying image maps and drilling down to more detailed maps. As the HCM administrator, you define a hierarchy of map tables. When users create a geographic analysis, they select the entire hierarchy or a subset of the hierarchy of maps to display.

Each map table contains a SEGMENT column and X and Y columns that define areas of the map. A map table also contains link columns that are used to link from one map to the next, and from the map to an information table.

These map tables are supplied by default:

***Table 3.1*** *Default Map Tables*

| Level | Map Table | Link Columns |
|---|---|---|
| World | HRV_GBL | COUNTRY_CD |
| Country | HRV_US | COUNTRY_CD<br>STATE_REGION_CD |
| State | HRV_STATE | COUNTRY_CD<br>STATE_REGION_CD<br>COUNTY_NM |

Notice that the link columns are cumulative. Each lower level in the hierarchy must contain all the link columns from the previous levels. The map for the third level in the example (HRV_STATE) contains a column that represents counties within a state. That column is used for statistics display even if it is not used to drill down to a lower-level map.

An information table contains the data that is displayed in a geographic analysis, whether in map view or table view. When users create a geographic analysis, they select an information table, which must be a master table or a history table. It must contain columns that correspond to the map link columns.

## Modify the Drill Level Hierarchy

The **Drill Level Hierarchy** section specifies default levels that are available for drilling down into a geographic analysis. If a user creates a geographic analysis and chooses to begin with Level 2, for example, the default values for Level 1 are used.

*Note:* Modifying the drill level hierarchy can affect existing geographic analyses.

1. In the HCM Administration application, click the **Customize** tab.
2. Select **Geographic Analysis**.
3. Open the **Drill Level Hierarchy** section.

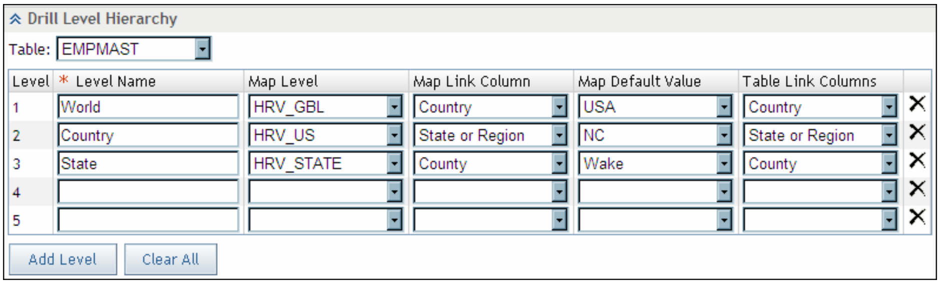


Table: EMPMAST

| Level | * Level Name | Map Level | Map Link Column | Map Default Value | Table Link Columns | |
|---|---|---|---|---|---|---|
| 1 | World | HRV_GBL | Country | USA | Country | ✕ |
| 2 | Country | HRV_US | State or Region | NC | State or Region | ✕ |
| 3 | State | HRV_STATE | County | Wake | County | ✕ |
| 4 | | | | | | ✕ |
| 5 | | | | | | ✕ |

Each row represents a level in the hierarchy of image maps. The first level is the highest—in this case, the world map.

4. From the **Table** drop-down list, select a master table to serve as the default in a geographic analysis.

   The default information table must contain the same columns as the link columns in each of the maps that are defined for a hierarchy. For example, the EMPMAST table in the sample data contains country, state, and county columns, each of which corresponds to the link columns in the default maps.

   When users create a geographic analysis, they can select a different table, which might not contain all the table link columns that are defined in the drill level hierarchy. In that case, the default information table acts as a backup. For example, if the selected information table had no COUNTY_NM column (and COUNTY_NM was one of the table link columns), the analysis would use the COUNTY_NM column from the default table. For best performance, however, each master or history table should contain all the table link columns.

5. Complete the remaining fields as follows:

   **Level Name**
   Enter a name to identify this level. This name is used in the bread crumbs at the top of a geographic analysis display.

   **Map Level**
   From the drop-down list, select the map table for this level.

   **Map Link Column**
   From the drop-down list, select the link column in the map table.

**Map Default Value**
From the drop-down list, select the default value for the map link column (in other words, the default drill-down value). This value is applied in a geographic analysis that starts with a level other than Level 1.

**Table Link Columns**
From the drop-down list, select the column (in the master table) that corresponds to the map link column. For example, if the map link column contains state codes (such as **NC**), the table link column must contain state codes. If the map link column contains formatted values (such as **North Carolina**), then the table link column must also contain formatted values.

*Note:* If the map link column and table link column do not match, a geographic analysis displays an empty map, with no statistics or colored regions. However, the legend displays the employee population, to match what is displayed in Table View mode.

6. To create another level in the hierarchy, click **Add Level**.
7. To remove all levels from the hierarchy, click **Clear All**.
8. Click **Apply Changes**.

## *Modify the Geographic Analysis Actions*

In addition to setting the drill level hierarchy, the HCM Administrator can define external links, called actions, that are available to users in a geographic analysis. For instructions, see “Define an External Action” on page 52.

## *Create Maps for Geographic Analysis*

The default map tables are listed in Table 3.1 on page 57. You can import additional map tables if necessary. For example, to add a map of Poland, you would follow a procedure like this:

1. Insert COUNTRY_CODE(POL) and its corresponding SEGMENT, X and Y into the world map table.

   *Note:* The country code will be the map link column for a geographic analysis. This value must match the table link column in the information table. Typically, they are both codes that represent a country, but if one is a formatted value, the other must be also.

2. Insert COUNTRY_CODE, PROVINCE_CODE and its corresponding SEGMENT, X and Y into the Country map table.
3. Insert COUNTRY_CODE, PROVINCE_CODE, DISTRICT_CODE and its corresponding SEGMENT, X, and Y into a Province map table.
4. Add employee records to the information table, which must contain COUNTRY_CODE, PROVINCE_CODE, and DISTRICT_CODE values for each employee added.
5. Import the map files with the following steps.
   a. On the **Data** tab, select **New Table**.
   b. Select the **External** radio button.
   c. From the **Type** drop-down list, select **SAS DataSet**.

   d. From the **General Settings**, select the following options:

      - **Create a map table**
      - **Use format**
      - **Allow security access for HCM roles**

      Do not select these options:

      - **Use as a master table**
      - **Register table in metadata repository**

   For further instructions about importing tables, see “Add a Table” on page 11. If your SAS data set contains columns with formats attached to them, see “Importing SAS Data Sets with Formatted Columns” on page 10.

6. Refresh the cache.
7. On the **Customize** tab, modify the **Drill Level Hierarchy** section appropriately. (See “Modify the Drill Level Hierarchy” on page 58.)
8. Apply your changes and refresh the cache.

# Customizing Organization Analysis

## *About Organization Analysis*

In an organization analysis, users can view the structure of an organization in a hierarchical table (analysis view) or a graphical organization chart (presentation view). An organization analysis displays employee statistics and reporting relationships. Users can simulate an organization restructuring, creating what-if scenarios and exporting the results to a Microsoft Excel worksheet.

## *Modify the Organization Analysis Defaults*

The default attribute determines the text that is displayed in the navigation tree for a new organization analysis. Users can modify this attribute on the Options page of an analysis.

To modify the default attribute:

1. In the SAS Human Capital Management Administration application, click the **Customize** tab.
2. In the navigation tree, click **Organization Analysis**.
3. In the list of attributes, select a radio button:
   - Select **Name** to display the name of the manager or department.
   - Select **Description** to display the description of the manager or department.
   - Select **Manager Name** to display the hierarchy code for this member.
4. Click **Apply Changes**.

# Working with Templates

## About Templates

On the **Customize** tab of the Administration application, you can select templates that apply to the Home page and templates that apply to employee profiles.

The Home page template determines what a user sees in the display area after logging on to SAS Human Capital Management. There are two Home page templates:

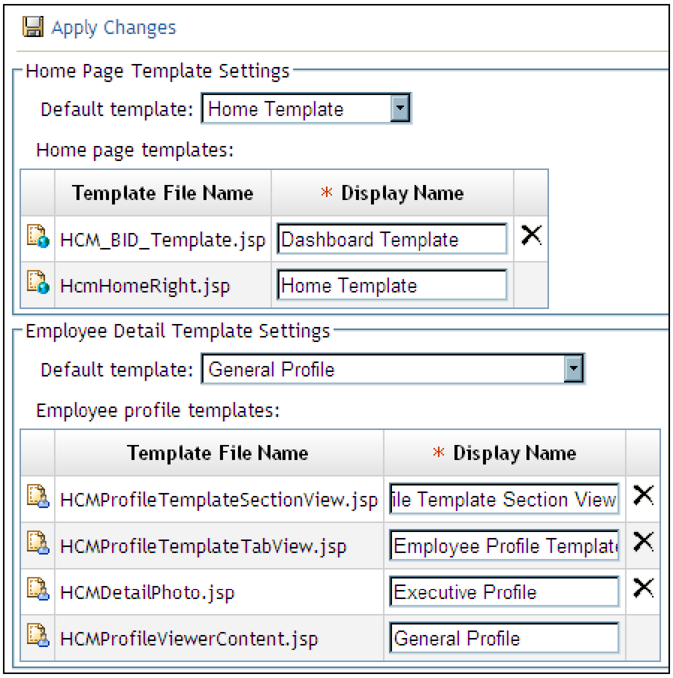
- HcmHomeRight.jsp displays the general search user interface (see “The General Search Utility” in the *SAS Human Capital Management: User's Guide*).
- HCM_BID_Template.jsp displays the general search user interface and a BI Dashboard portlet, from which users can select among available BI dashboards (see “Displaying Key Metrics with SAS BI Dashboard” in the *SAS Human Capital Management: User's Guide*).

The employee profile templates affect the display of employee detail information. They do not affect employee summaries or search results. For example, the employee display area might contain drop-down lists for selecting categories of data (such as compensation or absence information), or it might present that information in a set of tabs. It might display a picture of the employee.

Both the Home page and the employee profile templates are stored as JSP files in the deployed SAS Human Capital Management application, along with the standard JSPs. The employee profile templates use the HCM Public API to access HCM data, and they can be customized. For information about the API and an example, see “Customizing the Employee Profile Templates” on page 101.

## Select Templates

To select a template, click the **Customize** tab of the Administration application. From the navigation tree, select **Templates**. Then follow these steps:



Apply Changes

Home Page Template Settings

Default template: Home Template

Home page templates:

| | Template File Name | * Display Name | |
|---|---|---|---|
| | HCM_BID_Template.jsp | Dashboard Template | ✕ |
| | HcmHomeRight.jsp | Home Template | |

Employee Detail Template Settings

Default template: General Profile

Employee profile templates:

| | Template File Name | * Display Name | |
|---|---|---|---|
| | HCMProfileTemplateSectionView.jsp | ile Template Section View | ✕ |
| | HCMProfileTemplateTabView.jsp | Employee Profile Templat | ✕ |
| | HCMDetailPhoto.jsp | Executive Profile | ✕ |
| | HCMProfileViewerContent.jsp | General Profile | |

1. **Home Page Template Settings**. From the **Default template** drop-down list in this section, select a predefined template.
2. **Employee Detail Template Settings**. From the **Default template** drop-down list in this section, select a predefined template.

   When an administrator creates an employee profile, this template is presented as the default.
3. Click **Apply Changes**.

On this page, you can also modify the display name for a template or delete a template, as follows:

- To modify the display name for a template, type a new name in the **Display Name** box, and click **Apply Changes**.
- To delete a template, click the Delete button ✕ to the right of the template.

  *Note:* You cannot delete the template that is currently selected.

### *Add a Template*

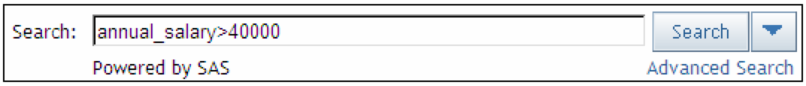
To add a template to the list of predefined templates, click **New Template**. In the New Templates dialog box, follow these steps for each template that you want to define:

1. Enter a display name for the template.
2. (Optional) Enter a description.
3. Enter a filename for the template, including the .jsp suffix.

   *Note:* The filename can contain only alphanumeric characters, underscores, periods, and the forward slash (/).
4. Select the template type: **Home Page Template** or **Employee Profile Template**.
5. Click **OK**.

# Select the General Search Default Columns

The general search is available on the home page of SAS Human Capital Management. The general search performs a query on the default search table (see "Application Properties" on page 96).



You can designate certain columns as default search columns. For those columns, users do not need to include the search column name in the query string. They can simply enter the values. For example, if employee name is a default search column, a user could simply type **`smith`** to search for all employees with "smith" as part of the name.

On the **Customize** tab of the Administration application, select **Search** from the navigation tree. To designate a default search column:

1. Move the column from the **Available Columns** list to the **Selected Columns** list.
2. Click **Apply Changes**.

*Note:* We recommend that you select only a few default search columns that are not easily confused, such as employee name, employee ID, hire date, and age or annual salary.

# *Chapter 4*
# Securing Objects and Tables

**Security in SAS Human Capital Management** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **66**
Managing Security in the Administration Application . . . . . . . . . . . . . . . . . . . . . . 66
Additional Security Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

**Importing Users** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **66**

**Securing Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **67**
About Object Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
How Object Permissions Are Interpreted . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
Add Permissions for an Object . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
Modify Direct Permissions for an Object . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
Secure a Custom Object . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72

**Searching for Identities** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **73**

**Securing Table Rows** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **74**
About Row-Level Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
Default Row-Level Filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
How Row-Level Filters Are Applied . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
Enable Row-Level Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
Add a Row-Level Filter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
Modify a Row-Level Filter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
Delete a Row-Level Filter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

**Hierarchical Filters** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **80**
About Hierarchical Filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
Select a Hierarchy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
How a Hierarchical Filter Is Applied . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
Tables Without Hierarchical Filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
The Power User Group . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

**Securing Table Columns** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **83**
About Column Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
How Column Permissions Are Applied . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
Modify Permissions for a Table Column . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
Add an Identity to Table Column Permissions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85

# Security in SAS Human Capital Management

### *Managing Security in the Administration Application*

On the **Security** tab of the Administration application, HCM administrators can manage these elements of security:

- **Object security.** Assign permissions for actions, such as creating a geographic analysis or opening the Employee Browser.
- **Row-level security.** Assign permissions for access to rows in a table. Row-level security can include filters that are associated with users, groups, or roles. It always includes filters that are associated with a hierarchy.
- **Column security.** Assign permissions for access to table columns.

*Note:* Any changes that you make to object security or row-level security are not effective until you click **Refresh Cache** or restart the managed server. Column security changes are effective immediately (because they are changes to the metadata repository, whereas object security changes and row-level security changes affect the HCM database).

### *Additional Security Measures*

In addition to administering security for HCM objects and tables, you will need to administer security for the folders that hold content in SAS Human Capital Management. For example, you might need to create folders in which users can share content, and you might need to restrict some folders to certain users or groups.

- Administrators, and users with the appropriate permissions, can modify document and folder properties in the workspace.

  Chapter 3 of the *SAS Human Capital Management: User's Guide* describes how the workspace is organized and explains how to set permissions.
- If you have administrative privileges for SAS Management Console, you can create folders and modify permissions via that mechanism.

The primary source for information about these permissions is the *SAS Intelligence Platform: Security Administration Guide*. The book contains an in-depth look at SAS security features, including authentication and authorization. It also contains information about managing security for objects such as SAS repositories, libraries, and OLAP data.

The *SAS Intelligence Platform: System Administration Guide* contains information about implementing operating-system protection at a site.

Both books are available at `support.sas.com/92administration`.

# Importing Users

In order to log on to SAS Human Capital Management, a user must meet the following criteria:

- The user must belong to one of the three SAS Human Capital Management roles: HCM User, HCM Analyst, or HCM Administrator.

  *Note:* Users should belong to only one of these roles. If they belong to more than one, the role with the fewest privileges applies.

- The user must be a member of the HCM Solution Users group.
- The user must have a valid entry in the SAS_USER_EMPLOYEE table.

The SAS_USER_EMPLOYEE table associates user names that are defined in the metadata repository with employee IDs for the employee population that you are analyzing with SAS Human Capital Management. This table, which is located in the HCM Data Mart, is part of the security structure that restricts each user's view to an appropriate subset of employees.

The SAS_USER_EMPLOYEE table gets some of its values from the EMPGEN table, and there are ETL jobs to load both the EMPGEN table and the SAS_USER_EMPLOYEE table. Typically, a site runs both jobs on a regular schedule.

However, there are two occasions under which you might want to update the SAS_USER_EMPLOYEE table without waiting for the ETL jobs to be run:

- You add a user (who is already represented in the EMPGEN table) to the HCM Solution Users group, and you want the effects to be available immediately.
- You remove a user from the HCM Solution Users group, and you want the deletion to be effective immediately.

In those situations, click **Import Users** on the **Security** tab of the Administration application.

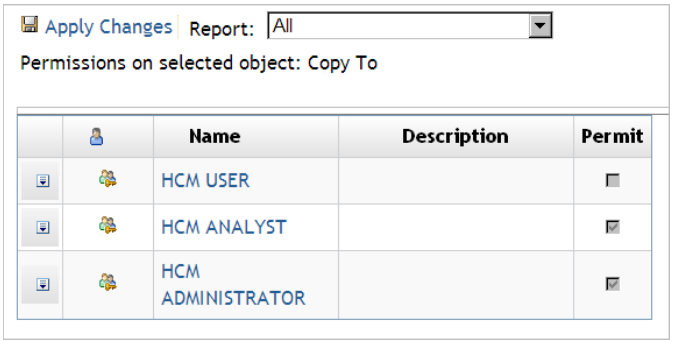
Import Users loads the SAS_USER_EMPLOYEE table with the user ID and employee ID of all active employees who are members of the HCM Solution Users group and who also have entries in the EMPGEN table. It does not reflect changes in employee status (from active to inactive, or vice versa). In addition, it does not reflect new employees who have not yet been added to the EMPGEN table. In those cases, you must run the ETL jobs that rebuild or refresh the HCM Data Mart.

## Securing Objects

### *About Object Security*

Object security refers to actions such as exporting an employee summary to Microsoft Excel, refreshing a cube, or creating an organization chart. If a user has permission for an object, the user can perform that action, and the associated menu item is displayed in SAS Human Capital Management.

For example, consider the **Copy To** feature in an organization analysis. Users with the HCM Administrator role or the HCM Analyst role have permission to perform this action. Users with the HCM User role do not. Assuming no user or group permissions apply, the permissions for the **Copy To** object would look like this:



Apply Changes Report: All

Permissions on selected object: Copy To

| | | Name | Description | Permit |
|---|---|---|---|---|
| | | HCM USER | | ☐ |
| | | HCM ANALYST | | ☑ |
| | | HCM ADMINISTRATOR | | ☑ |

Notice that the **Permit** boxes are dimmed, because the HCM role permissions cannot be changed.

In an organization analysis, a user with the HCM Administrator or HCM Analyst role would see the **Copy To** menu item in the toolbar. A user with the HCM User role would not.

When the object is an application or an external URL, object security means only that the link does not appear in SAS Human Capital Management. A user who knows the URL might still be able to access the application directly. In particular, within the same browser session, a user might be able to take advantage of browser caching. Users should be reminded that when they logoff, they should close the browser.

In addition, there are no hierarchical relationships in object security. For example, if users are denied access to the Administration application but permitted access to a specific action such as adding a filter, a user who knows the URL of the filter action can still perform the task.

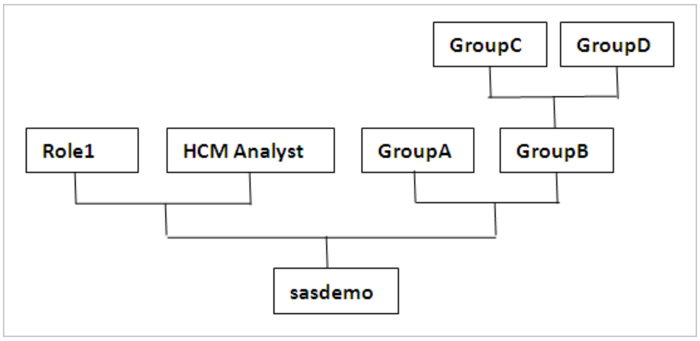
*Note:* Object security should not be considered equivalent to data security. Use object security to restrict the actions that a user can perform and the menu items that are available to a user. Use data security, such as row-level security or column security, to restrict the data that a user can access.

For descriptions of the objects that are included in object security, see "About Object Security" on page 673.

## How Object Permissions Are Interpreted

In thinking about object permissions, it is easiest to use an example of an inheritance tree. For any object, a user might have direct permission (a grant or a denial), and the user might also inherit permissions from one or more groups and from one or more roles. The following figure shows a situation in which a user belongs to multiple groups and roles.

***Figure 4.1*** *Example Inheritance Tree for Object Permissions*



In this example, the SAS Demo User belongs to two groups and two roles (one is a default role, HCM Analyst). GroupB in turn belongs to two groups. Some or all of these identities might have permissions for a particular object. Permissions are decided in this way:

1. The direct permissions (if any) for GroupC and GroupD are merged, becoming the inherited group permission for GroupB. For each merge of groups at the same level, the most restrictive permission applies. For example, if GroupC is granted permission for an object, but GroupD is denied permission, the denial takes precedence.

   *Note:* Because GroupC and GroupD have no groups above them, only directly assigned permissions are considered.

2. The GroupB permission is determined by looking at the direct permission for GroupB (if any) along with the permission that it inherited from GroupC and GroupD. Precedence is applied as follows:

   - If GroupB has a direct permission, that permission is applied (whether it is a grant or denial).

   - Otherwise, if GroupB has an inherited group permission, that permission is applied (whether it is a grant or a denial).

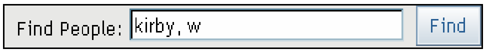
   The result becomes the GroupB permission.

3. The GroupB permission is merged with the GroupA permission to become the inherited group permission for sasdemo.

4. The direct permissions for Role1 and HCM Analyst are merged, becoming the inherited role permissions for sasdemo. Again, the most restrictive permission applies.

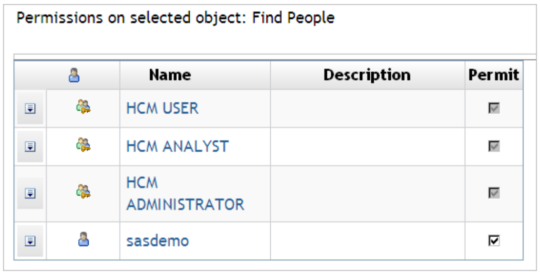
5. The sasdemo permission is determined by comparing its direct permission (if any) with its inherited group and role permissions. Precedence is applied as follows: direct permission, followed by inherited group permission, followed by inherited role permission.

Here are three examples of applying security to such an inheritance tree, using the **Find People** object. **Find People** provides a quick way to find employees or departments in the Employee Browser. It looks like this:



Find People: kirby, w Find

- In the first example, the sasdemo user is granted permission for the **Find People** object.

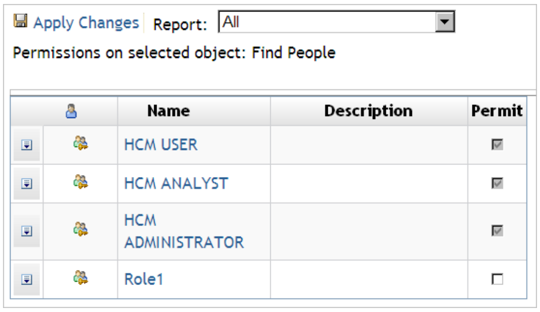
  

  Permissions on selected object: Find People

  | | | Name | Description | Permit |
  |---|---|---|---|---|
  | | | HCM USER | | ☑ |
  | | | HCM ANALYST | | ☑ |
  | | | HCM ADMINISTRATOR | | ☑ |
  | | | sasdemo | | ☑ |

  Because the user was assigned a direct permission for the object, that permission is applied and all other permissions are ignored. As a result, the **Find People** box appears in the Employee Browser.

- In the second example, sasdemo has no direct permissions. Instead, Role1 is denied permission for the object, and HCM Analyst is granted permission.



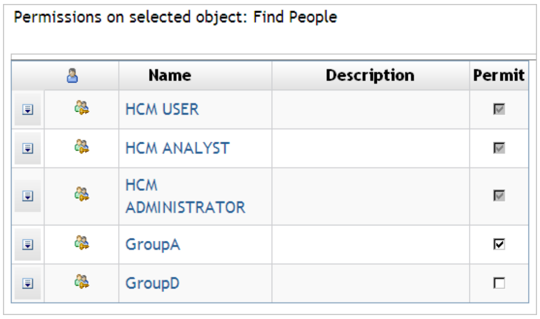
Apply Changes Report: All

Permissions on selected object: Find People

| | | Name | Description | Permit |
|---|---|---|---|---|
| | | HCM USER | | ☑ |
| | | HCM ANALYST | | ☑ |
| | | HCM ADMINISTRATOR | | ☑ |
| | | Role1 | | ☐ |

When the two role permissions are merged, the denial takes precedence. As a result, sasdemo is denied use of **Find People**.

- In the last example, GroupA is directly granted permission for the object, and GroupD is directly denied permission. The HCM Analyst role is also directly granted permission for the object.



Permissions on selected object: Find People

| | | Name | Description | Permit |
|---|---|---|---|---|
| | | HCM USER | | ☑ |
| | | HCM ANALYST | | ☑ |
| | | HCM ADMINISTRATOR | | ☑ |
| | | GroupA | | ☑ |
| | | GroupD | | ☐ |

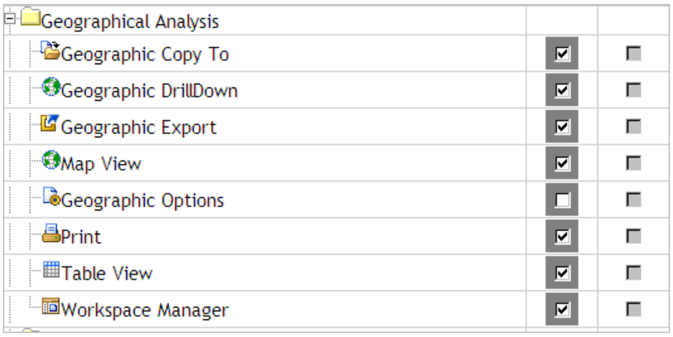
GroupB inherits the denial from GroupD. When the permissions for GroupA and GroupB are merged, the more restrictive permission (the denial) is applied. although the HCM Analyst has a direct grant for the object, the group permissions take precedence over the role permissions. As a result, sasdemo cannot use **Find People**.

*Note:* In SAS Human Capital Management, only users should belong to roles, and only users inherit role permissions.

## Add Permissions for an Object

To add permissions for an object:

1. In the Administration application, click the **Security** tab.
2. Click **Search Users** in the toolbar, and search for a user, group, or role. (See "Searching for Identities" on page 73.)
3. In the search results, click the action menu ▾ at the left of an identity and select **View Permissions**. The list of objects is displayed, along with the permissions for that identity.



| | | |
|---|---|---|
| Geographical Analysis | | |
| Geographic Copy To | ☑ | ☐ |
| Geographic DrillDown | ☑ | ☐ |
| Geographic Export | ☑ | ☐ |
| Map View | ☑ | ☐ |
| Geographic Options | ☐ | ☐ |
| Print | ☑ | ☐ |
| Table View | ☑ | ☐ |
| Workspace Manager | ☑ | ☐ |

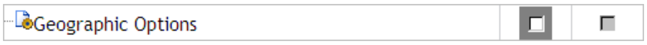
The **Permit** check boxes indicate four possible states:

| Permit | State |
|---|---|
| ☑ | Direct grant |
| ☐ | Direct denial |
| ☑ | Inherited grant |
| ☐ | Inherited denial |

*Note:* If a group has no parent group and no direct permissions, then its permissions are displayed as an inherited denial. However, only direct permissions from such a group are passed on to its subgroups. See "How Object Permissions Are Interpreted" on page 68.

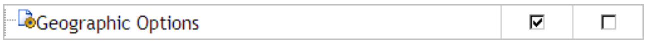
4. To apply or modify the direct permission for an object, select or clear the **Permit** check box for that object.

   For example, this user inherits a denial for the **Geographic Options** action in a geographic analysis:

   Geographic Options ☐ ☐

   To grant the permission directly, you would click the **Permit** box until it contained a check mark, with no highlighting:

   Geographic Options ☑ ☐

5. To apply an inherited permission, click the **Inherit** box so that it turns gray.
6. Click **OK** to save your changes.

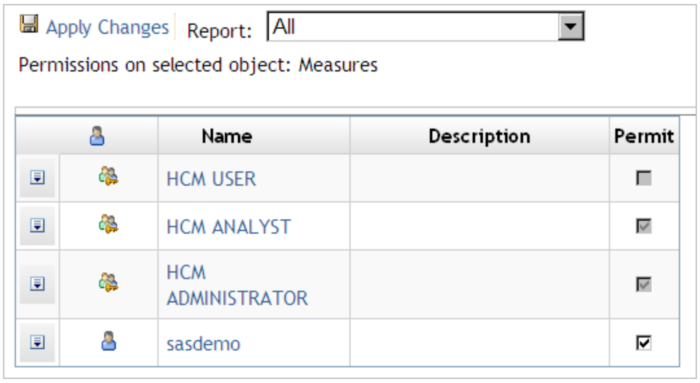
## *Modify Direct Permissions for an Object*

To modify the permissions that are directly set on an object:

1. On the **Security** tab of the Administration application, select **Objects** ⇨ ***category*** ⇨ ***object-name***.

   The display pane lists any identities (users, groups, or roles) that are directly granted or denied permission to perform the action. A check in the **Permit** column signifies that the permission is granted. If the check box is not selected, the permission is denied.

In this example, the sasdemo user and users with the HCM ANALYST or HCM ADMINISTRATOR role are granted permission to modify the measures in an organization analysis. However, users with the HCM USER role are denied this permission:



Apply Changes Report: All

Permissions on selected object: Measures

| | 👤 | Name | Description | Permit |
|---|---|---|---|---|
| ▾ | 👥 | HCM USER | | □ |
| ▾ | 👥 | HCM ANALYST | | ☑ |
| ▾ | 👥 | HCM ADMINISTRATOR | | ☑ |
| ▾ | 👤 | sasdemo | | ☑ |

Notice that the **Permit** boxes are dimmed for all three roles, signifying that you cannot change those permissions.

2. To change the permission for an identity, select or clear the **Permit** check box for that identity.
3. Click **Apply Changes** to save your changes.

To delete a permission, select **Remove** from the action menu ▾. (There is no need to click **Apply Changes**.)

If you delete a permission, the inherited permission for that identity applies.

## Secure a Custom Object

In addition to the standard objects that are part of SAS Human Capital Management, a site can secure its own custom objects. Currently, the only custom objects that are supported are in JavaServer Pages (JSPs).

*Note:* You can define an action that links to a JSP in the employee profile or a geographic analysis. For details, see "Define an External Action" on page 52.

To secure a custom object:

1. In the Administration application, click the **Security** tab.
2. In the navigation tree, select **Objects** ⇨ **Custom**.
3. On the Custom page, enter a name to identify the custom object. (The name does not need to correspond to the JSP name.)
4. In the **ID** box, enter an ID that is unique to the HCM application.
5. You can add multiple custom objects. If there are no available rows, click **Add Row**.
6. Click **Apply Changes**.
7. The custom object now appears in the navigation tree for object permissions. However, no default permissions are set for this object. Add permissions by following the instructions in "Add Permissions for an Object" on page 70.

   *Note:* You cannot associate permissions for custom objects with the default HCM roles.

8. If you are applying security to a link: In the JSP, embed the link in an **IF** block (JSP scriptlet) that is similar to the following:

   ```
   <%if(SecurityUtil.isPermitted( custom-object-id, request)) { %>
     HTML code for the link
      <% } %>
   ```

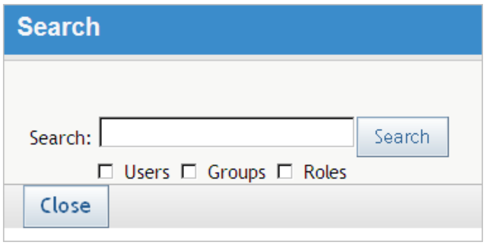
   The code to apply security to a button would be similar: embed the button code within an IF block in the JSP.

To delete a custom object, click the Delete button ✕ next to its object ID.

One example of securing a custom object might be restricting the users who can view employee photographs in the Employee Browser. If you have an employee profile JSP that displays employee photographs, you could embed the code that displays the photographs in the **IF** block.

## Searching for Identities

On the **Security** tab of the HCM Administration application, clicking **Search Users** displays a dialog box for selecting users, groups, or roles.



To search for users, groups, or roles:

1. In the **Search** box, enter a search string.

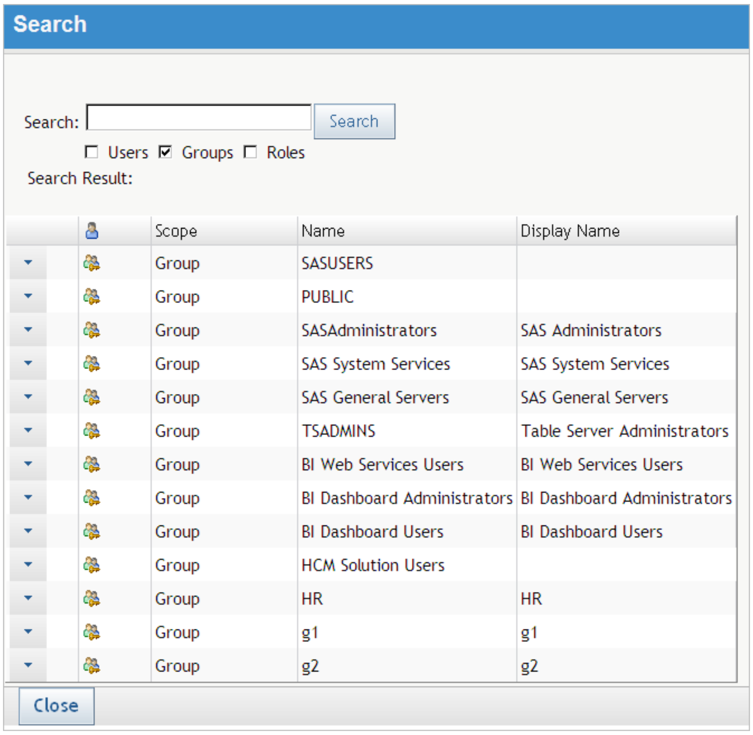
   The search string is not case-sensitive. You can enter the name of an identity (such as **sasdemo**) or its display name (such as **SAS Demo User**), or a partial search string (such as **demo**). Do not use quotation marks.

   To search for all identities within the specified scope, leave the search string empty.

2. Limit the scope of the search by selecting one or more of the following: **Users**, **Groups**, **Roles**.

3. Click **Search**.

   The search results display all matching identities.



Search

Search: [ ] Search

□ Users ☑ Groups □ Roles

Search Result:

|  | 👤 | Scope | Name | Display Name |
|---|---|---|---|---|
| ▾ | 👥 | Group | SASUSERS |  |
| ▾ | 👥 | Group | PUBLIC |  |
| ▾ | 👥 | Group | SASAdministrators | SAS Administrators |
| ▾ | 👥 | Group | SAS System Services | SAS System Services |
| ▾ | 👥 | Group | SAS General Servers | SAS General Servers |
| ▾ | 👥 | Group | TSADMINS | Table Server Administrators |
| ▾ | 👥 | Group | BI Web Services Users | BI Web Services Users |
| ▾ | 👥 | Group | BI Dashboard Administrators | BI Dashboard Administrators |
| ▾ | 👥 | Group | BI Dashboard Users | BI Dashboard Users |
| ▾ | 👥 | Group | HCM Solution Users |  |
| ▾ | 👥 | Group | HR | HR |
| ▾ | 👥 | Group | g1 | g1 |
| ▾ | 👥 | Group | g2 | g2 |

Close

4. From the action menu ▾ beside an identity, select an action. The available actions depend on the context of the search.

## Securing Table Rows

### *About Row-Level Security*

Row-level filters are one way of securing access to tables in SAS Human Capital Management. Each table has its own set of filters, each of which is associated with a user, group, or role. The most common filter grants access to all rows of a table, as in this example:

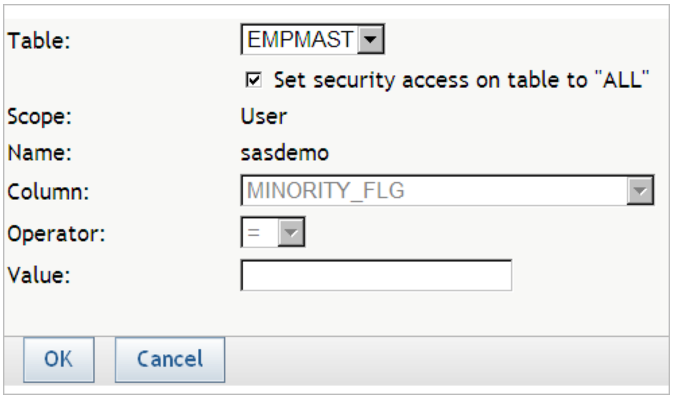


Table: EMPMAST ▾

☑ Set security access on table to "ALL"

Scope: User

Name: sasdemo

Column: MINORITY_FLG ▾

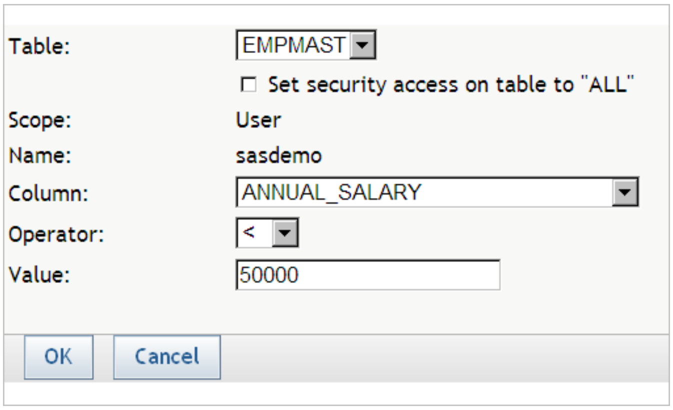
Operator: = ▾

Value:

OK Cancel

Because **Set security access on table to ALL** is selected, the **Column**, **Operator**, and **Value** boxes are dimmed. With this setting, the sasdemo user has access to all rows of the EMPMAST table, subject to hierarchical security (see "Hierarchical Filters" on page 80) and column security.

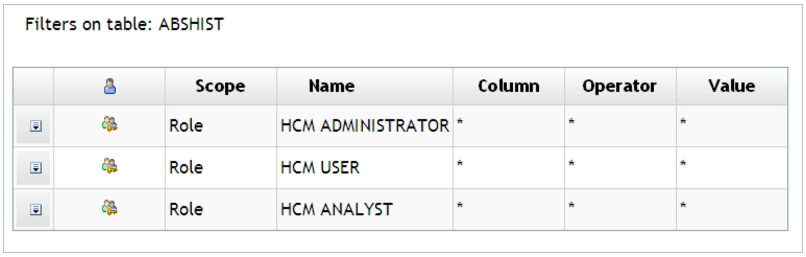
In this next example, the sasdemo user has access to the EMPMAST table only for employees with an annual salary that is less than $50,000.



| | |
|---|---|
| Table: | EMPMAST ▾ |
| | □ Set security access on table to "ALL" |
| Scope: | User |
| Name: | sasdemo |
| Column: | ANNUAL_SALARY ▾ |
| Operator: | < ▾ |
| Value: | 50000 |

## *Default Row-Level Filters*

By default, each of the SAS Human Capital Management roles (HCM USER, HCM ANALYST, and HCM ADMINISTRATOR) has complete access to each of the default tables, again subject to hierarchical filtering. This example shows the default filters for the ABSHIST table. The asterisks in the **Column**, **Operator**, and **Value** columns signify that any column, operator, or value is accepted. (The filters were defined with the **Set security access on table to ALL** setting.)



Filters on table: ABSHIST

| | 👤 | Scope | Name | Column | Operator | Value |
|---|---|---|---|---|---|---|
| ▾ | 👥 | Role | HCM ADMINISTRATOR | * | * | * |
| ▾ | 👥 | Role | HCM USER | * | * | * |
| ▾ | 👥 | Role | HCM ANALYST | * | * | * |

***CAUTION:***

> Each user must have at least one complete-access filter in order to have any access to a table. In the ABSHIST example, if you deleted the default filter for the HCM USER role, and did not add another filter that allowed complete access to the table, then no user with the HCM USER role would be able to access the table.

The complete-access filter can be applied to the user directly or to a role or group that the user belongs to, directly or indirectly. We recommend that you do not delete or modify the default role filters. Instead, use additional filters (for the user, group, or another role) to restrict a user's access to the tables. Hierarchical filters apply further restrictions.

## *How Row-Level Filters Are Applied*

Row-level filters apply only to tables that have row-level security enabled. (See "Enable Row-Level Security" on page 77.) With this attribute enabled, access is denied unless the user has a row-level filter that allows complete access to the table.

A row-level filter is implemented as an SQL WHERE clause. You can assign different filters, or combinations of filters, to different users, groups, or roles.

Row-level filters apply to the following applications and reports:

- the Employee Browser.
- organization analysis.
- geographic analysis.
- stored process reports that call the %BLDVIEW macro. (The standard SAS Human Capital Management reports call %BLDVIEW.)
- information maps (based on tables, not cubes) that are displayed via SAS Web Report Studio, or SAS Information Map Studio.
- the general search.
- supplemental schedules in forms (see "SAS for Workforce Planning & Budgeting" on page 149).
- the HCM public API (see "Customizing the Employee Profile Templates" on page 101).

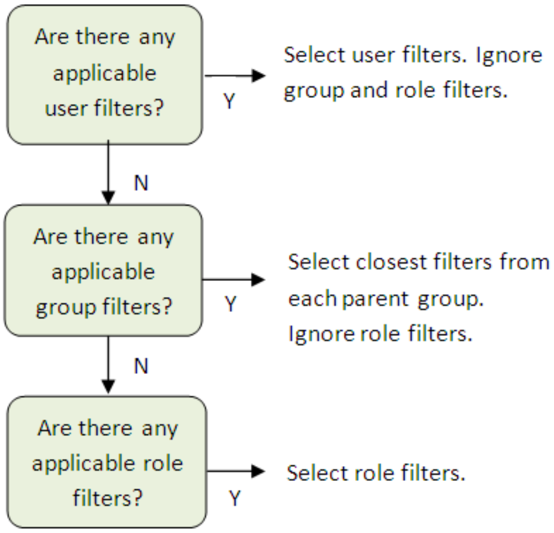
When a user tries to access a secured table, the security code retrieves the filters that are associated with that user, based on user identity and group and role memberships.

1. If the user has no complete-access filter for the table, access is denied.

   Otherwise, the WHERE clauses that make up the filters are then combined as described in the following steps:

2. Filters on the same column are grouped together, using the following rules:

   a. User filters have the highest precedence, group filters are next, and role filters are last, as shown in the following diagram:


Are there any applicable user filters?
Y
Select user filters. Ignore group and role filters.
N
Are there any applicable group filters?
Y
Select closest filters from each parent group. Ignore role filters.
N
Are there any applicable role filters?
Y
Select role filters.

   For any column, an identity might have multiple filters (for example, to select values within a range). In addition, multiple groups or roles can contribute filters for a column. However, if a group and any of its parent groups have filters for the same column, only the closest group filter is applied. (A filter for a parent group might still be applied to a different column.)

b. For the selected filters on the same column, if any operator is the equality operator (=) or the IN operator, then the WHERE clauses are combined with the OR operator. For example:

```
(EMPMAST.STATE_REGION_CD = 'NC' OR EMPMAST.STATE_REGION_CD = 'SC')
```

c. Otherwise, the clauses are combined with the AND operator. For example:

```
(EMPMAST.AGE > 35 AND EMPMAST.AGE <  65)
```

3. The AND operator is used to combine the group of filters on the same column with the next filter, which could be another group of filters or a single filter; and so on. For example:

```
WHERE ((EMPMAST.STATE_REGION_CD = 'NC' OR EMPMAST.STATE_REGION_CD = 'SC')
    AND (EMPMAST.AGE > 35 AND EMPMAST.AGE <  65))
```

*Note:* If there are more restrictive filters than the required **Set security access on table to ALL** filter, then the more restrictive filters apply.

4. The resulting filter string is combined with the hierarchical filter using the AND operator. For example:

```
WHERE ((EMPMAST.STATE_REGION_CD = 'NC' OR EMPMAST.STATE_REGION_CD = 'SC')
    AND (EMPMAST.AGE > 35 AND EMPMAST.AGE <  65) AND
    (EMPMAST.INTORG_MGR IN ('16407', '6917', '8272'))
```

5. Finally, the employee ID is added to the WHERE clause, using the OR operator:

```
WHERE ((EMPMAST.STATE_REGION_CD = 'NC' OR EMPMAST.STATE_REGION_CD = 'SC')
    AND (EMPMAST.AGE > 35 AND EMPMAST.AGE <  65) AND
    (EMPMAST.INTORG_MGR IN ('16407', '6917', '8272')) OR employee_id='8272')
```

*Note:* Each user is able to view all his or her own information. In this example, the age and state restrictions do not apply to the user's own records.

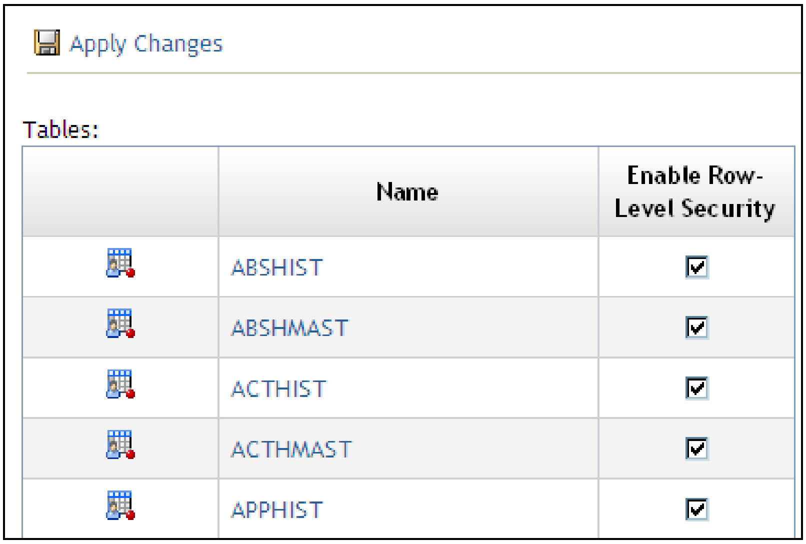
*Note:* You cannot disable the hierarchical filter. If the table does not contain the link field (such as INTORG_MGR), then only the employee ID is considered. For more information about hierarchical filters and the link field, see "Hierarchical Filters" on page 80.

## *Enable Row-Level Security*

By default, the standard HCM tables have row-level security enabled. When you add a table to SAS Human Capital Management, row-level security is enabled by default. To enable or disable row-level security for a table:

1. In the Administration application, click the **Security** tab.
2. From the navigation tree at the left, select **Tables**.

   The right pane lists the tables that are registered in SAS Human Capital Management, along with their row-level security settings:



Apply Changes

Tables:

| | Name | Enable Row-Level Security |
|---|---|---|
| | ABSHIST | ☑ |
| | ABSHMAST | ☑ |
| | ACTHIST | ☑ |
| | ACTHMAST | ☑ |
| | APPHIST | ☑ |

3. To enable row-level security for a table, select the check box to the right of the table name. To disable row-level security, clear the check box.

4. Click **Apply Changes**.

If row-level security is enabled for a table, a user must be directly or indirectly associated with a row-level filter for that table. Otherwise, access is denied.

### Add a Row-Level Filter

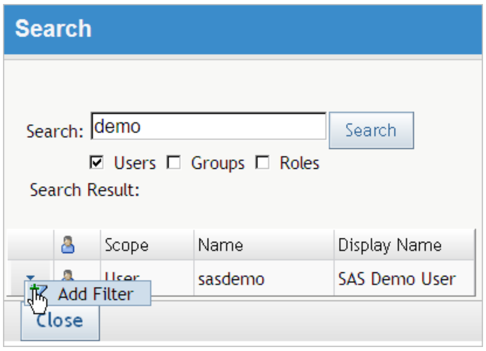
To add a row-level filter to a table:

1. In the Administration application, click the **Security** tab.

2. From the navigation tree at the left, select **Tables** or **Tables** ⇨ ***table-name***.

3. Click **Search Users** in the toolbar, and search for a user, group, or role.(See "Searching for Identities" on page 73.)

4. In the search results, select **Add Filter** from the action menu ▾ beside an identity.



Search

Search: demo

☑ Users □ Groups □ Roles

Search Result:

| | Scope | Name | Display Name |
|---|---|---|---|
| | User | sasdemo | SAS Demo User |

5. To allow access to all table rows, select **Set security access on table to "ALL"**.

   In this case, the **Column**, **Operator**, and **Value** boxes are dimmed.

6. Otherwise, select a **Column**, **Operator**, and **Value**.

   Valid operators are the following:

| | |
|---|---|
| = | Equal |
| != | Not equal |

| | |
|---|---|
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| IN | SQL IN operator, used to compare an expression to a set of expressions |

In the **Value** box, type an expression for the comparison (for example, a string or a numeric value). With the IN operator, enter a comma-separated set of values, such as **`red, green, blue`** or `2, 3, 4`.

*Note:* For columns with formats, only the codes—not the formatted values—are permitted.

7. Click **OK** to create the filter.

Filters are applied after the cache is refreshed (or the Web application server is restarted).

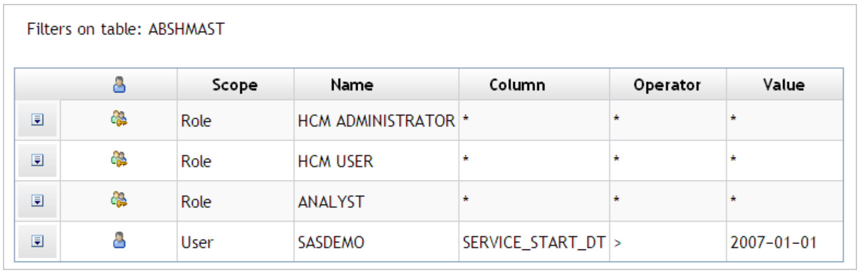
*Note:* You can also create row-level filters by means of a batch load script. See the files in the ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\Utilities\batchload`** directory on the middle tier.

### *Modify a Row-Level Filter*

To modify a row-level filter:

1. In the Administration application, click the **Security** tab.
2. From the navigation tree at the left, select **Tables** ⇨ ***table-name***.

   The display pane lists any identities (users, groups, or roles) that are associated with a filter for this table.



Filters on table: ABSHMAST

| | 👤 | Scope | Name | Column | Operator | Value |
|---|---|---|---|---|---|---|
| ▣ | 👥 | Role | HCM ADMINISTRATOR | * | * | * |
| ▣ | 👥 | Role | HCM USER | * | * | * |
| ▣ | 👥 | Role | ANALYST | * | * | * |
| ▣ | 👤 | User | SASDEMO | SERVICE_START_DT | > | 2007-01-01 |

3. Click the action menu ▣ beside the table and select **Properties**.

   The Filter Properties dialog box is displayed. For information about the fields in this display, see “Add a Row-Level Filter” on page 78.

*Note:* To create an additional filter for a user, group, or role, click the action menu ▣ beside the identity and select **Add Filter**. For detailed instructions,see “Add a Row-Level Filter” on page 78.

### *Delete a Row-Level Filter*

To delete a row-level filter:

1. In the Administration application, click the **Security** tab.

2. From the navigation tree at the left, select **Tables** ⇨ ***table-name***.
3. From the action menu ▾ next to a filter, select **Delete**.

# Hierarchical Filters

## *About Hierarchical Filters*

SAS Human Capital Management uses hierarchical filters to restrict access to data, based on the user's position in a specified hierarchy. These filters are an essential part of row-level security. They affect the data that is displayed, for example, when a user opens the Employee Browser, creates a geographic analysis or organization analysis, or runs the SAS Human Capital Management stored process reports.

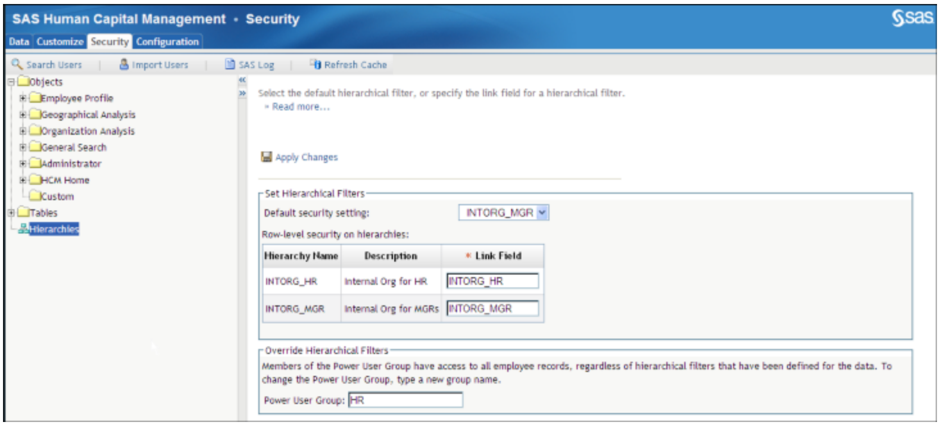
Consider the Employee Browser as an example. When users are browsing employee information, you want them to be able to view their own employee profiles. You also want managers to be able to view the profiles for their subordinates. However, you want to restrict users from viewing information about other employees at their level or farther up in the organization's hierarchy.

In SAS Human Capital Management, a hierarchical filter enforces those restrictions. Employees with no subordinates can view only their own records. Managers can view their own records as well as their subordinates' records.

## *Select a Hierarchy*

To select the hierarchy that is used for hierarchical filters:

1. In the Administration application, click the **Security** tab.
2. In the navigation tree, select **Hierarchies**.
3. Select a hierarchy from the **Default security setting** drop-down list.



Select the default hierarchical filter, or specify the link field for a hierarchical filter.

Set Hierarchical Filters

Default security setting: INTORG_MGR

Row-level security on hierarchies:

| Hierarchy Name | Description | * Link Field |
|---|---|---|
| INTORG_HR | Internal Org for HR | INTORG_HR |
| INTORG_MGR | Internal Org for MGRs | INTORG_MGR |

Override Hierarchical Filters

Members of the Power User Group have access to all employee records, regardless of hierarchical filters that have been defined for the data. To change the Power User Group, type a new group name.

Power User Group: HR

4. The link field for a hierarchical filter is the same as the name of the hierarchy.

   (If for some reason the site has changed the code that constructs the hierarchy, and you need to change the link field, type a new value in the **Link Field** box.)

   For information about the way hierarchical filters are constructed, see "How a Hierarchical Filter Is Applied" on page 81.

5. To change the name of the group that can override hierarchical filters, type a new name in the **Power User Group** box.

   For details, see "The Power User Group" on page 82.

6. Click **Apply Changes**.

## *How a Hierarchical Filter Is Applied*

Hierarchical filters are automatically applied to all tables that have row-level security enabled and that contain the link field for the hierarchical filter. For more information about the link field, see "Select a Hierarchy" on page 80 and "Tables Without Hierarchical Filters" on page 81.

The security code looks up the user's place in the specified hierarchy and gathers up all child members. It then generates a filter of the following form:

```
WHERE colName IN (child1, child2, child3, ...)
   OR employee_id=employee_id
```

- *colName* is the name of the column on which the filtering is applied, such as **INTORG_MGR**.
- *child1*, *child2*, and so on, represent the identification codes for all the subordinate members of the hierarchy.

Here is the WHERE clause for a hierarchical filter for the sample data. The user is the company CEO and has an employee ID of 10433. She can view her own data, as well as the data for all the managers who report to her, as well as their descendants.

```
WHERE (EMPMAST.INTORG_MGR IN ('11988','11301','19516','4119','8346','977',
'10203','11850','16153','6978','10537','12739','13890','16371','5588','6572',
'7702','12273','12136','12586','16442','1754','7344','8883','15994','10035',
'17054','17443','17498','7219','9222','17130','11766','16945','16948','17518',
'5261','5473','6777','7885','3654','10072','10510','11081','11685','7248','
7480','8165','3757','11490','12089','12679','16964','17559','1829','4837',
'7911','8012','4638','16875','3597','5431','9639','5469','10079','10314',
'11935','12284','12336','16059','16872','17180','5200','7590','774','12352',
'4134','8248','9369','10644','11938','18612','3414','6328','17191','11420',
'17208','7828','7602','16407','6917','8272','16074','1146','9600','11661',
'11778','13224','16838','17177','2681','7903','9713','8217','10327','10676',
'11726','11870','17320','4661','4845','5235','8977','9092','955','547',
'2973','9929','10433') OR EMPMAST.employee_id = '10433')
```

If there are any applicable row-level filters, those filters are combined with the hierarchical filter using the AND operator. (See "How Row-Level Filters Are Applied" on page 75.) If the table does not have a complete-access filter that applies to this user (directly or indirectly), then access is denied, regardless of the hierarchical filters.

Users who are not managers can view only their own information (for example, their own employee profile, based on a match with their employee ID).

For users who belong to the group that is designated as the **Power user** group, the hierarchical filters are ignored. (See "The Power User Group" on page 82.)

## *Tables Without Hierarchical Filters*

If row-level security is enabled for a table and the link column for the hierarchical filter is not present, the table is filtered by the user's employee ID. The user must still have at least one complete-access filter (assigned directly or indirectly) in order to view the table contents.

***CAUTION:***

> If you define any additional row-level filters for the table (other than complete-access filters), then the user is able to view the table data unfiltered by employee ID, but subject to the additional filters. (This caution applies only to tables without the hierarchical link field.)

The POS table does not allow hierarchical filtering at all. If row-level security is enabled, that table is filtered by the employee ID. There are three additional tables (APPHMAST, OPOSMAST, and OPOSSUM) that are built based on the POS table and contain the link field for the hierarchy that is designated as the default in the %PREBUILD macro, but no other link fields. If you select a different hierarchy on the **Security** tab, those three tables will not allow hierarchical filtering. They will be filtered by the employee ID. You can still create additional row-level filters for any of these tables (subject to the caution above). You can also set permissions on table columns.(See "Securing Table Columns" on page 83.)

## The Power User Group

At a site, you might need some users to have access to all rows of employee data, regardless of their position in the organization.

To override the hierarchical filters for all HCM tables, assign these users to the group that has been designated as the **Power user** group (by default, the HR group). The users then have access to the entire hierarchy (in other words, the hierarchical filter does not apply). However, other row-level security filters and other forms of security (such as object security and column security) still apply to these power users. For example, you might restrict access to columns that contain salary data.

There is still a way to restrict power users to a subset of the hierarchy, by creating a row-level filter on one or more of the hierarchy link fields. In that case, the user is restricted to the hierarchy members that meet the filter criteria, and subordinate members. For information about the hierarchy link fields, see "Select a Hierarchy" on page 80.

For example, assume that INTORG_HR is a hierarchy link field. To restrict power users to the department (which is represented in the INTORG_HR column), you might use a filter like this:

```
INTORG_HR = ‘QA’
```

At run time, the user could view records for employees in the QA department, as well as any subordinate departments.

If you create filters on more than one hierarchy link field, the filters are combined using the AND operator. For example, assume that you created these two filters for the ABSHMAST table in the sample data:

- **`INTORG_HR in (‘QA’, ‘COMP’)`**
- **`INTORG_MGR = ‘4638’`**

The resulting filter would look like this:

```
(ABSHMAST.INTORG_HR IN ('QAC','QAE','QAF','QAG','QAO','QAQ',
'QAR','QAS','QAT','QA','COMP') AND ABSHMAST.INTORG_MGR IN ('11490','12089',
'12679','16964','17559','1829','4837','7911','8012','4638')
OR ABSHMAST.employee_id = '10433')
```

Both the ‘QA’ and ‘4638’ hierarchy members would be expanded. The ‘COMP’ member would not. The INTORG_HR and INTORG_MGR filter clauses would be joined with the AND operator. The records would be filtered to include only members of the selected departments who also have one of the selected managers (as well as the user’s own record).

*Note:* This use of filters with hierarchy link fields applies only to power users. Otherwise, filters on hierarchy link fields are treated just like any other column filters.

## Securing Table Columns

### *About Column Security*

In addition to securing table rows, you can secure access to specific columns in a table. For example, you might want to hide salary information from users who do not have the appropriate authorization. Or you might want to prevent most users from seeing employees' Social Security numbers.

In order for a user to view data in a column, the user must have **ReadMetadata** permission for the column. In order for a user to modify column permissions, the user must have **WriteMetadata** permission for that column. These permissions can be directly granted to the user or inherited from a group the user belongs to.

*Note:* Role permissions do not apply to columns.

Column permissions also apply to searches. In a search dialog box, a column does not appear unless the user has access to that column. Even if the user types in the column name, it is ignored in the search.

### *How Column Permissions Are Applied*

ReadMetadata and WriteMetadata authorization for columns is governed by the metadata server. In evaluating column permissions, the metadata server begins with permissions that are set directly on the column. Are there any direct permissions that are assigned to the user, or to a group or role the user belongs to?

If there are no direct permissions for a column, the authorization process looks at permissions for the table the column belongs to, and works its way up the inheritance tree, in the same way that it determines table permissions or folder permissions. For details, see the *SAS Intelligence Platform: Security Administration Guide*.

If you encounter unexpected results—for example, if testing shows that users can view a column that you thought was restricted—reexamine your permission assignments.

*Note:* Occasionally you might restrict a column that was already selected for display or selected as a search criterion in the Employee Browser, in an organization analysis, or in a geographic analysis. In those cases, the column continues to appear. However, users cannot search on the column, and the column values are zero (for numeric data) or empty strings (for character data). If you want to hide the column entirely, set the **Hide Column** attribute in the column properties on the **Data** tab. If you simply want to restrict searching on a column, clear the **Searchable** attribute in the column properties.

## Modify Permissions for a Table Column

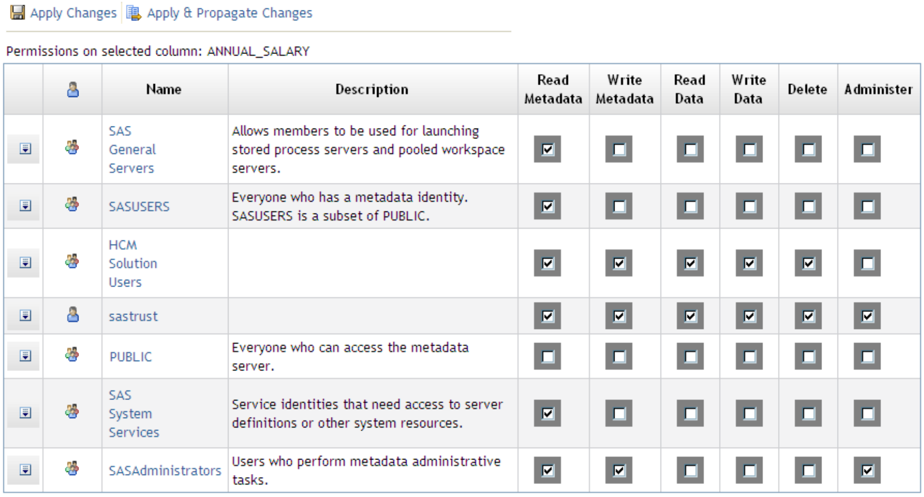
**CAUTION:**

Do not restrict access to any columns that serve as key attributes or link attributes in an employee profile, in a hierarchy mapping, or in a geographic analysis, an organization analysis, or the general search. If you restrict access to these columns, the applications will not function correctly.

To modify permissions for a table column:

1. In the Administration application, select the **Security** tab.
2. From the navigation tree at the left, select **Tables** ⇨ ***table-name*** ⇨ ***column-name***.

   The column permissions are displayed.



Permissions on selected column: ANNUAL_SALARY

| | 👤 | Name | Description | Read Metadata | Write Metadata | Read Data | Write Data | Delete | Administer |
|---|---|---|---|---|---|---|---|---|---|
| ▾ | 👥 | SAS General Servers | Allows members to be used for launching stored process servers and pooled workspace servers. | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| ▾ | 👥 | SASUSERS | Everyone who has a metadata identity. SASUSERS is a subset of PUBLIC. | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| ▾ | 👥 | HCM Solution Users | | ☑ | ☑ | ☑ | ☑ | ☑ | ☐ |
| ▾ | 👤 | sastrust | | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| ▾ | 👥 | PUBLIC | Everyone who can access the metadata server. | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| ▾ | 👥 | SAS System Services | Service identities that need access to server definitions or other system resources. | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| ▾ | 👥 | SASAdministrators | Users who perform metadata administrative tasks. | ☑ | ☑ | ☐ | ☐ | ☐ | ☑ |

A permission can have these states:

| Permission | State |
|---|---|
| ☑ | Direct grant |
| ☐ | Direct denial |
| ☑ | Inherited grant |
| ☐ | Inherited denial |

When you add an identity to the list, it receives a direct grant for ReadMetadata and inherits all its other permissions.

3. To override an inherited permission, click its check box. To restore the inherited setting, click the check box again.
4. To delete permissions for an identity, click the action menu ▾ at the left of the identity and select **Remove**.
5. To apply your changes to this table only, click **Apply Changes**.

6. To apply your changes to columns with the same name in all HCM tables, click **Apply & Propagate Changes**.

*Note:* Column permission changes apply immediately. **Refresh Cache** is not required.

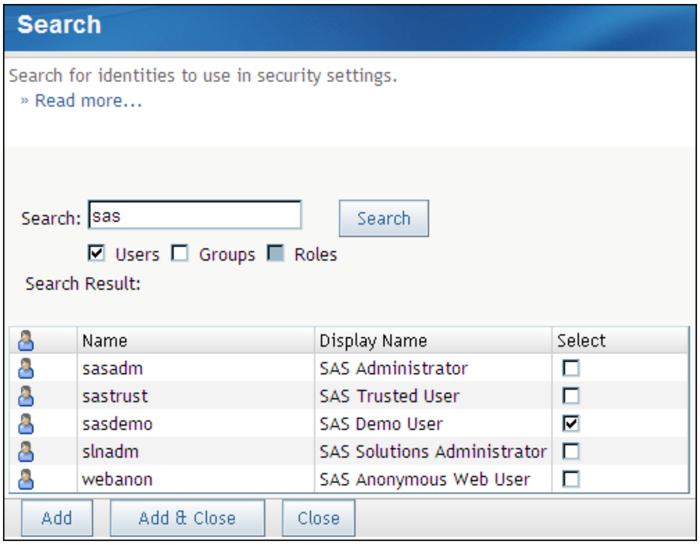
For information about the meaning of column permissions, see “About Column Security” on page 83. For information about the way permissions are applied, see “How Column Permissions Are Applied” on page 83.

## *Add an Identity to Table Column Permissions*

To add one or more identities to the set of table column permissions:

1. In the Administration application, select the **Security** tab.
2. From the navigation tree at the left, select **Tables** ⇨ ***table-name*** ⇨ ***column-name***.
3. Click **Search Users** and search for one or more identities.

   For column permissions, search only for users and groups. Role permissions do not apply to columns. (For search instructions, see “Searching for Identities” on page 73.)
4. From the search results, select the check box for one or more identities.



Search for identities to use in security settings.

Search: sas

☑ Users ☐ Groups ▨ Roles

Search Result:

| | Name | Display Name | Select |
|---|---|---|---|
| | sasadm | SAS Administrator | ☐ |
| | sastrust | SAS Trusted User | ☐ |
| | sasdemo | SAS Demo User | ☑ |
| | slnadm | SAS Solutions Administrator | ☐ |
| | webanon | SAS Anonymous Web User | ☐ |

5. Select an action:
   - Click **Add** to add these identities to the column permissions page.
   - Click **Add & Close** to add the identities and return to the column permissions page.
   - Click **Close** to return to the column permissions page without adding any identities

   Follow the instructions in “Modify Permissions for a Table Column” on page 84 to set permissions for the identities that you added.

# *Chapter 5*
# Configuring SAS Human Capital Management

**The Diagnostic Utility** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **87**
About the Diagnostic Utility . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
Summary of Diagnostic Tests . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
Running the Diagnostic Utility in Stand-alone Mode . . . . . . . . . . . . . . . . . . . . . . . . 90
Running the Diagnostic Utility from the Administration Application . . . . . . . . . . . 92
Viewing the Diagnostic Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

**About the Configuration Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **94**

**System Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **94**

**Application Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **96**

**Custom Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **98**

**The SAS_DEFAULT_PROPERTIES Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **99**
About the SAS_DEFAULT_PROPERTIES Table . . . . . . . . . . . . . . . . . . . . . . . . . 99
Graph Property Defaults for the Search Results . . . . . . . . . . . . . . . . . . . . . . . . . . 99

## The Diagnostic Utility

### *About the Diagnostic Utility*

The SAS Human Capital Management Diagnostic Utility checks connection pool settings, pings servers, checks that users have been set up correctly, and performs other similar tasks. Diagnostics are deployed on the middle-tier machine at the following path (Diagnostics Root):

```
SAS-config-dir\Lev1\Applications
\SASHumanCapitalManagement5.2\Diagnostics
```

The Diagnostics Root contains subfolders with diagnostics information, including diagnostics components, and the last run diagnostics report (DiagnosticsResults.html). The `Results` folder under the Diagnostics Root contains archived reports. The following table provides an overview of configuration files in the Diagnostics Root:

| File | Description |
|---|---|
| CommonDiagnosticsConfig.xml | Contains information about the connection settings for the container and metadata server for a deployment. Also, contains information about the library locations referred to by the Diagnostics Utility. |
| DiagnosticsMasterConfig.xml | Contains information about other diagnostics configuration files. Also, contains what components to run diagnostics for by solution and platform, and e-mail settings. The DiagnosticsMasterConfig.xml file helps enable diagnostics if multiple solutions are configured. |
| HCMDiagnosticsConfig.xml | Contains information about SAS Human Capital Management diagnostics tests.<br>The HCMDiagnosticsConfig.xml file is the primary file used to run diagnostics tests. It contains configuration information about what tests to run, how to run them, what parameters to use while running them, and so on. |
| PlatformDiagnosticsConfig.xml | Contains information about platform-level diagnostics tests. |

The following sections explain how to run the Diagnostic Utility in stand-alone mode or from SAS Human Capital Management, and how to interpret diagnostics results.

*Note:*

- You might be prompted to specify a password when using the Diagnostics Utility. Passwords are preserved in encrypted form in HCMDiagnosticsConfig.xml.
- If you encounter warnings or errors when running a diagnostic, or you performed a migration, you might need to complete additional setup tasks before using the Diagnostic Utility. See "Post-Configuration Steps" on page 342. Also, see the "HCM Modifications for Running Diagnostics" section in the *SAS Performance Management Solutions 5.2: 9.1.3 to 9.2 Migration Guide*.
- If you installed SAS Human Capital Management in a language other than English, you must first edit the diagnostics configuration script. For instructions, see "Post-Configuration Steps" on page 342.

### Summary of Diagnostic Tests

The following table provides information about the diagnostic tests that you can choose to run by module. For more information about selecting tests in stand-alone mode, see "Select Tests" on page 91. For more information about selecting tests in SAS Human Capital Management, see "Running the Diagnostic Utility from the Administration Application" on page 92.

***Table 5.1*** *Diagnostic Tests by Module*

| Test | Tab | Module | Description |
|---|---|---|---|
| Availability of Key Tables/Columns | HCM | Database | Checks whether key tables and columns (used for checking metadata tables) are available. |
| Connection Pool Settings | HCM | Container | Checks whether the HCM connection pool is running, and gets some basic connection pool properties if the HCM connection pool is running. |
| Critical Seed Values | HCM | Database | Checks for the presence of seed data within metadata tables. |
| Database Availability | HCM | Database | Checks whether databases are available within the database server. |
| Deployed Applications | Platform, HCM | Container | Checks whether applications (EARs, WARs) are successfully running within the container. |
| HCM Event Listener | HCM | HCM Custom Test | Checks whether the HCM Event Listener is running. This test also checks whether the SAS event SAS.Solutions.Service.Requested is registered and whether its listener in the SAS Human Capital Management application is running correctly. |
| HCM Roles and Groups | HCM | Metadata | Checks for the presence of HCM roles and groups within the metadata server. |
| HCM Solution - Datatier file(s)/ folder(s) | HCM | FileSystem | Checks for the presence of files or folders on the data tier. |
| HCM Solution - Midtier file(s)/ folder(s) | HCM | FileSystem | Checks for the presence of files or folders on the middle tier. |
| Ping Database | HCM | Database | Checks whether the database server is running and gets basic properties if the database server is running. |
| Ping Metadata Server | Platform | Metadata | Checks whether the metadata server is reachable. |
| Ping OLAP Server(s) | Platform | Metadata | Checks whether the OLAP server is reachable. |
| Ping Stored Process Server(s) | Platform | Metadata | Checks whether the stored process server is reachable. |
| Ping Workspace Server(s) | Platform | Metadata | Checks whether the workspace server is reachable. |

| Test | Tab | Module | Description |
|---|---|---|---|
| SAS Product Expiration | Platform, HCM | Metadata | Checks SAS product expiration information. |
| Server Properties | Platform | Container | Connects to the Container and gets basic information about the server, such as the Java vendor, version, container version, and so on. |
| Software Components | Platform, HCM | Metadata | Checks whether software components are available in the metadata server. |
| Software Components' Properties | Platform, HCM | Metadata | Used to print properties of software components. This test also checks whether software components are available and reports an error if a component is not found. |
| Verify Content Server | HCM | Metadata | Verifies that the content server is reachable. |
| Verify HCM Content Types | HCM | Metadata | Checks that the HCM content types have been registered in the metadata and their respective Java class associations. |
| Verify HCM Data Library | HCM | Metadata | Verifies the existence of the HCM data library as well as the tables registered under the library. This test also checks for the presence and correctness of the value of the HonorTableSecurity (HonourRowLevelSecurity) flag within the metadata server. |
| Verify SAS License | Platform | Metadata | Checks for a valid SAS license and reports related warnings. |
| Verify Users have some HCM Role | HCM | Metadata | Checks whether all users under configured HCM groups have at least one of the configured HCM roles. |
| Verify Users have valid HCM Employee mapping | HCM | HCM Custom Test | Verifies that all users that are part of configured HCM groups have valid HCM Employee record association. |

## Running the Diagnostic Utility in Stand-alone Mode

### About Stand-alone Mode

Running the Diagnostic Utility in stand-alone mode does not require SAS Human Capital Management to be running, and can be performed if you have access to the middle-tier machine. It is recommended that you run the Diagnostic Utility in stand-alone mode as soon as the SAS Human Capital Management middle-tier and data-tier configuration is complete. It is important to run the Diagnostic Utility at this point because it can determine whether the middle-tier application is running properly. If it is not, then the Diagnostic Utility cannot be run through the SAS Human Capital

Management application, because a user would be unable to log on to SAS Human Capital Management.

The Diagnostic Utility can be run in two stand-alone modes. Both modes can be run with .bat files located at the Diagnostics Root.

### *Select Tests*

The first stand-alone mode can be launched with the launchDiagnostics_UI.bat file. The launchDiagnostics_UI.bat file opens a window that can be used to select the tests to be carried out and launch the Diagnostic Utility. After beginning the diagnostic, a command window opens to track the progress of the diagnostics process. When the diagnostics are complete, a diagnostic report opens.

To run the launchDiagnostics_UI.bat file, perform the following steps:

1. Navigate to the Diagnostics Root.
2. Double-click launchDiagnostics_UI.bat. A window opens where you can select the tests to be carried out and launch the Diagnostic Utility.

   For more information about the purpose of each diagnostic test, see "Summary of Diagnostic Tests" on page 88.
3. Select the check box for each test that you want to include.
4. (Optional) Select **Email Diagnostics Report** and enter a comma-separated list of e-mail addresses in the associated box.
5. Click **Diagnose**. After beginning the diagnostic, a command window opens to track the progress of the diagnostic. When the diagnostic has completed running, an HTML report is displayed.

The diagnostic report lists the findings for each module and element. For more information about interpreting the HTML diagnostic report, see "Viewing the Diagnostic Results" on page 92.

*Note:* If you encounter errors running the Diagnostic Utility, see the System Administration part of this book for help with resolving them.

### *Run Preselected Tests*

The second stand-alone mode can be launched with the launchDiagnostics_cmd.bat file. Each time the Diagnostic Utility is run, information about which tests have been selected to run is saved in the HCMDiagnosticsConfig.xml file. When the Diagnostic Utility is launched from the launchDiagnostics_cmd.bat file, previously saved selections are used.

To run the launchDiagnostics_cmd.bat file, perform the following steps:

1. Navigate to the Diagnostics Root.
2. Double-click launchDiagnostics_cmd.bat. After beginning the diagnostic, a command window opens to track the progress of the diagnostic.

When the diagnostic has completed, a report is not automatically launched. The diagnostic report can be accessed by opening the DiagnosticsResults.html file in the `Results` folder at the Diagnostics Root.

*Note:* If you encounter errors running the Diagnostic Utility, see the System Administration part of this book for help with resolving them.

## Running the Diagnostic Utility from the Administration Application

The Diagnostic Utility can be run from the **Configuration** tab of the Administration application in SAS Human Capital Management. To run the diagnostic utility:

1. Click **Run Diagnostics**.

   General diagnostics information appears on the **Diagnostics** tab. Click the **Setup** tab for information about credentials settings. Click the **Platform** tab for a list of platform-level diagnostics tests to select. Click the **HCM** tab for a list of diagnostics tests for SAS Human Capital Management.

   For more information about the purpose of each diagnostic test, see "Summary of Diagnostic Tests" on page 88.

2. For each element that you want to be included in the diagnostics, select the **Select to verify** check box.

3. (Optional) Select **Email Diagnostics Report** and enter a comma-separated list of e-mail addresses in the associated box.

4. Click **Diagnose**. When the diagnostic has completed running, an HTML report is displayed.

*Note:*

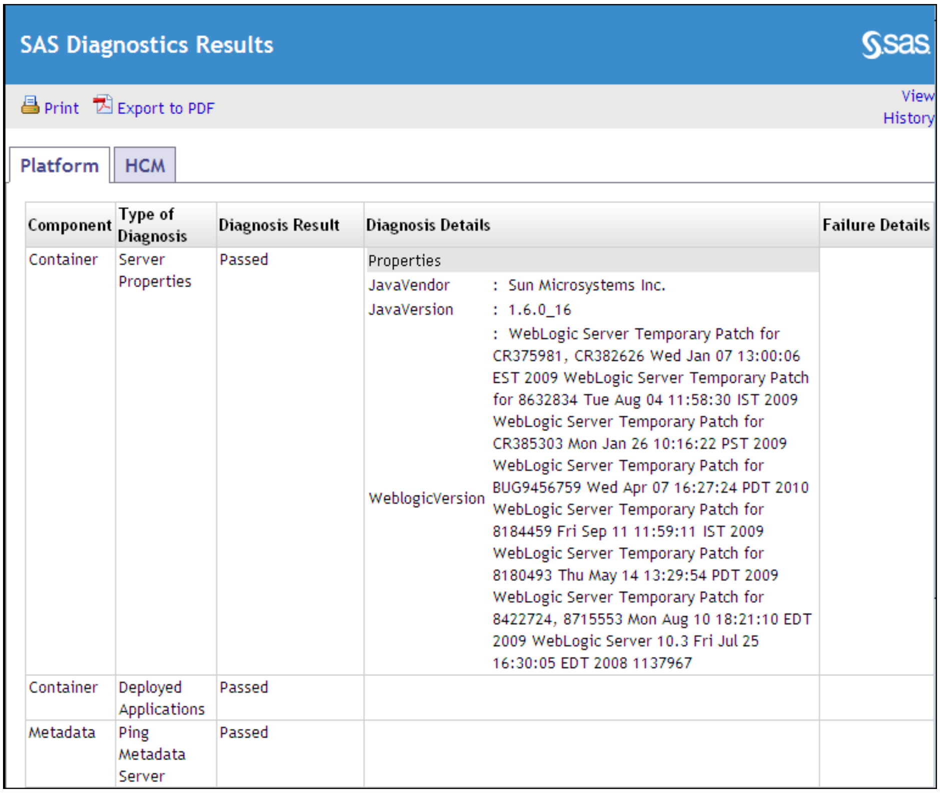
- If you encounter errors running the Diagnostic Utility, see the System Administration part of this book for help with resolving them.
- The Diagnostic Utility cannot be run from the SAS Human Capital Management Administration application when SAS Human Capital Management and SAS Financial Management are both deployed on the same instance of version 7 of the WebSphere Application Server. Diagnostics can still be run (or batched, if required) in stand-alone mode from the middle-tier server.

The diagnostics report lists the findings for each module and element. For more information about interpreting the HTML diagnostic report, see "Viewing the Diagnostic Results" on page 92.

*Note:* Depending on your installation configuration, the section about users and roles might report that the SAS Trusted User (sastrust) does not have an HCM role and does not have an employee mapping. Neither is required for that identity. See "Importing Users" on page 66.

## Viewing the Diagnostic Results

After running a diagnostic, results are grouped and presented in a table format:



SAS Diagnostics Results

| Component | Type of Diagnosis | Diagnosis Result | Diagnosis Details | Failure Details |
|---|---|---|---|---|
| Container | Server Properties | Passed | Properties<br>JavaVendor : Sun Microsystems Inc.<br>JavaVersion : 1.6.0_16<br>WeblogicVersion : WebLogic Server Temporary Patch for CR375981, CR382626 Wed Jan 07 13:00:06 EST 2009 WebLogic Server Temporary Patch for 8632834 Tue Aug 04 11:58:30 IST 2009 WebLogic Server Temporary Patch for CR385303 Mon Jan 26 10:16:22 PST 2009 WebLogic Server Temporary Patch for BUG9456759 Wed Apr 07 16:27:24 PDT 2010 WebLogic Server Temporary Patch for 8184459 Fri Sep 11 11:59:11 IST 2009 WebLogic Server Temporary Patch for 8180493 Thu May 14 13:29:54 PDT 2009 WebLogic Server Temporary Patch for 8422724, 8715553 Mon Aug 10 18:21:10 EDT 2009 WebLogic Server 10.3 Fri Jul 25 16:30:05 EDT 2008 1137967 | |
| Container | Deployed Applications | Passed | | |
| Metadata | Ping Metadata Server | Passed | | |

*Note:* You can export the report in SAS Human Capital Management by clicking **Export to PDF**.

Diagnostic tests are divided into the following categories based on the type of test.

- Container
- Database
- HCM Custom Test
- Metadata
- FileSystem

SAS Human Capital Management diagnostics are configured to work with a two-tier setup—one data-tier machine and one middle-tier machine. Diagnostics are deployed on the middle-tier machine.

Each diagnostic test has a default configuration that you can use. The HCMDiagnosticsConfig.xml file can be modified to accommodate customizations, such as a change to a logical server name, application name, or a table name. The ConnectionSettings section in the HCMDiagnosticsConfig.xml file contains entries for the data-tier and middle-tier machines on which SAS Human Capital Management is deployed, and the respective connection credentials. The Diagnosis sections in the HCMDiagnosticsConfig.xml file control all of the available SAS Human Capital Management diagnostic tests. Each XML node corresponds to a diagnostic test. The Connection section contains details about which elements are to be diagnosed in a particular diagnostic test, and in some cases how an element is to be diagnosed.

## About the Configuration Properties

The configuration properties are grouped as follows:

- **system properties** that apply to SAS Human Capital Management system utilities (see "System Properties" on page 94)
- **application properties** that apply to SAS Human Capital Management applications (see "Application Properties" on page 96)
- **custom properties** that are defined at a site (see "Custom Properties" on page 98)

These properties reflect default values for SAS Human Capital Management, as well as values that were set when the SAS Intelligence Platform and SAS Human Capital Management were installed and configured. Some of the properties are informational only (for example, the current version numbers for the SAS Intelligence Platform and SAS Human Capital Management). Other properties (such as the e-mail properties for a general search or for the employee profiles) can be reconfigured on this page without redeploying the application or restarting the Web application server.

If you make changes to the metadata repository or the Web application server, you can use the **Configuration** tab to update HCM properties so that they reflect those changes. After you make changes to the properties for a category, click **Apply Changes** to update the HCMConfig.xml file. After you finish making all your changes, click **Refresh Cache** if you want SAS Human Capital Management to begin using the new values immediately. Otherwise, the new values are used the next time the Web application server is restarted.

***CAUTION:***

Before making any changes, we suggest that you make a backup copy of the HCMConfig.xml file, which is stored in the ***`SAS-config-dir`*****`/Lev1/AppData/SASHumanCapitalManagement5.2`** directory. Your changes update this file, and you might want access to the file's previous values.

## System Properties

The system properties apply to SAS Human Capital Management system utilities.

If you are using a 64-bit Windows machine and have trouble importing an Excel 2007 file with an .xlsx extension, perform the following steps:

1. Navigate to HCMConfig.xml, which is stored in the ***`SAS-config-dir`*****`/Lev1/AppData/SASHumanCapitalManagement5.2`** directory.
2. Open HCMConfig.xml, and find the OSVersion property.
3. If the value of the OSVersion property is set to **`false`**, set it to **`true`**.

*Note:* You can check system properties in the Administration application by selecting **HCM System Properties** ⇨ **Version** on the **Configuration** tab.

***Table 5.2*** *HCM System Properties*

| Category | Information |
|---|---|
| **Configured Servers** | The names of the SAS servers that were initially configured, including the SAS OLAP server, SAS stored process server, and SAS workspace server. If you created an additional server after initial configuration (for example, if you created a pooled workspace server), it is not listed. These properties are Read-Only. |
| **Data Sources** | Data sources that are used by SAS Human Capital Management. All properties are Read-Only. |
| **Diagnostics Properties** | Properties that are used by the diagnostics tool. The **Diagnostics Installation Root** is configurable. It holds the path to the diagnostics configuration file. |
| **External Application Actions** | Actions that are used to open SAS Human Capital Management and other applications that are external to SAS Human Capital Management, such as SAS Web Report Studio. These values are Read-Only. They can be helpful in working with Technical Support. |
| **Folder Locations** | Default folder locations. You can configure the following properties:<br>• **HCMDefaultETLJobLocation**: When a user imports a table (on the **Data** tab), the user can choose to create an ETL job that loads the table in the HCM database. The job is stored in this location.<br>• **HCMDefaultEEPDocumentLocation**: Default location in which employee profiles are stored.<br>In order to use an employee profile, a user must have ReadMetadata permission for the profile document. You can use this permission to restrict access to specific profiles.<br>In order to update a profile, a user must also have WriteMetadata permission for the folder and its contents.<br>• **IMPORT_CONTENT_DAV**: The WebDAV location that holds documents that are imported into the workspace.<br>• **employeeImagesPath**: The path to the employee image files, within the deployed HCM application. The default is **`images/EmployeeImages`**.<br>• **employeeImageExtension**: The file extension for employee images (such as **`gif`** or **`png`**). Do not use a period. In the Employee Browser, an employee image is displayed if one is available, in the form *employee_id.extension*.<br>• **noPhotoFileName**: The filename of the picture to be displayed if no photo is available for an employee. Include an extension. The default is **`nophoto.png`**. The file must be located in **employeeImagesPath**. |
| **Logger Settings** | Logging levels for HCM applications. The **HCMLoggingLevel** is configurable. From most to least inclusive, the possible values are: **`DEBUG`**, **`INFO`**, **`WARN`**, **`ERROR`**, or **`FATAL`**. A value of **`DEBUG`** logs everything. A value of **`FATAL`** logs only fatal errors.<br>Logging is configured in the SASHumanCapitalManagement-log4j.xml file in the ***`SAS-config-dir`*****`\Lev1\Web\Common\LogConfig`** directory on the middle tier. If you leave the **HCMLoggingLevel** empty, the settings from the log4j file are applied. If you select a logging level, it affects any classes that start with **`com.sas.solutions.hcm`**. |
| **General Properties** | The General Properties apply to SAS for Workforce Planning & Budgeting:<br>• **newEmployeeIdTemplate**: This value is used as the prefix when a user adds a new employee in a planning form. (A unique identifier is appended to each new employee entry.) The default value is **`NewEmployee`**. The maximum length of this string is 16 characters. |

| Category | Information |
|---|---|
| **Localization Settings** | The Localization Settings contain two configurable properties:<br>• **HCMFontForNonDBCSCharacter**<br>• **HCMFontForDBCSCharacter**<br>These values are the default fonts for single- and double-byte character sets, respectively. They are used when a user saves data to a PDF file from the Employee Browser, a geographic analysis, an organization analysis, or the general search results.<br>The default value for **HCMFontForDBCSCharacter** is **`MHei-Medium`**. If your site supports DBCS languages, you should set the HCMFontForDBCSCharacter property appropriately for those languages. (Not all DBCS fonts support all DBCS languages.)<br>The following DBCS fonts are recommended:<br>• For Chinese Simplified: STSong-Light and STSongStd-Light<br>• For Chinese Traditional: MHei-Medium, MSung-Light and MSungStd-Light<br>• For Japanese: HeiseiMin-W3, HeiseiKakuGo-W5 and KozMinPro-Regular<br>• For Korean: HYGoThic-Medium, HYSMyeongJo-Medium and HYSMyeongJoStd-Medium<br>*Note:* If users encounter problems displaying a double-byte character set (DBCS), recommend that they upgrade their browsers to Windows Internet Explorer 7 or later. |
| **Security** | The default names of the three roles that are required in SAS Human Capital Management: **HCM Administrator**, **HCM Analyst**, and **HCM User**. If the site wishes to use different names for these roles, you can update the values in this section.<br>The new values must match roles that you have defined (and assigned) in the metadata repository. In addition, you must modify any properties that you have set using the old role names. For example, you must create new row-level security filters and modify any object security settings that were based on the default role names. |
| **Software Components** | The names of the software components that SAS Human Capital Management interacts with. These properties are configurable. For example, you could modify a software component to point to a different instance of an application. |
| **Web Services Inbound** | The Web services that are consumed by SAS Human Capital Management. These properties are Read-Only. |
| **Web Services Outbound** | The Web services that are provided by SAS Human Capital Management. These properties are Read-Only. |
| **Workspace Filters** | The document types that are supported in the SAS Human Capital Management workspace. These properties are Read-Only. |
| **Version** | The current versions of SAS Human Capital Management and the SAS Intelligence Platform. These properties are Read-Only. |

## Application Properties

The application properties apply to SAS Human Capital Management applications.

***Table 5.3*** *HCM Application Properties*

| Category | Information |
|---|---|
| **Common Default Settings** | Properties for determining active employees:<br>• **date_format_database**: Format in which dates are stored in the HCM database. This property is used to facilitate conversions between stored dates and displayed dates in the SAS Human Capital Management applications.<br>• **Employee_Status_Cd_Column**: Name of the column that contains the employee status code. Read-Only.<br>• **Employee_Status_Cd_Value**: The value that signifies active employees (such as **A**).<br>These values are used when SAS Human Capital Management needs to discriminate between active and inactive employees (for example, when you create an organization analysis). |
| **Employee Profile Default Settings** | You can set these properties:<br>• **eep_emailColumn**: The column that contains employees' e-mail addresses.<br>• **eep_emailTable**: The table that contains employees' e-mail addresses.<br>The **eep_emailColumn** and **eep_emailTable** properties apply when a user selects the e-mail option within the Employee Browser. This column and table apply regardless of the table that is associated with the current employee profile. As a result, a profile can use a table that does not contain employee e-mail information. However, only employees with an employee ID in the e-mail table are eligible to receive mail.<br>If the table or column is not available (if it does not exist, is hidden, or is not authorized), then the mail client opens with blank **To** fields and an error message is logged.<br>• **no_of_fmt_tooltip_examples**: On the **Search** tab, data tips (examples) are displayed for formatted columns. This value specifies the number of examples to display.<br>• **DefaultFixedCategoryName**: When you create an employee profile (on the **Customize** tab), a default fixed category is created. This value specifies the category name.<br>• **eep_printfromjsp**: If this value is **true**, then the **Save to PDF** action for the employee details uses a JavaServer page, HCMProfileDetailsPrint.jsp. This JSP is customizable. If the value is **false** (the default), then an internal mechanism is used instead. |
| **General Search Default Settings** | Properties that are used by the general search utility. You can set the following properties:<br>• **search_dataSource**: Data source on which the general search is performed. The default is the Employee Master table (EMPMAST).<br>• **search_emailColumn**: Column (within the specified data source) that contains employees' e-mail addresses.<br>• **search_keyAttribute**: Column (such as employee ID) that is used to retrieve an employee record.<br>• **search_linkField**: Column (such as employee name) that is used to link to an employee record.<br>The key attribute and link attribute always appear in the search results. |
| **Geographic Analysis Default Settings**s | Default beginning and ending colors for the maps in a geographic analysis. Specify the **Geo_StartColor** and **Geo_EndColor** as hexadecimal values (**#RRGGBB**). In a geographic analysis, users can override these values on the **Options** page. |

| Category | Information |
|---|---|
| **Organization Analysis Default Settings** | Default properties for organization analysis. You can set the following properties:<br>• **org_Measures**: The default measures to display in an organization analysis. The following values are valid:<br>  • **HCNT** (headcount)<br>  • **MN!*column-name*** (mean)<br>  • **MAXN!*column-name*** (maximum)<br>  • **MINN!*column-name*** (minimum)<br>  • **NN!*column-name*** (count)<br>  • **SN!*column-name*** (sum)<br>Separate multiple values by spaces—for example, **AGE!MIN AGE!MAX**.<br>• **org_Unassigned**: Text to display for hierarchy nodes that are not assigned to a department or manager.<br>• **org_Analysis_Cols**: Default columns to be displayed in an organization analysis. Separate column names with spaces (for example, **AGE CITY_NM ANNUAL_SALARY**). |

## Custom Properties

Custom properties are defined at a site to include any additional configuration properties that the site might require. They can be referenced in the same way as any configuration property, using the following code:

```
<%@page import="com.sas.solutions.hcm.core.util.ConfigurationUtility"%>
...
String variable-name = ConfigurationUtility.getPropertyValue("propertyID");
```

To define a custom property, follow these steps:

1. Open the HCMConfig.xml file for editing.

   This file is located in the ***SAS-config-dir*/Lev1/AppData/SASHumanCapitalManagement5.2** directory.

2. Find the block that begins with this line:

   ```
   <PropertyGroups Name="Custom" Desc="Custom Properties">
   ```

3. To add a property, modify the default property names (such as **Custom1**), IDs, and values, or add another line using the following syntax:

   ```
   <Property Id="propertyID" Name="propertyName" Value="propertyValue"
       ReadOnly="{true|false}"/>
   ```

4. Save the file.

# The SAS_DEFAULT_PROPERTIES Table

## *About the SAS_DEFAULT_PROPERTIES Table*

The SAS_DEFAULT_PROPERTIES table in the HCM database contains default display formats, graph options, and similar properties.

## *Graph Property Defaults for the Search Results*

Several properties contain the defaults for graphs that the user can create from the general search results.

- For properties with names ending in **Category**, the default value should be a character column such as INTORG_HR, MANAGER_NAME, or CITY_NM.
- For properties with names ending in **Response**, the default value should be a numeric (measures) column, such as AGE, ANNUAL_SALARY, or COUNT. To add statistics to a column, use the following syntax:

  ```
  column-name::statistic
  ```

The following statistics are available:

- COUNT
- MEAN
- MAX
- MIN
- FIRST
- LAST
- PERCENTAGE_FREQ
- CUMULATIVE_FREQ
- PERCENTAGE_SUM
- CUMULATIVE_SUM
- CUMULATIVE_PERCENTAGE_FREQ
- CUMULATIVE_PERCENTAGE_SUM

***Table 5.4*** *Default Properties for Graphs*

| Property | Default Value | Description |
|---|---|---|
| **BIPBarCategory** | INTORG_HR | Bar Chart Category variable |
| **BIPBarResponse** | COUNT | Bar Chart response variable |
| **BIPBarLineCategory** | INTORG_HR | Bar-Line Chart Category variable |
| **BIPBarLineResponse** | COUNT | Bar-Line Chart response variable |

| Property | Default Value | Description |
|---|---|---|
| **BIPBarLine_YResponse** | AGE::MEAN | Bar Line - Y response variable |
| **BIPLineCategory** | INTORG_HR | Line Graph Category variable |
| **BIPLineResponse** | COUNT | Line Graph response variable |
| **BIPPieCategory** | INTORG_HR | Pie Chart Category variable |
| **BIPPieResponse** | COUNT | Pie Chart response variable |
| **BIPScatterplotResponse_X** | AGE | Scatterplot Graph response-X variable |
| **BIPScatterplotResponse_Y** | ANNUAL_SALARY | Scatterplot Graph response-Y variable |

# Chapter 6
# Customizing the Employee Profile Templates

**Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **101**

**A Look at the Available Templates** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **102**
The Executive Profile Template . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
Additional Templates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103

**Customizing a Template** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **104**
About Customizing a Template . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
Required Imports . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
Styles and Style Sheets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
Displaying Header Details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
Displaying the Category List . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
Displaying Details for the Selected Category . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
Using Methods from the PublicAPIInterface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
Saving Employee Details in a PDF File . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112

**Making a Template Available** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **112**
Deploy the Custom JSP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
Make the Template Known to SAS Human Capital Management . . . . . . . . . . . . . 113

## Overview

SAS Human Capital Management supports custom employee profile templates, in the form of JavaServer Pages (JSPs) that use classes and methods of the HCM Public API. These templates determine the display of the profile header and profile details that appear in the Employee Browser, in terms of both format and content.

*Note:* The templates do not affect the profile summary (that a user might see upon selecting a node) or data that is displayed on the search page.

On the **Customize** tab of the Administration application, you associate a template with an employee profile. In addition to selecting from the templates that are available with SAS Human Capital Management, you can create custom templates that are appropriate for your site. The recommended approach is as follows:

1. Make a copy of an existing template (JSP file).
2. Customize your copied JSP. For example, you might modify the font or the layout of profile details, or you might display additional information in the profile detail area.
3. Make the template available.

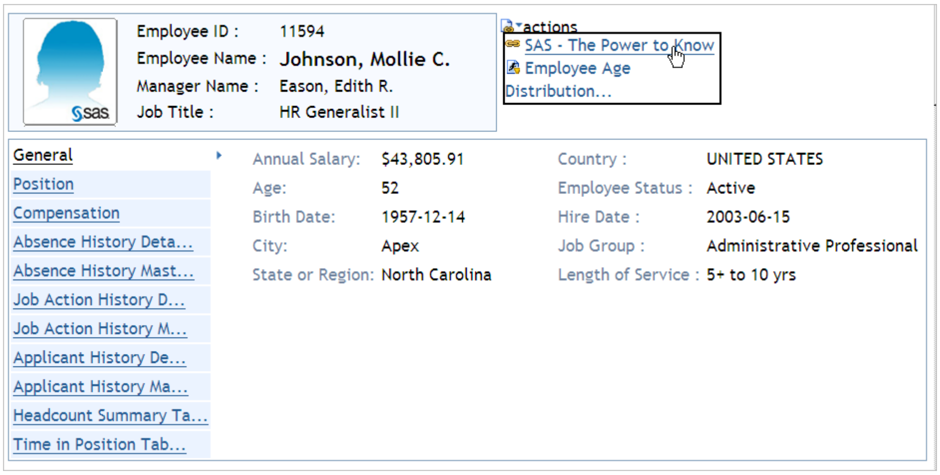
After you create or modify a template, you need to deploy it and then add it to the list of available templates. For instructions, see "Making a Template Available" on page 112.

# A Look at the Available Templates

## *The Executive Profile Template*

This image shows part of the Employee Browser. The user is displaying detailed information for employee Mollie Johnson.

***Figure 6.1*** *Profile Details with the Executive Profile Template*



Employee ID : 11594
Employee Name : Johnson, Mollie C.
Manager Name : Eason, Edith R.
Job Title : HR Generalist II

Annual Salary: $43,805.91
Age: 52
Birth Date: 1957-12-14
City: Apex
State or Region: North Carolina

Country : UNITED STATES
Employee Status : Active
Hire Date : 2003-06-15
Job Group : Administrative Professional
Length of Service : 5+ to 10 yrs

The employee profile is using the Executive Profile template (HCMDetailPhoto.jsp). This template organizes the display in a series of tables. In the first row of the outermost table, the following items are displayed:

- **the employee photo.** If there is no matching photo (as in this case), a default image is displayed.
- **the employee header.** The fields for the header are defined in the employee profile.
- **any external actions** that are defined with this profile. If the user is not granted permission to access external actions, or no external actions are defined, then nothing is displayed.

The second row of the outermost table displays these items:

- **a list of categories.** A category can be one of the following:
  - **a fixed category.** The fixed categories are defined in the employee profile. They consist of columns from one or more tables, organized into logical groupings (categories).
  - **a custom category.** In the Employee Browser, a user can add one or more tables or information maps to his or her profile. These are called custom categories. When the user selects a custom category from the profile details, the rows that match the employee ID are displayed.

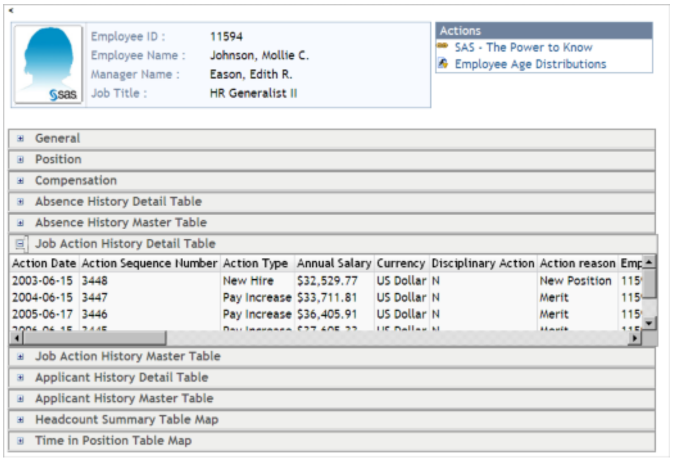
- **details for the selected category.** Notice that the profile details page contains links for several categories, including **General**, **Position**, and **Compensation**. When the user selects a category, the details (such as annual salary, age, and birthdate) are displayed.

## *Additional Templates*

Other available templates are as follows:

- **Section View Profile** (HCMProfileTemplateSectionView.jsp) displays categories in sections. The user clicks the plus sign beside a section title to display its contents.

  ***Figure 6.2*** *Employee Details with the Section View Template*



| | |
|---|---|
| Employee ID : | 11594 |
| Employee Name : | Johnson, Mollie C. |
| Manager Name : | Eason, Edith R. |
| Job Title : | HR Generalist II |

Actions
SAS - The Power to Know
Employee Age Distributions

⊞ General
⊞ Position
⊞ Compensation
⊞ Absence History Detail Table
⊞ Absence History Master Table
⊟ Job Action History Detail Table

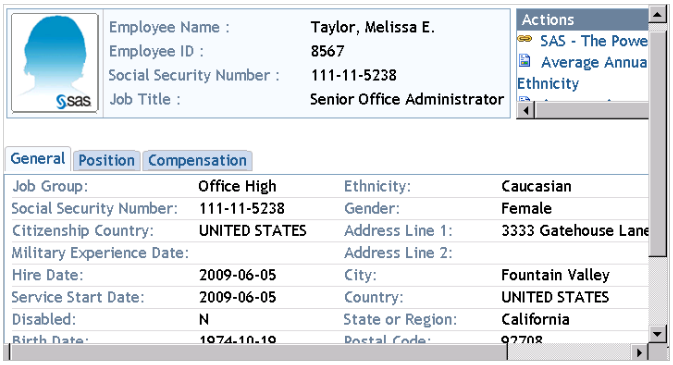
| Action Date | Action Sequence Number | Action Type | Annual Salary | Currency | Disciplinary Action | Action reason | Emp |
|---|---|---|---|---|---|---|---|
| 2003-06-15 | 3448 | New Hire | $32,529.77 | US Dollar | N | New Position | 115 |
| 2004-06-15 | 3447 | Pay Increase | $33,711.81 | US Dollar | N | Merit | 115 |
| 2005-06-17 | 3446 | Pay Increase | $36,405.91 | US Dollar | N | Merit | 115 |

⊞ Job Action History Master Table
⊞ Applicant History Detail Table
⊞ Applicant History Master Table
⊞ Headcount Summary Table Map
⊞ Time in Position Table Map

- **Tab View template** (HCMProfileTemplateTabView.jsp) displays categories as a set of tabs. A user clicks a tab to display the corresponding information.

  *Note:* This template should be used only for a small number of categories.

  ***Figure 6.3*** *Employee Details with the Tab View Template*



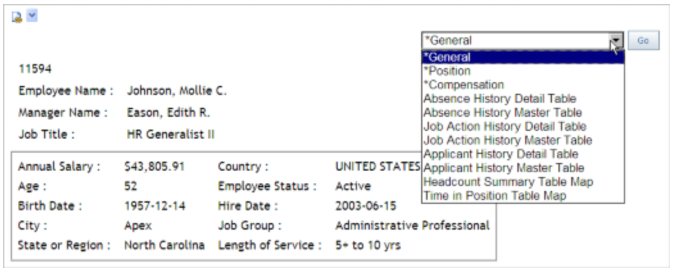
| | |
|---|---|
| Employee Name : | Taylor, Melissa E. |
| Employee ID : | 8567 |
| Social Security Number : | 111-11-5238 |
| Job Title : | Senior Office Administrator |

Actions
SAS - The Powe
Average Annua
Ethnicity

| | | | |
|---|---|---|---|
| Job Group: | Office High | Ethnicity: | Caucasian |
| Social Security Number: | 111-11-5238 | Gender: | Female |
| Citizenship Country: | UNITED STATES | Address Line 1: | 3333 Gatehouse Lane |
| Military Experience Date: | | Address Line 2: | |
| Hire Date: | 2009-06-05 | City: | Fountain Valley |
| Service Start Date: | 2009-06-05 | Country: | UNITED STATES |
| Disabled: | N | State or Region: | California |
| Birth Date: | 1974-10-19 | Postal Code: | 92708 |

- **General Profile** (HCMProfileViewerContent.jsp) uses a drop-down list for selecting the category.

***Figure 6.4*** *Employee Details with the General Profile Template*


*General
Go
*General
*Position
*Compensation
Absence History Detail Table
Absence History Master Table
Job Action History Detail Table
Job Action History Master Table
Applicant History Detail Table
Applicant History Master Table
Headcount Summary Table Map
Time in Position Table Map

11594

Employee Name : Johnson, Mollie C.

Manager Name : Eason, Edith R.

Job Title : HR Generalist II

| | | | |
|---|---|---|---|
| Annual Salary : | $43,805.91 | Country : | UNITED STATES |
| Age : | 52 | Employee Status : | Active |
| Birth Date : | 1957-12-14 | Hire Date : | 2003-06-15 |
| City : | Apex | Job Group : | Administrative Professional |
| State or Region : | North Carolina | Length of Service : | 5+ to 10 yrs |

*Note:* This template does not use the Public API and is not customizable.

These JSPs are available on the middle tier at ***`SAS-config-dir`*`\Lev1\Web\Staging\exploded\sas.humancapitalmanagement5.2.ear\sas.humancapitalmanagement.war`**.

# Customizing a Template

## *About Customizing a Template*

The remainder of this chapter describes how to use the Public API to customize an employee profile template. The examples are taken from the Executive Profile template. The other example templates also call HCM Public API methods to display employee detail information. As noted earlier, each template has a different layout, such as a drop-down list or tabs to hold the list of categories. Like the Executive Profile template, they can be customized to display additional information or to use a different layout.

For additional documentation, see "The Public API" on page 706.

## *Required Imports*

The following classes must be imported in each JSP template:

```
<%@ page import="com.sas.solutions.hcm.publicapi.beans.ProfileBean" %>
<%@ page import="com.sas.solutions.hcm.publicapi.beans.GenericBean" %>
<%@ page import="com.sas.solutions.hcm.publicapi.beans.CategoryBean" %>
<%@ page import="com.sas.solutions.hcm.publicapi.PublicAPIInterface" %>
<%@ page import="com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory" %>
<%@ page import="com.sas.solutions.hcm.publicapi.models.PublicAPITableModel" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.Iterator" %>
```

## *Styles and Style Sheets*

The example templates define style attributes internally, as in this example from the Executive Profile template:

```
<style type="text/css">
table.tableview {
        border: solid 1px #7f9db9;
        border-collapse: collapse;
        font-size: small;
        width: 100%;
        }
.style2 {color: #6F829D}
</style>
```

In your custom template, you might decide to include a reference to a style sheet (CSS file) that would need to be deployed with the JSP.

## Displaying Header Details

### Overview

The first part of the JSP displays the profile header, which might include header attributes, an employee photograph, and external actions.

### Get the Selected Category

The category that the user selects is available on the request object. This method retrieves the selected category:

```
String selectedCategory = request.getParameter("select");
```

### Get a Reference to a PublicAPIInterface Object

The next step is to get a reference to an object that implements the PublicAPIInterface, using a static factory method:

```
PublicAPIInterface employeeProfilePublicAPI =
    PublicAPIFactory.getEmployeeProfilePublicAPI();
```

### Get a ProfileBean

The getProfileDetails method returns a ProfileBean, from which the JSP can retrieve header attributes, the path to employee images, and category attributes:

```
ProfileBean profileBean =
    employeeProfilePublicAPI.getProfileDetails(selectedCategory, request);
```

The ProfileBean class has the following methods:

***Table 6.1*** *Selected Methods of the ProfileBean Class*

| Method | Description |
|---|---|
| **`java.util.List<CategoryList> getCategoriesList()`** | Gets the categories list, which includes the category attributes. |
| **`public java.lang.String getEmpPhotoSrc()`** | Gets the source of an image of the currently displayed employee.<br>The returned value is in the form ***`path/employee_id.extension`***.<br>The *path* and *extension* are set on the **Configuration** tab of the Administration application. |
| **`public java.util.List<GenericBean> getExternalActions()`** | Gets the list of external actions that are attached to this employee profile. |

| Method | Description |
|---|---|
| **`public java.util.List<GenericBean> getHeaderAttributesList()`** | Gets the headerAttributes. |
| **`public boolean isCustomCategorySelected()`** | Returns **`True`** if the selected category is a custom category. |

### Display the Header Attributes

The example JSP gets a reference to the list of header attributes and iterates through the list. If it encounters the employee name, it displays that value in a larger, bold font:

```
<%
List headerAttrList = profileBean.getHeaderAttributesList();
Iterator itrHeaderAttrList = headerAttrList.iterator();
while (itrHeaderAttrList.hasNext()) {
     GenericBean headerAttrBean = (GenericBean)itrHeaderAttrList.next();
%>
<tr>
  <td width="25%" nowrap="nowrap"><%=headerAttrBean.getLabel()%> :</td>
<% if (headerAttrBean.getName().equals("EMPLOYEE_NAME"))
{
%>
  <td width="75%"><b><font size="3">
      <%=headerAttrBean.getValue()%></font></b></td>
<%
}
 else
 {
 %>
 <td width="75%" ><%=headerAttrBean.getValue()%></td>
```

The GenericBean class is used to store profile properties such as header attributes, category attributes, and actions. The following methods are available:

***Table 6.2** Selected Methods of the GenericBean Class*

| Method | Description |
|---|---|
| **`public java.lang.String getName()`** | Gets the property name (for example, the attribute name or action name). |
| **`public java.lang.String getValue()`** | Gets the property value. |
| **`public java.lang.String getLabel()`** | Gets the property label. |

### Display the Employee Image

To display employee images, the example JSP uses the ProfileBean to get the source of an image of the currently displayed employee.

```
String src = profileBean.getEmpPhotoSrc();
```

The returned value is in the form ***`path/employee_id.extension`***. The *path* and *extension* are set on the **Configuration** tab of the Administration application.

The employee image is displayed in the profile header, using code like this:

```
<tr><td align="center" background="<%=src%>"></td></tr>
```

If there is no image for an employee, a default image is displayed, as defined on the **Configuration** tab of the Administration application. For more information, see "System Properties" on page 94.

### Display the External Actions List

The example JSP gets a list of external actions from the ProfileBean, after first checking to make sure that the user has permission to view external actions and that actions are available:

```
<% if(employeeProfilePublicAPI.isActionPermitted("EEPActions",request))
{
   List externalActionsList = profileBean.getExternalActions();
   if(externalActionsList.size()>0)
   {
   ...
```

The JSP iterates through the list and displays each action as a hyperlink:

```
<img src="images/TemplateImages/<%= externalActionImage %>"
   width="15px" height="15px"/>
<a href="javaScript:openExternalAction('<%= externalActionLink %>')"
   title="<%= externalActionName %>"> &nbsp;
   <%= externalActionDisplayName %></a>
```

## Displaying the Category List

### Overview

Employee information is divided into categories. Only one category of information is displayed at a time.

A category can be fixed (defined by an administrator) or custom (defined by a user). A fixed category consists of a set of columns, which might not all come from the same table. A custom category can be a table or an information map.

Certain restrictions are applied:

- Only information about the selected employee is displayed. For example, if the custom category is the Absence History table, the display includes only absence details for the selected employee.
- The information is subject to security (for example, row-level security or column-level security).

Your custom JSP does not need to consider these issues unless you are displaying data that is external to SAS Human Capital Management.

### Display the Category List

From the ProfileBean, the example JSP gets a reference to the list of categories:

```
List categoriesList = profileBean.getCategoriesList();
```

Then it iterates through the list to display the category descriptions (labels). The HTML code for each unselected category is a link to the code that displays that category's information, as in this example:

```
<td nowrap="nowrap" bgcolor="#EBF3FF">
   <a href="<%=employeeProfilePublicAPI.getCategoryLink(categoryName,request)%>"
   title="<%=categoryLabel%>"><u><%=categoryLabelDisplayed%></u></a>
   </td>
```

## Displaying Details for the Selected Category

### Get a CategoryBean Object

After displaying the category list, the example JSP displays the details for the selected category. It sets a CategoryBean object to the selected category:

```
CategoryBean selectedCategoryBean  = new CategoryBean() ;
...
if (null != categoryName && categoryName.equals(selectedCategory))
{
    selectedCategoryBean = categoryBean ;
    ...
}
```

*Note:* In the initial display, the first fixed category is displayed.

The CategoryBean has the following methods:

***Table 6.3*** *Selected Methods of the CategoryBean Class*

| Method | Description |
|---|---|
| **`java.util.List<GenericBean> getCategoryAttributesList()`** | Gets the list of attributes of the selected fixed category. |
| **`public javax.swing.table.DefaultTableModel getCustomCategoryAttributesTableModel()`** | Gets the table model of the selected custom category. |
| **`public java.lang.String getLabel()`** | Gets the category label. |
| **`public java.lang.String getName()`** | Gets the category name. |

### Display Attributes for a Fixed Category

To display a fixed category, the example JSP first gets the list of attributes for the category:

```
if(! profileBean.isCustomCategorySelected()){
    List categoryAttributesList = selectedCategoryBean.getCategoryAttributesList();
...
}
```

Then it iterates through the list, using GenericBean methods to get the label and value of each attribute:

```
GenericBean genericBean = (GenericBean) categoryAttributesList.get(i-1) ;
out.print(genericBean.getLabel());
...
out.print(genericBean.getValue());
...
```

### Display Attributes for a Custom Category

To display custom category attributes, the JSP uses methods of the PublicAPITableModel class:

```
// If a table/infomap is selected
PublicAPITableModel tableModel = (PublicAPITableModel)
   selectedCategoryBean.getCustomCategoryAttributesTableModel();
Object[] columnLabels = tableModel.getColumnInfoNames();
if(tableModel != null)
{
   int columnCount = tableModel.getColumnCount();
   int rowCount = tableModel.getRowCount();
%>
```

The following methods are available in the PublicAPITableModel class:

***Table 6.4*** *Selected Methods of the PublicAPITableModel Class*

| Method | Description |
|---|---|
| **`public int getColumnCount()`** | Gets the number of columns that were returned for the table. |
| **`public Object getColumnInfo(int index, String columnInfoName)`** | Gets the column label of a column based on its name. |
| **`public Object[] getColumnInfoNames()`** | Gets the column labels. |
| **`public int getRowCount()`** | Gets the number of rows that were returned for the table. |
| **`public Object getValueAt(int i, int j)`** | Gets the value of the cell at (**`i, j`**). |

## Using Methods from the PublicAPIInterface

Some of the methods that are defined in the PublicAPIInterface take the employee ID as one of their parameters and can be used to retrieve information about other employees. For example, a template might use these methods to get details about an employee's manager. Other methods are utility methods, used to get a list of authorized columns or to get a table model.

The following methods are available:

*Table 6.5 Selected Methods of the PublicAPIInterface*

| Method | Description |
|---|---|
| `java.util.List <GenericBean> getAuthorizedColumns(java.lang.String , HttpServletRequest)` | Returns a list of authorized column names, given a table name. |
| `java.lang.String getCategoryLink(java.lang.String categoryName, HttpServletRequest request)` | Returns a link to view a category. |
| `java.lang.String getConfigValue(java.lang.String configKey, HttpServletRequest request)` | Returns the value corresponding to the key passed, from the HCM configuration. |
| `GenericResultBean getEmployeeDetails(java.lang.String employeeId, HttpServletRequest request)` | Returns the specified employee's details from the default search table (set on the **Configuration** tab of the Administration application). |
| `GenericResultBean getEmployeeDetails(java.lang.String tableName, java.lang.String employeeId, HttpServletRequest request)` | Returns the specified employee's details from the specified table. |
| `GenericResultBean getEmployeeDetails(java.lang.String tableName, java.lang.String employeeId, java.util.List<java.langString> columnList, HttpServletRequest request)` | Returns details of an employee from the specified table and columns. |
| `GenericResultBean getEmployeeList(HttpServletRequest request)` | Returns the details for all employees from the default search table (set on the **Configuration** tab of the Administration application). |
| `java.lang.String getEmployeeList(java.lang.String tableName, HttpServletRequest request)` | Returns details for all employees from the specified table. |
| `java.lang.String getEmployeeList(java.lang.String tableName, java.util.List<java.lang.String> columnList, HttpServletRequest request)` | Returns details for all employees from the specified table and columns. |

| Method | Description |
|---|---|
| **`java.lang.String getEmployeePhotographSrc(java.lang.String employeeId, HttpServletRequest request)`** | Returns the relative path to a photograph of the specified employee. This path is set on the **Configuration** tab of the Administration application. |
| **`java.util.List<GenericBean> getHeaderDetails(HttpServletRequest request)`** | Returns the employee profile's header details. |
| **`FastRelationshipTree getHierarchyTree(java.lang.String hierarchyCode, HttpServletRequest request)`** | Returns an object of FastRelationshipTree type, given a hierarchy code and a table name. |
| **`ProfileBean etProfileDetails(java.lang.String categoryId, HttpServletRequest request)`** | Returns employee details for the specified category of the current profile, as selected by the user. |
| **`java.util.List<java.lang.String> getTableList(HttpServletRequest request)`** | Returns a list of all tables that are authorized for the current user. |
| **`PublicAPITableModel getTableModel(java.lang.String[] columnNames, java.lang.String tableName, java.lang.String where, HttpServletRequest request)`** | Returns a PublicAPITableModel object based on the parameters passed. Results are filtered according to the security that has been defined. |
| **`boolean isActionPermitted(java.lang.String actionName, HttpServletRequest request)`** | Returns true if the specified action is permitted for this user. |

This example shows the use of PublicAPIInterface methods (and GenericResultBean methods) to retrieve details for a particular employee:

```
PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();

 GenericResultBean employeeDetailsBean = hcmPublicAPI.getEmployeeDetails(
               "empmast", "10433", request);

 List columnNameList = employeeDetailsBean.getColumnNameList();
 Iterator itrColumnName = columnNameList.iterator();

 List rowDataList = employeeDetailsBean.getRowDataList();
 Iterator itrRowDataList = rowDataList.iterator();

 while (itrColumnName.hasNext()) {
        String columnName = (String) itrColumnName.next();
        System.out.print(columnName);
 }
 while (itrRowDataList.hasNext()) {
        List rowList = (List) itrRowDataList.next();
        Iterator itrRowList = rowList.iterator();

        while (itrRowList.hasNext()) {
                String value = (String) itrRowList.next();
                System.out.print(value);
        }
 }
```

*Note:* For descriptions of GenericResultBean methods, see “The Public API” on page 706.

### Saving Employee Details in a PDF File

The templates use a customizable JSP to save employee information in a PDF file. The JSP for the Executive Profile template contains this Javascript for the **Save As** operation:

```
<script type="text/javascript">
    function saveProfileViewAsPDF(){
        var form = document.getElementById('profileTemplate');
        form.profileOperation.value="save" ;
        form.action="PrintProfile.do" ;
        form.submit();
    }
</script>
```

In the Javascript, **`profileTemplate`** is the HTML form name as specified in the JSP:

```
<form name="profileTemplate" id="profileTemplate" action="">
    <input type="hidden" name="profileOperation" value="" />
</form>
```

If the eep_printfromjsp property is set to **`true`** in the configuration properties, the action is executed by a JSP (HCMProfileDetailsPrint.jsp) that can be customized at a site. (The same JSP is used to print employee details, although profile templates do not contain any code for printing.) Otherwise, internal code is used for printing as well as saving to a PDF. (See “Application Properties” on page 96 for information about setting this property.)

## Making a Template Available

### Deploy the Custom JSP

After you create a custom JSP for displaying employee details, you need to deploy it (and any supporting files, such as images or CSS files) to the deployed Web application where the other templates reside. For information about deploying these files, see “Web Application Custom Content” in the Instructions.html file from your middle-tier installation. This file is available at ***`SAS-config-dir`*****`\Lev1\Documents`**.

*Note:* The SAS Human Capital Management application is available at ***`SAS-config-dir`*****`\Lev1\Web\Staging\exploded\sas.humancapitalmanagement5.2.ear`**. Refer to that file structure to determine the structure for your custom content.

## *Make the Template Known to SAS Human Capital Management*

After you deploy a template, add it to the list of templates that are available for an employee profile. For instructions, see "Working with Templates" on page 61. After a template is available, it can be assigned to an employee profile. See "Create an Employee Profile" on page 48.

# *Chapter 7*
# Forecasting in SAS Human Capital Management

**About Forecasting in SAS Human Capital Management** . . . . . . . . . . . . . . . . . . . . **115**
Overview of Forecasting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
Overview of the Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116

**Preparing the Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **117**

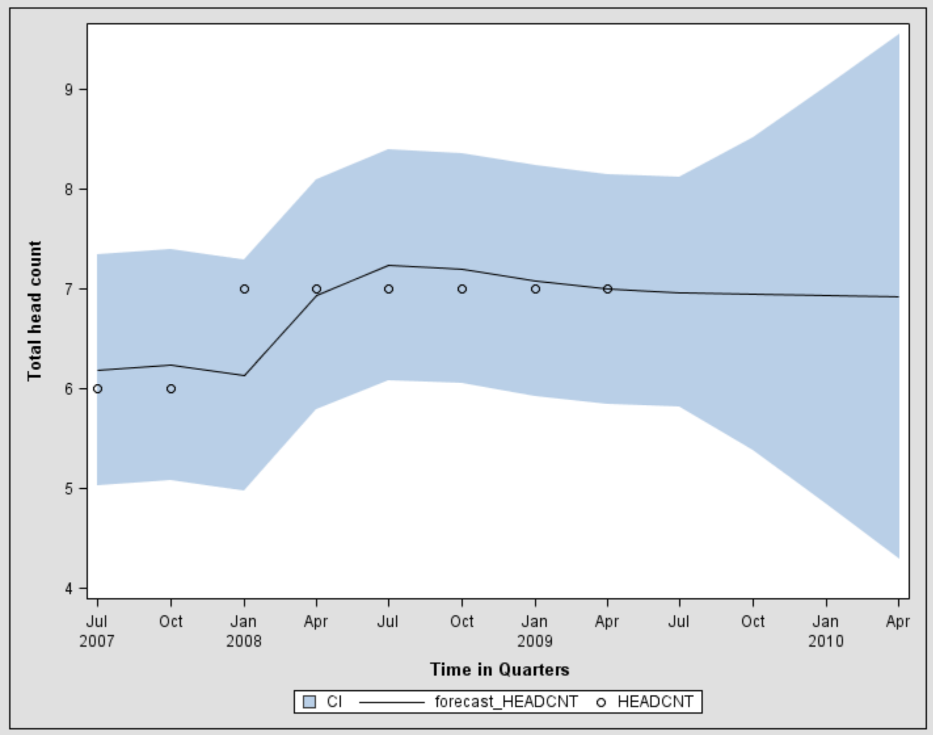
**Create a Stored Process Definition** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **120**

**Customize HCM Forecasting** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **124**
Add Classification Variables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124

## About Forecasting in SAS Human Capital Management

### *Overview of Forecasting*

The forecasting feature in SAS Human Capital Management predicts the value of a variable, based on the variable’s historical value. The forecast variable represents a measure of interest to the organization, such as headcount, voluntary terminations, or involuntary terminations. End users can generate a graph of the results, as in this example, which shows predicted headcount within a company over the coming years.



To generate the forecast data, an ETL job calls SAS procedures that perform high-performance forecasting. The forecasting software uses sophisticated automatic model-selection techniques to choose the best-fitting model for the data. It reconciles forecasts at different levels of aggregation, so that (for example) the forecast for a division is consistent with the forecasts for the individual departments that make up that division.

End users call a stored process that generates a graph of the forecast results. The graph displays the historical and predicted value of a variable. Users can filter the results by classification variables such as job group code, hierarchy variables, EEO-1 classification, or the job's permanence (such as regular or temporary). Here are some possible uses:

- projecting fluctuations in contract hiring over the next two years
- projecting headcount changes in a particular division
- projecting voluntary terminations (retirements and resignations) for a particular job group

A forecast does not make predictions for individual employees. It bases its predictions solely on the forecast variable's historical values, not on other variables that might contribute to its value. The more data points available, the more accurate the predictions.

## Overview of the Process

There are two parts to the forecasting process:

1. Generate the forecast data:
   a. In SAS Data Integration Studio, set the transformation properties for the ETL jobs that will generate the forecast data sets.
   b. Run the ETL jobs.
   c. In SAS Management Console, create a stored process definition for each of the forecast data sets.

The prompts for the stored process must match the selections that you made in the ETL job.

2. Run the stored process:

   a. An end user logs on to SAS Human Capital Management and runs a forecasting stored process.

   b. The stored process generates a graph, based on the forecast data set.

      End users can run the stored process multiple times, making different selections each time. Each execution would provide a different view into the same data set. In other words, the forecasting is done when the ETL job is run. The stored process filters the data, based on user selections, and generates the graph.

You can regenerate the forecast data sets periodically (for example, to collect additional historical data or to change the forecast variables or classification variables). If you modify the transformation properties in the ETL job, you must also modify the prompts in the corresponding stored process definition.

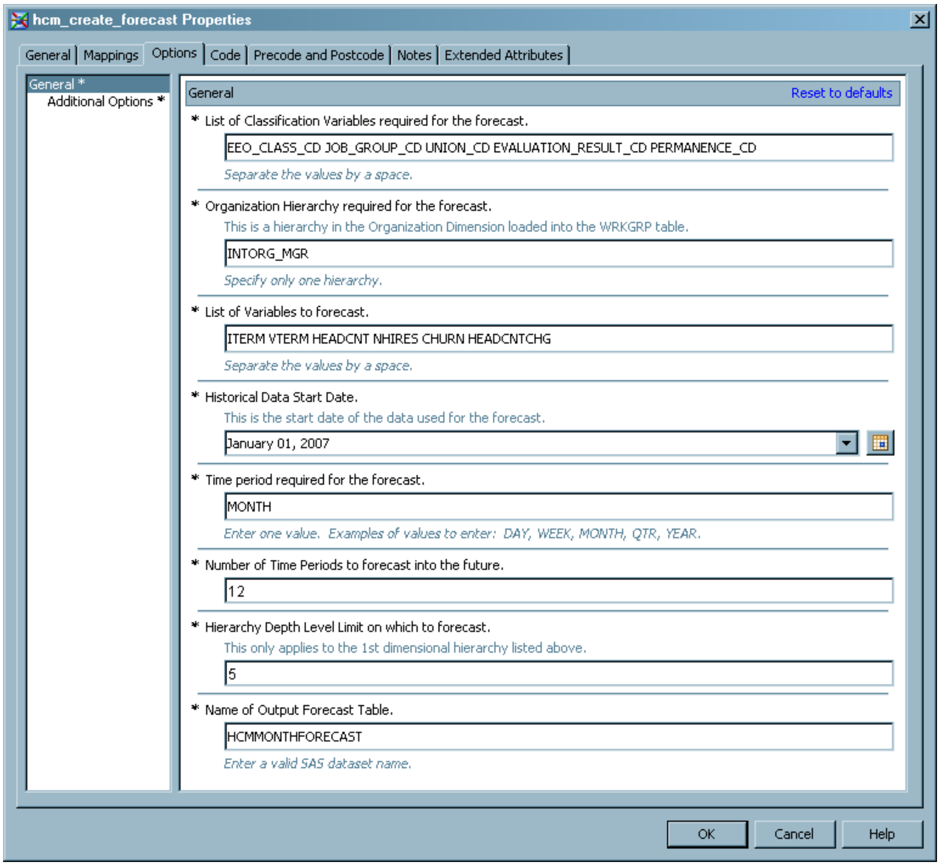
## Preparing the Data

An ETL job generates the forecasting data. To prepare the job for execution, follow these steps:

1. Log on to SAS Data Integration Studio as a user with the Data Administrator role.
2. Open the job in the process designer window.

   There might be multiple forecasting jobs. By default the software includes the following three jobs, for monthly, quarterly, and yearly forecasting:

   - hcm_126050_run_month_forecast
   - hcm_126100_run_quarter_forecast
   - hcm_126150_run_year_forecast
3. Right-click the transformation and select **Properties**.
4. On the **Options** tab, specify the following options:



* List of Classification Variables required for the forecast.
EEO_CLASS_CD JOB_GROUP_CD UNION_CD EVALUATION_RESULT_CD PERMANENCE_CD
*Separate the values by a space.*

* Organization Hierarchy required for the forecast.
This is a hierarchy in the Organization Dimension loaded into the WRKGRP table.
INTORG_MGR
*Specify only one hierarchy.*

* List of Variables to forecast.
ITERM VTERM HEADCNT NHIRES CHURN HEADCNTCHG
*Separate the values by a space.*

* Historical Data Start Date.
This is the start date of the data used for the forecast.
January 01, 2007

* Time period required for the forecast.
MONTH
*Enter one value. Examples of values to enter: DAY, WEEK, MONTH, QTR, YEAR.*

* Number of Time Periods to forecast into the future.
12

* Hierarchy Depth Level Limit on which to forecast.
This only applies to the 1st dimensional hierarchy listed above.
5

* Name of Output Forecast Table.
HCMMONTHFORECAST
*Enter a valid SAS dataset name.*

**List of Classification Variables required for the forecast**

Specify one or more variables, separated by spaces.

Classification variables enable you to categorize the forecast data. These variables have no effect on the forecast itself. Instead, they act as filters when an end user views the data in a graph. When the ETL job generates the forecast data, each classification variable becomes a column in the resulting data set. When users run the forecasting stored process, they can select one or more classification variables as filters for the data set. For example, they might select a specific job classification, a geographic region, or an age group.

Classification variables must exist in the Job Action History master table (ACTHMAST) and at least one of the following tables:

- Job Action History detail table (ACTHIST)
- Jobs detail table (JOBS)
- Employee General detail table (EMPGEN)
- Positions detail table (POS)
- Grades table (GRADE)

The following classification variables are included in the sample jobs. You can add to or replace these variables:

**EEO_CLASS_CD**

specifies the Equal Employment Opportunity classification for an employee.

**JOB_GROUP_CD**

specifies the job group code.

**UNION_CD**
specifies the union membership code.

**EVALUATION_RESULT_CD**
specifies the results of evaluations. For example, in the sample data, some possible evaluation results are **`Meets Expectations`**, **`Consistently Exceeds Expectations`**, and so on.

**PERMANENCE_CD**
specifies a job classification in terms of permanence (for example, **`Regular`** or **`Temporary`**).

**Organization Hierarchy required for the forecast**
Specify one hierarchy code. This hierarchy must be in the Work Group Table (WRKGRP), and it must be a populated column in the Job Action History Table (ACTHIST). This step is optional.

**List of Variables to forecast**
Specify the forecast variables to include in the transformation. Separate items in the list by spaces.

In SAS Human Capital Management, a single forecast variable can be viewed or graphed at a time. The ETL job can include multiple forecast variables, but the data for each variable is forecasted independently of the others. When users run a forecasting stored process, they select a single forecast variable to view in a graph.

Forecasting is supported for the following variables:

**VTERM**
specifies voluntary terminations—the number of employees that are projected to leave the organization voluntarily during the forecast period.

**HEADCNTCHG**
specifies the projected change in headcount (additions or subtractions) during the forecast period.

**NHIRES**
specifies the projected new hires during the forecast period.

**HEADCNT**
specifies the projected headcount during the forecast period.

**ITERM**
specifies involuntary terminations—the number of employees that are projected to leave the organization involuntarily during the forecast period.

**CHURN**
specifies the projected internal movement (such as employees moving to different departments) during the forecast period.

**Historical Data Start Date**
Specify the starting date for the data on which the forecasting software bases its predictions. For best results, specify a period that includes a minimum of 20 data points (for monthly forecasts, 20 months; for quarterly forecasts, 20 quarters; and so on). With fewer data points, the accuracy of the forecast is diminished.

**Time Period required for the forecast**
Specify a required time period. The example ETL jobs use these time periods: **`MONTH`**, **`QTR`**, and **`YEAR`**.

You can specify other time periods, such as weekly or 10-day intervals. For a complete list and description, see "About Date and Time Intervals" in the *SAS(R) 9.2 Language Reference*.

**Number of Time Periods to forecast into the future**
Specify how far into the future you want to forecast, in terms of the selected time period. For a monthly forecast, you would enter **12** to forecast a year into the future, **24** to forecast 2 years into the future, and so on.

**Hierarchy Depth Level Limit on which to forecast**
This field applies to the Hierarchy variable that you specify. It determines the number of levels for which forecast data is generated. As this number gets larger, the forecast generation time gets longer and the data set gets larger. We recommend that you start with a level of **1**, which means that forecast data would be available only at the top level of the hierarchy.

**Name of Output Forecast Table**
Give the output table a unique name. The table is stored in the HCM database and registered in the metadata repository.

For more information about forecasting with SAS, see SAS High-Performance Forecasting documentation at `http://support.sas.com/documentation/onlinedoc/hpf`.

## Create a Stored Process Definition

The default installation includes a stored process template and three example stored process definitions: **Monthly Forecast**, **Quarterly Forecast**, and **Yearly Forecast**. To create a new stored process definition, copy the template or one of the examples and modify it as necessary.

As you modify a stored process definition, the important thing to remember is that most of these parameters must match the variables that were specified in the ETL job that generated the forecast data set. The graph type and output type parameters are the only exceptions.

In the examples, parameters are grouped. You can rearrange those groupings for a different user interface. You can also give some parameters default values and hide them from end users, or make other site-specific changes.

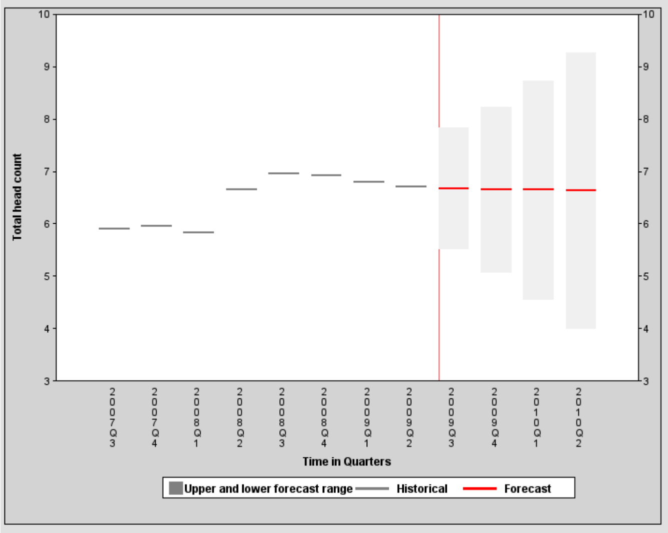
The default parameters are as follows:

**FORECAST_VARIABLE_NAME**
specifies the variable whose value you want to forecast. This variable must be one of the forecast variables specified in the ETL job that produced the forecast.

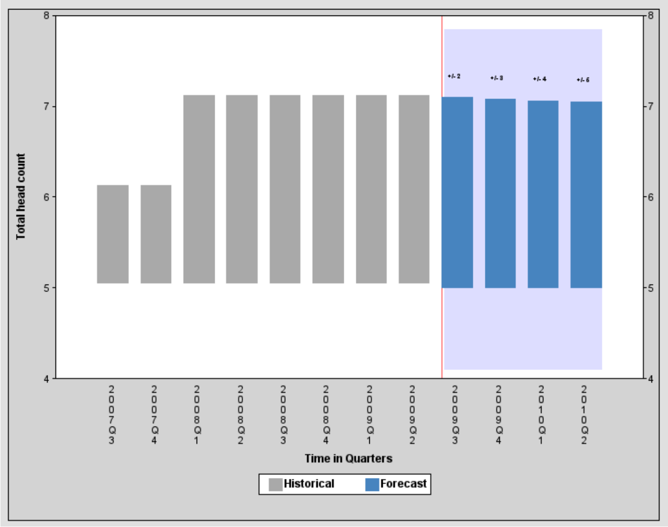
**HCMGRAPHTYPE_VAR**
specifies the type of graph in which to convey the historical and projected values of the forecast variable. The historical data is displayed to the left of the red line; the forecast data, to the right of the red line. The gray area represents the confidence interval.

By default, these selections are available:

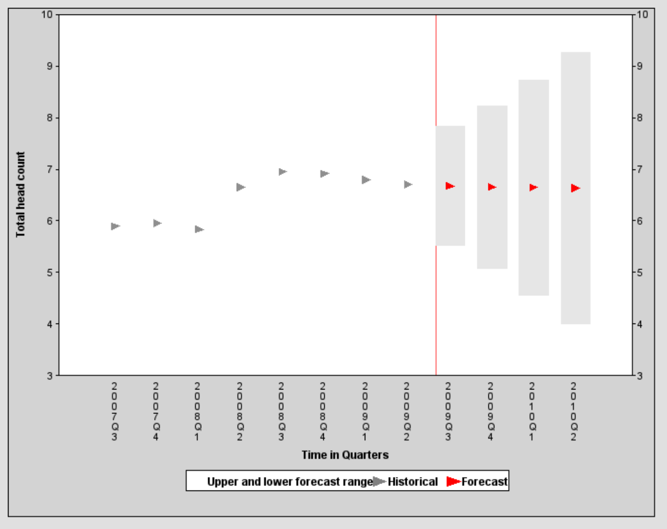
- `dlines` (Floating Lines) specifies a modified line graph. Confidence intervals are displayed as gray bar areas above and below the forecasted line.



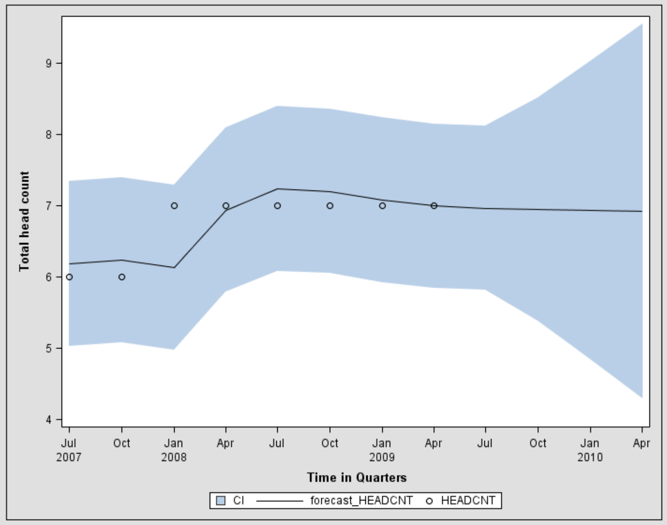
- **`bbars`** (Vertical Bars) specifies a traditional histogram. Confidence intervals can be displayed on the Time axis or on top of the histogram.



- **`sarrows`** (Floating Arrows) specifies a scatter plot that uses arrow symbols as markers.



- **`sgraph`** (Statistical Graph) specifies a traditional statistical graph. Confidence intervals are displayed in areas above and below the line plot.



### Data Input

Specify the name and interval of your forecast data.

**FORECAST_DATA_NAME**

is the name of the forecast data set, which must be registered in the metadata repository. This prompt is hidden from end users because other prompts like class and hierarchy are dependent on its value.

**FORECAST_TIME_INTERVAL_VAR**

specifies the time period that is associated with the forecast data. Only one time period can be specified. Its value must match the time period that was selected in the ETL job that generated the forecast data set. This prompt should be hidden from end users.

There are three possible values for the default stored processes. **`MONTHLY`** divides a year into 12 months and assumes that the input data is in that form. **`QTR`** divides a year into 4 quarters and assumes that the input data is in that form. **`YEARLY`** assumes a time period of 1 year for the forecast data set.

**Class variables**

These variables act as filters on the forecast data. For example, if the forecast data contains a category for job group code, the end user can select a specific job group value and view a graph that contains data only for that job group.

The prompts for these parameters should be lookups into the forecast data set. For example, the EEO_CLASS_VALUE prompt takes its possible values from the EEO_CLASS_CD column of the forecast data set. The following parameters are supported by default in the stored process. However, the parameters must match the classification variables that were selected in the ETL job that generated the forecast data.

**EEO_CLASS_VALUE**
specifies the EEO classification.

**JOB_GROUP_VALUE**
specifies the job group code.

**UNION_VALUE**
specifies the union membership code.

**EVALUATION_RESULT_VALUE**
specifies the results of evaluations.

**PERMANENCE_VALUE**
specifies the job position permanence (with values such as **`Regular`** or **`Temporary`**).

To add a prompt for your own class variables, see "Add Classification Variables" on page 124.

**Hierarchy variables**

These variables filter the results by hierarchy level (for example, by department or by manager).

The hierarchy variables can go only as deep as the hierarchy depth that you specified for the ETL job transformation. For example, the INTORG_HR hierarchy has these five levels: INTORG_HR5 (the highest level), INTORG_HR4, INTORG_HR3, INTORG_HR2, and INTORG_HR1. Assume that in the ETL job, you selected INTORG_HR as the hierarchy code, and three as the hierarchy depth. The forecast data would then include these columns: INTORG_HR5, INTORG_HR4, and INTORG_HR3. In the stored process definition, you would typically prompt for values for each of those three hierarchy variables. There is no point in drilling deeper into the organization, because there is no data available at lower levels.

*Note:* If you also specified a security hierarchy in the ETL job, then all levels of that hierarchy are available in the forecast data.

To ensure accuracy and to make things easier for your end users, we recommend using cascading prompts for these values, so that selecting a value at one level of the hierarchy restricts the prompts at the next lower level.

**Graph output parameters**

The output parameters modify how your graph is displayed.

**CI_DISPLAY_TYPE_VAR**
specifies the confidence interval display type.

**NUM_OF_HIST_LIMIT**
specifies the number of historical values. This prompt can be useful to produce a graph with fewer historical values and be less crowded, for example.

**CONFIDENCE_RANGE_LOCATION_VAR**
specifies the confidence range location.

**graphwidth**
specifies the width of the graph in pixels.

**graphheight**
specifies the height of the graph in pixels.

# Customize HCM Forecasting

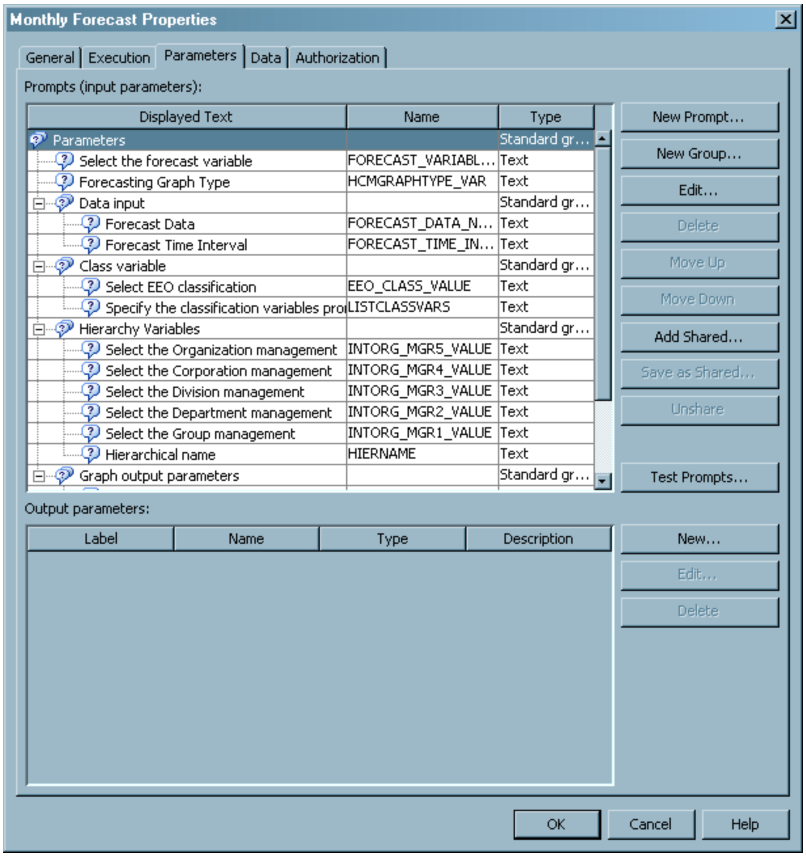
## *Add Classification Variables*

When creating your stored process definition, you can add prompts for additional classification variables with the following steps:

1. In SAS Management Console, navigate to the location of the forecasting stored process template. A typical path for this template in SAS Management Console on the **Folders** tab is **SAS Folders** ⇨ **Products** ⇨ **SAS Human Capital Management** ⇨ **Reports** ⇨ **Forecast**.

2. Copy the template to make a new stored process to modify, and then double-click on your new stored process.

3. Select the **Parameters** tab.



Prompts (input parameters):

| Displayed Text | Name | Type |
|---|---|---|
| Parameters | | Standard gr... |
| Select the forecast variable | FORECAST_VARIABL... | Text |
| Forecasting Graph Type | HCMGRAPHTYPE_VAR | Text |
| Data input | | Standard gr... |
| Forecast Data | FORECAST_DATA_N... | Text |
| Forecast Time Interval | FORECAST_TIME_IN... | Text |
| Class variable | | Standard gr... |
| Select EEO classification | EEO_CLASS_VALUE | Text |
| Specify the classification variables pro | LISTCLASSVARS | Text |
| Hierarchy Variables | | Standard gr... |
| Select the Organization management | INTORG_MGR5_VALUE | Text |
| Select the Corporation management | INTORG_MGR4_VALUE | Text |
| Select the Division management | INTORG_MGR3_VALUE | Text |
| Select the Department management | INTORG_MGR2_VALUE | Text |
| Select the Group management | INTORG_MGR1_VALUE | Text |
| Hierarchical name | HIERNAME | Text |
| Graph output parameters | | Standard gr... |

Output parameters:

| Label | Name | Type | Description |
|---|---|---|---|

4. Expand the **Class variable** list in the Forecast Properties dialog box, and double-click the name **LISTCLASSVARS**. This name corresponds to the **Specify the classification variables prompted for** displayed text.

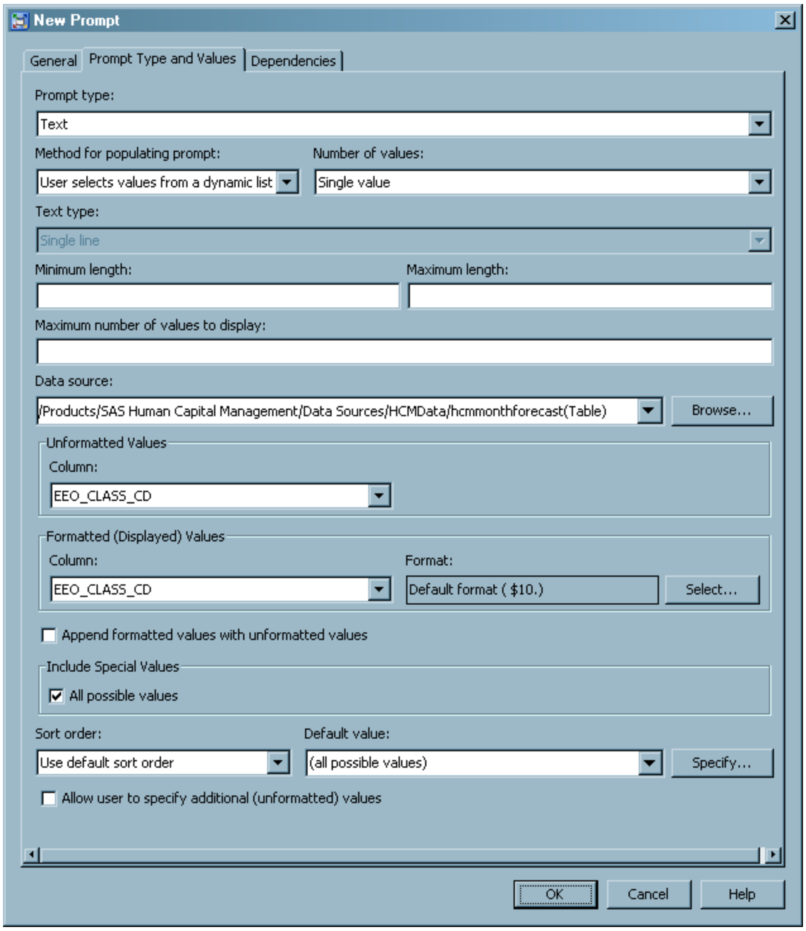
5. Select the **Prompt Type and Values** tab in the Edit Prompt dialog box, and specify available classification variables for a user to select in the **Default value** text box. Be sure to separate each value by a space. The column names of the classification variables that you specify in the **Default value** text box should match the class variables in the ETL job that created the data input FORECAST_DATA_NAME. Click **OK** when you have completed entering class variable names.

   *Note:* LISTCLASSVARS is a hidden variable and should not be added in the **Default value** text box.

6. For every class variable listed in the **Default value** text box, you need to create a prompt. To create a prompt for a class variable, select the **Class variable** group to highlight it. Next, click the **New Prompt** button. On the **General** tab, type the name of the prompt in the **Name** text box. This name needs to be the same as the class variable name, with "_VALUE" concatenated on the end. For example, if the class variable has a name of "HIRED_EMPLOYEES", then the prompt name that you need to enter is "HIRED_EMPLOYEES_VALUE".

7. Type the label for the prompt that you want a user to see in the **Displayed text** text box.

8. Select the **Prompt Type and Values** tab.



Prompt type: Text

Method for populating prompt: User selects values from a dynamic list

Number of values: Single value

Text type: Single line

Minimum length:

Maximum length:

Maximum number of values to display:

Data source: /Products/SAS Human Capital Management/Data Sources/HCMData/hcmmonthforecast(Table)

Unformatted Values

Column: EEO_CLASS_CD

Formatted (Displayed) Values

Column: EEO_CLASS_CD

Format: Default format ( $10.)

☐ Append formatted values with unformatted values

Include Special Values

☑ All possible values

Sort order: Use default sort order

Default value: (all possible values)

☐ Allow user to specify additional (unformatted) values

9. Select **User selects values from a dynamic list** in the **Method for populating prompt** drop-down menu.

10. Click the **Browse** button next to the **Data source** drop-down menu, and select the forecast table that contains the forecast data set. Note that this is the same data source as the FORECAST_DATA_NAME.
11. In the **Unformatted Values** section, select the class column name for the unformatted values from the **Column** drop-down menu.
12. In the **Formatted (Displayed) Values** section, select the column name for the formatted values from the **Column** drop-down menu. In the **Format** box, define the format that matches that value.
13. Select the **All possible values** check box to enable it.
14. Select **Use default sort order** from the **Sort order** drop-down menu.
15. Select **(all possible values)** from the **Default value** drop-down menu.
16. Click **OK**. Repeat this procedure to create prompts for each of the class variables that you specified in the **Default value** text box in the Edit Prompt dialog box for the LISTCLASSVARS class variable.
17. Next you need to set up cascading prompts for the hierarchy variables. Expand the **Hierarchy Variables** group. Double-click **HIERNAME** to open the Edit Prompt dialog box for that variable.
18. Select the **Prompt Types and Values** tab.
19. Select **User selects values for a static list** in the **Method for populating prompt** drop-down menu. In the List of Values pane make sure that there is one value, which is the name of the hierarchy without the level. Click **OK**.
20. Create prompts for each hierarchy level by beginning with the highest level and ending with the lowest level. It is important to set up the hierarchy levels in order to set up dependencies correctly. When creating the prompt for the highest level, select the **All possible values** check box.
21. After you have created prompts for each hierarchy variable, beginning with the highest level and ending with the lowest level in the organization, you need to configure the prompts. Configure the prompts beginning with the highest level and ending with the lowest level. Double-click the name for the highest level in your organization on the **Parameters** tab to open the Edit Prompt dialog box.
22. Select the **General** tab, and specify the name of the prompt and the text to display.
23. Select the **Prompt Type and Values** tab.
24. Make sure that **(all possible values)** is selected from the **Default value** drop-down menu for the hierarchy prompt that corresponds to the highest level in your hierarchy. For all subsequent prompts that are at levels below the highest level in your organization **(none)** should be selected.
25. If you are configuring the prompt for the highest level in your organization, click **OK**, and skip the rest of this step. If you are configuring a prompt for a level that is below the highest level in your hierarchy, select the **Dependencies** tab.

    All levels below the highest level are dependent upon higher levels. Click **Add**, and select the level for each level in the hierarchy that is above the current level that you are configuring, beginning with the top level in the hierarchy. For example, consider a hierarchy with three levels, 1, 2, and 3, with 1 being the highest level and 3 being the lowest level. It is important that you add dependencies beginning with the highest level in the organization, and work down to the level right above the current level that you are configuring. Otherwise, the path might not be correct to set up the cascading hierarchy prompts. When configuring the dependencies for level 3, add the prompt for level 1, and then add the prompt for level 2. When you have completed adding dependencies for each level above the current level in the hierarchy, click **OK**.

# *Chapter 8* Retention Analysis

**Introduction to Retention Analysis** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **129**
Retention Analysis in SAS Human Capital Management . . . . . . . . . . . . . . . . . . . 129
The Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130

**Extracting the Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **131**
Overview of Extraction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
Modifying Macro Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
Requirements and Assumptions for Extraction . . . . . . . . . . . . . . . . . . . . . . . . . . 132
Modify the %PREBUILD Macro . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
Modify the %VTAPB Macro . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
The Output Table (HRVANLY2) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 137

**Generating the Scoring Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **138**
Overview of Transformation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
Generate the Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
Generate the Scoring Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
Contents of the Scoring Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145

**Working with the Results** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **146**
Merging the Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146
Using the EMPSCORES Table in a Report . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
Updating the Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 148

## Introduction to Retention Analysis

### *Retention Analysis in SAS Human Capital Management*

Predictive modeling is a way of projecting the values of one or more variables (target variables), based on historical data. A predictive model is a structure and process for predicting those values, using one or more techniques such as linear regression, decision trees, or neural networking.

For SAS Human Capital Management, one rewarding use of predictive modeling is for retention analysis—predicting which employees in an organization are likely to leave within a specified time period. Retention analysis does not predict the date that an employee is likely to terminate; rather, it predicts the likelihood that an employee will terminate within the specified time period.

Using SAS Human Capital Management, consultants and power users can generate reports based on the retention analysis data that correlate various factors with retention probabilities. Such reports might analyze retention probabilities by job group, geographical regions, length of service, or divisions within the organization. Organizations can use this information to plan for the future (for example, to plan additional recruiting efforts in areas that are expected to be affected, or to take corrective action to help reduce the termination rate in high-risk groups).

This document describes the steps involved in retention analysis for SAS Human Capital Management. The document is written for use by HCM consultants, site administrators, or other persons who need to create the utility table that is the input for retention analysis, work with the analytical consultant who creates the retention model, run the code to create the output, or process the results.

*Note:* For more background information about the predictive modeling process, see "It's 9:00am - Do You Know Where Your Critical Talent Is? Retention Analytics and SAS Human Capital Management," a paper from SAS Global Forum 2008 (`http://support.sas.com/rnd/papers/sgf2008/Retentionmodelingpaper.pdf`).

## The Process

At a site with SAS Human Capital Management, the general process for retention analysis is as follows:

1. The extraction stage. You (and perhaps a database administrator) meet with the analytic consultant to examine the site’s data. You refresh the data warehouse, customize some macro variables, and extract the data for the analysis.

   The input comes from several tables. The output of the extraction process is a single table, with one row per employee. At least two years of historical data is required.

   See “Extracting the Data” on page 131. For a list of input tables, see “Required Tables” on page 132.

2. The transformation stage. Using SAS Enterprise Miner, the analytic consultant generates the predictive model. The result is a scoring function that is given to the site.

   At the site, you use the scoring function to generate the Voluntary Termination Scoring table. You wrap the scoring function in a SAS macro, customize a number of macro variables (based on information from the analytic consultant), and run a program that generates the scoring table.

   See “Generating the Scoring Table” on page 138.

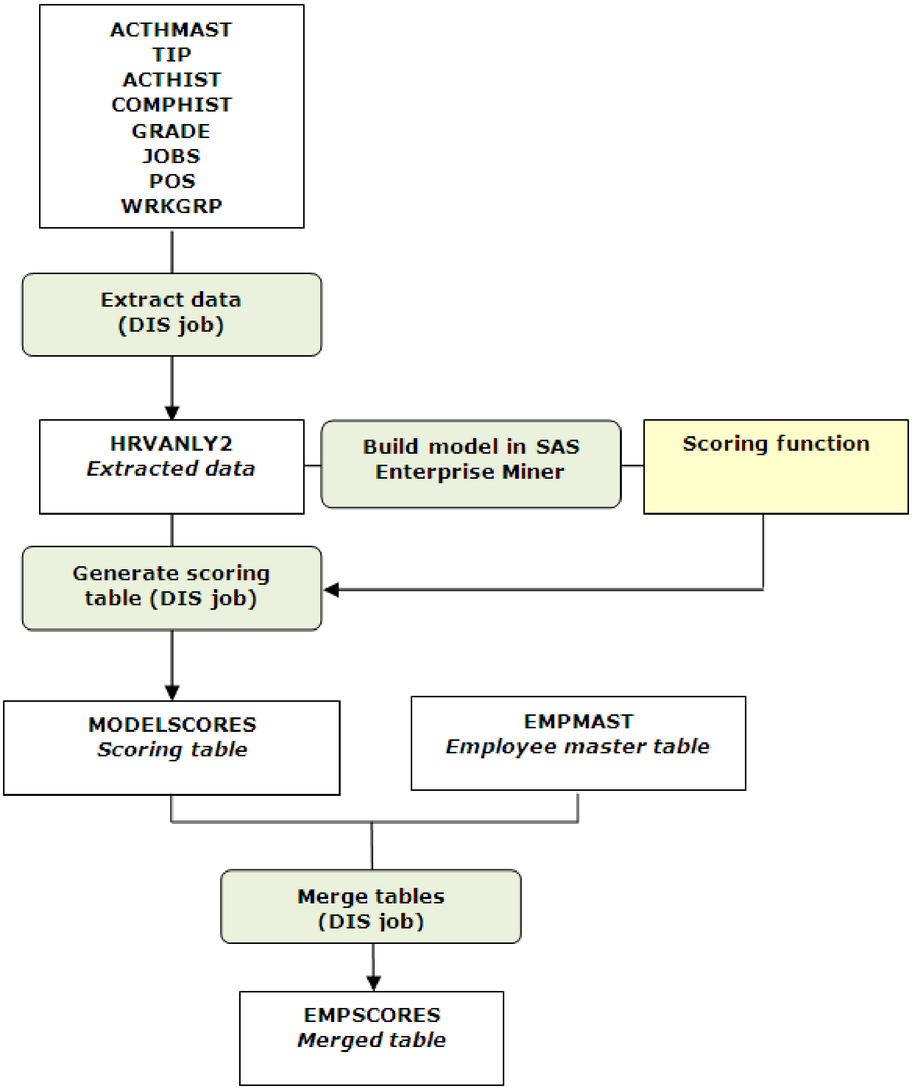
3. The merge stage. You merge the scoring table with the Employee Master table (EMPMAST) to generate a table (EMPSCORES) that combines employee data with predicted termination scores.

   You (or perhaps a power user) create cubes and reports from the merged table.

   See “Working with the Results” on page 146.

The following diagram illustrates the overall retention analysis process:

*Figure 8.1 Retention Analysis Process*



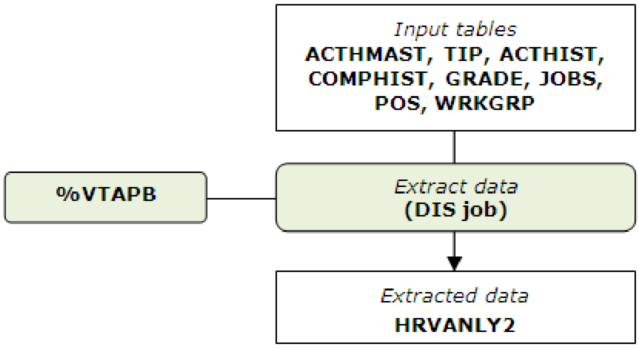
At various times in the future, the site might want to perform a new analysis. With only minor customizations (such as dates) to the macro variables, the site can re-run the entire process and generate a new scoring table, using the existing scoring function.

# Extracting the Data

## *Overview of Extraction*

The extraction stage of the retention analysis process builds the Voluntary Termination Analysis Utility table, HRVANALY2, which provides the input that SAS Enterprise Miner uses in generating the retention model. The following diagram illustrates the extraction stage of the process. Notice the role of the %VTAPB macro, which is called by the SAS Data Integration Studio job that extracts the data.

*Figure 8.2 Extraction Stage of Retention Analysis Process*



By default, the HRVANALY2 utility table is an extraction and summarization from SAS Human Capital Management's headcount summary, time in position, and detail tables. It contains one record per employee. To build the utility table, follow these steps:

1. Examine the requirements for the extraction process and make any necessary changes. See "Requirements and Assumptions for Extraction" on page 132.
2. Refresh the HCM data warehouse using your usual procedures.
3. Modify the %VTAPB macro to customize the process for your site.

   For details, see Table 8.3 on page 135.
4. In SAS Data Integration Studio, run the hcm_140050_load_hrvanly2_table job.

## Modifying Macro Files

The macros that are listed in this chapter are stored in one of the SASAUTOS folders (as defined in the SASV9.CFG file). When you modify one of the macro files (such as vtapb.sas or modlpb.sas), first copy it to the **SASMacro** directory that is located in ***SAS-config-dir*****Lev1\SASApp\SASEnvironment\HumanCapitalManagement** (on the data tier). Then make your modifications in that directory, rather than modifying the original version. At run time, macros in the **SASMacro** folder override the original macros.

Ensure that the operating system permissions for the **SASMacro** directory are the same as the permissions for the **SASApp** directory. Without these permissions, the macros cannot be executed. For details, see the table of "Recommended and Default Operating System Protections" in the *SAS Intelligence Platform: System Administration Guide*.

## Requirements and Assumptions for Extraction

### Required Tables

A minimum of two years of data is required to produce meaningful results. The extraction process relies on these tables from the HCMData library:

- ACTHMAST: Job Actions History Master table
- TIP: Time In Position Summary table
- ACTHIST: Job Actions History Detail table

- POS: Positions History Detail table
- JOBS: Current Job Codes Detail table
- WRKGRP: Work Group Detail table
- COMPHIST: Additional Compensation History Detail table
- EMPGEN: Employee General Information table
- (Optional) One additional table, which you define in the %VTAPB macro.

  If you want to add input from more than one additional table, first merge the tables into a single table with one row per employee, and with employees identified by employee ID. Make sure that you have ongoing access to this data so that it can be used in future predictions.

The EMPMAST (Employee Master) table is not used during the extraction process. However, it is used later to generate a table that merges employee data with scoring data.

### The ACTHMAST Table

ACTHMAST, the Job Action History Master table, must meet the following requirements for retention analysis:

- Voluntary Termination determination must be accessible in DATA step processing against ACTHMAST. The default (as defined in the %VTAPB macro) is:

  ```
  %let vt_crit=(%str(put(ACTION_TYPE_CD,$iaction.)='VTERM'));
  ```

- The ACTHMAST table must contain the following columns:

  ***Table 8.1*** *Required Columns in ACTHMAST Table*

| Column | Description |
|---|---|
| EMPLOYEE_ID | Employee ID |
| ACTION_DT | Action date |
| VALID_TO_DT | Valid to date |
| HIRE_DT | Hire date |
| GENDER_CD | Gender |
| ETHNICITY_CD | Ethnicity |
| BIRTH_DT | Birth date |
| AGE | Age |
| JOB_GROUP_CD | Job group |
| ACTION_TYPE_CD | Action type |
| EMPLOYEE_ACTION_REASON_CD | Action reason |
| PAY_LEVEL_STRUCTURE_CD | Pay level structure |

| Column | Description |
|---|---|
| _LASTREC | Most current record |

### The Temporary Salary History Table

The ACTHIST, POS, JOBS, WRKGRP, EMPGEN, and COMPHIST detail tables are input for a temporary salary history table, WORK.SALHSUM.

The WORK.SALHSUM table must contain any columns that are referenced by the regular full-time criteria (RFT_CRIT) in the %VTAPB macro. You must be familiar with the way the detail tables are used to build SALHSUM. For more information, refer to the code in the %VTAPB, %VTADATA, and %SALHSUM macros.

### The TIP Table

TIP, the Time in Position Summary table, must contain the following columns:

*Table 8.2 Required Columns in TIP Table*

| Column | Description |
|---|---|
| EMPLOYEE_ID | Employee ID |
| VALID_FROM_DT | Valid from date |
| TIP_YEARS | Years in position |
| VALID_TO_DT | Valid to date |

*Note:* If the TIP table is not available, you must comment out any code in the %VTADATA macro that refers to this table. The %VTADATA macro is called by the SAS Data Integration Studio job that extracts the data and loads the HRVANLY2 table.

## Modify the %PREBUILD Macro

The %PREBUILD macro is called by the SAS Data Integration Studio jobs that load the HRVANLY2 table, generate the scoring table, and merge the output. Before running the first of these jobs, you must modify prebuild.sas to reflect the data set date cutoff and the unit of time (such as a year or a quarter) that is used to summarize data.

In the prebuild.sas file, search for the following line:

```
****** Salary History Summary for Predictive Analysis ********
```

Make the following modifications if necessary:

| Variable | Description |
|---|---|
| PDCUTOFF | Last date for data extraction, in the form *DDMMMYYYY*. Must not be a leap year date. |
| PDUNIT | Time unit that is used to summarize data. |

| Variable | Description |
|---|---|
| PDSUMV | (Optional) Variable list to drop from the final table build. |

The default time unit is **`YEAR`**. You can specify other time units, such as **`QTR`** or **`MONTH`**. However, if you choose a time unit other than **`YEAR`**, the predictive analysis keeps only the first summary record in a year. For a complete list and description, see "About Date and Time Intervals" in the *SAS Language Reference*.

The retention analysis code assumes that the time period begins on the first day of the period (for example, January 1 for yearly periods, and January 1, April 1, July 1, and October 1 for quarterly periods).

*Note:* The remainder of this chapter describes yearly time units, to which the WORK.SALHSUM data is converted if its time period is not yearly.

## Modify the %VTAPB Macro

The %VTAPB macro identifies organizational characteristics of the site, including voluntary termination, involuntary termination, promotional criteria, regular full-time employees, and time-dependent analysis columns. It is similar to the %PREBUILD macro in that it enables you to customize data derivation and report creation without modifying the calling program.

By modifying the parameters in the %VTAPB macro, you can customize the derivation and transformation of data, as well as select and format the variables that appear in the reports. Working with the analytical consultant will produce the best results. This macro is used to build the utility table and to build the scoring table.

The following table describes the variables that are defined in the %VTAPB macro.

***Table 8.3*** *Variables in the %VTAPB Macro*

| Variable | Description |
|---|---|
| VT_CRIT | Defines the voluntary termination criteria as a valid DATA step statement that can be processed against ACTHMAST. Example:<br>`%let vt_crit=(%str(put(ACTION_TYPE_CD,$iaction.)='VTERM'));` |
| NVT_CRIT | Defines the involuntary termination criteria as a valid DATA step statement that can be processed against ACTHMAST. Example:<br>`%let nvt_crit=(%str(put(ACTION_TYPE_CD,$iterm.)='TERMS') and`<br>`  %str(put(ACTION_TYPE_CD,$iaction.) ne 'VTERM'));` |

| Variable | Description |
|---|---|
| ASUBVARS | (Optional) Defines any nonstandard variables that are used in the termination criteria and promotion criteria macro variables. “Non-standard” refers to columns that are not listed in Table 8.1 on page 133.<br>For example, assume that the site defines voluntary termination as a combination of ACTION_TYPE_CD and MYREASON (a nonstandard column):<br>`%let vt_crit = %str((put(ACTION_TYPE_CD,`<br>`   $iaction.)=’VTERM’)) and myreason = ’T’);`<br>In that case, you would assign asubvars as follows:<br>`%let asubvars = myreason;` |
| PRO_CRIT | (Optional) Defines promotion criteria as a valid DATA step statement that can be processed against ACTHMAST. Example:<br>`%let pro_crit=(%str(ACTION_TYPE_CD=’DPRO’));` |
| RFT_CRIT | Defines criteria for regular full-time employees as a valid DATA step statement that can be processed against the WORK.SALHSUM table. Example:<br>`%let rft_crit=%str(PERMANENCE_CD="R");` |
| SHAVARS | Defines the time-dependent analysis columns from the WORK.SALHSUM table. (To determine these columns, look at the columns in the permanent SALHSUM table.) Example:<br>`%let shavars=ANNUAL_SALARY TOTOTHER PAY_LEVEL_STRUCTURE_CD`<br>`   CHGAMT CHGPCT JOB_GROUP_CD;` |
| AHAVARS | Defines the time-dependent analysis columns from the ACTHMAST table. Example:<br>`%let ahavars=COMRATIO RNG_PENE;` |
| ASPECDAT | (Optional) Specifies the name of a table that contains custom parameters to be included in the analysis. This table might have been created from survey data or as the result of a query. The format of this data set must correspond to the format of HRVANLY2. Example:<br>`%let aspecdat=hcmdata.mytable;` |
| ASPECVAR | Defines the names of the columns to be analyzed from the optional table that is identified in ASPECDAT. Required if ASPECDAT is defined. Example:<br>`%let aspecvar=myvar02 myvar03 myvar04 myvar05;` |
| MEDVARS | (Optional) Specifies one or more analysis columns to be compared with the median values for the group that is specified in the MEDGRP variable. Example:<br>`%let medvars=CHGAMT RNG_PENE;` |

| Variable | Description |
|---|---|
| MEDGRP | (Optional) Defines the grouping column to be used to create the medians. For example, grouping by job group means that values for a particular employee are compared to the median for that employee's job group, rather than across the organization. Assume this example:<br>`%let medvars=ANNUAL_SALARY CHGPCT;`<br>`%let medgrp=JOB_GROUP_CD;`<br>In this case, ANNUAL_SALARY_CUR_MED would be derived as **0** if its value was below the median value and **1** if its value was at or above the median value of ANNUAL_SALARY for that job group. Naming conventions used are _CUR_ for the current period and _PRI_ for the previous period being analyzed. |

The %ANLYVAR1 macro provides a location for valid DATA step or nested DATA step syntax that can be processed against WORK.SALHSUM. Example:

```
%macro analyvar1;
if ANNUAL_SALARY <50000 and JOB_GROUP_CD in ('A','D','X')
    then newvar=1;
else ...;
%mend analyvar1;
```

The %DERAVARS macro provides a location for valid DATA step syntax that can be processed on WORK.HRVANLY2. Essentially, it allows for post processing of the final data, including additional data derivations or anything else that can be applied in a DATA step.

## The Output Table (HRVANLY2)

### Overview of HRVANLY2

HRVANLY2 is the Voluntary Terminations Utility table that is generated by the SAS Data Integration Studio hcm_140050_load_hrvanly2_table job. It contains the transformed data that is used for reporting and further analysis. The table contains one row per employee, with actual, derived, and summarized variables, on an annual basis.

The annual data is contained in the columns that are defined in the ahavars, anlyvar1, and shavars macro parameters of the %VTAPB macro. As a convention, the two-digit year is appended to the original column name. Essentially, this structure corresponds to transposing the WORK.SALHSUM data by employee, with the year of the effective date used in the name of the transposed parameter. For example, ANNUAL_SALARY becomes ANNUAL_SALARY02, ANNUAL_SALARY03, and so on.

### Annual Effective Dates

The analysis extracts annual effective dates from WORK.SALHSUM at the beginning of the annual period (January 1 for a calendar year). Therefore, all annual explanatory parameters (that is, parameters that are compared to the voluntary termination flag) represent data at the beginning of the annual period of the given year.

The voluntary termination flag is also represented on an annual basis, using the same naming convention. For the voluntary termination flag, the two-digit year defines an annual period that begins one day after the beginning of the year and ends at the beginning of the following year.

For the final analysis, a single overall voluntary termination flag is used (VTERM_F). In each case, the flag has a value of `0` (no) or `1` (yes).

### Current and Prior Year Parameters

Current and prior year parameters are derived from the annual parameters, depending on termination date. For example, assume that the current year is 2008 (and the annual period begins on January 1):

- An active employee's current and prior year ANNUAL_SALARY parameters would be **`ANNUAL_SALARY_CUR=ANNUAL_SALARY08`** and **`ANNUAL_SALARY_PRI=ANNUAL_SALARY07`**, respectively.
- For an employee who terminated between 02JAN2005 and 01JAN2006, the current and prior year ANNUAL_SALARY parameters would be **`ANNUAL_SALARY_CUR=ANNUAL_SALARY05`** and **`ANNUAL_SALARY_PRI=ANNUAL_SALARY04`**, respectively.

## Generating the Scoring Table

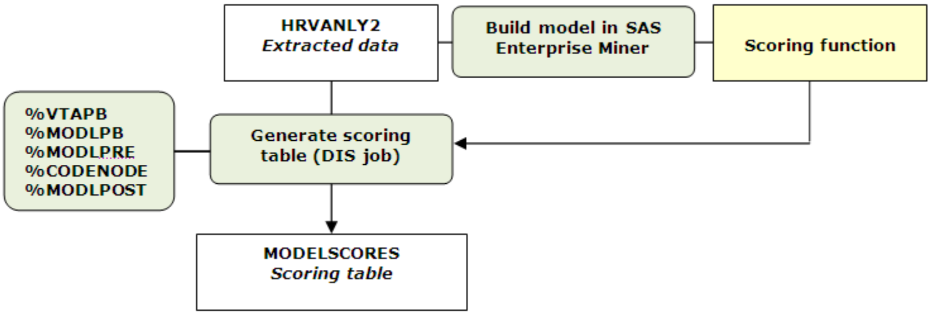
### Overview of Transformation

The transformation stage of the retention analysis process includes these steps:

1. The analytical consultant generates the model in SAS Enterprise Miner. The output is a scoring function: SAS code that you can run in SAS Human Capital Management to generate the scoring table.
2. With advice from the analytical consultant, you incorporate the scoring function in the %CODENODE macro and make additional modifications to the %MODLPB, %MODLPRE, and %MODLPOST macros.
3. You call a SAS Data Integration Studio job to produce the scoring table (MODELSCORES).

This diagram illustrates the transformation stage of the process.

***Figure 8.3*** *Transformation Stage of Retention Analysis Process*



## Generate the Model

After you extract the utility table (HRVANLY2), you hand it off to the analytical consultant. The data contains information about variables that might affect termination as well as information about which employees actually terminated. From this historical data, the analytical consultant defines the target variable (such as a termination flag) and derives additional input variables, such as change in salary or time in position since last promotion.

*Note:* In addition to the utility table, the analytical consultant uses the %RUNMODEL, %VTAPB, %MODLPB, and %MODLPRE macros. You will have already customized the %VTAPB macro in order to create the utility table. The analytical consultant typically makes any necessary customizations to both the %MODLPB and %MODLPRE macros.

Using SAS Enterprise Miner, the analytical consultant creates predictive models using a variety of algorithms (linear regression, decision tree, and neural networks). SAS Enterprise Miner runs these models over the historical data.

As the analytical consultant runs the models, the software generates diagnostic information for analyzing the results to see which model was most accurate in “predicting” the terminations that did occur. After building the model, the analytical consultant uses SAS Enterprise Miner to generate a scoring function—SAS code that can be incorporated into SAS Human Capital Management and used to predict future terminations. The analytical consultant gives this scoring function to the site and advises you about necessary updates to the %MODLPB, %MODLPRE, and %MODLPOST macros.

Working with the site, the analytical consultant subjectively groups the predicted probabilities into low, medium, and high values. (See the definition of the VTCUTS macro variable in Table 8.4 on page 140.) These groupings simplify the results presentation.

## Generate the Scoring Table

### Overview

A SAS Data Integration Studio job produces the MODELSCORES scoring table. Either directly or indirectly, it invokes the following macros:

- **%RUNMODEL** calls the remaining macros and then registers the MODELSCORES table in the HCMData library, with HonorTableSecurity set to `Y`.
- **%VTAPB** contains customizable macro variable assignments.
- **%MODLPB** contains customizable macro variable assignments.
- **%MODLPRE** contains preprocessing code for the scoring function.
- **%CODENODE** acts as a wrapper for the scoring function code from SAS Enterprise Miner.
- **%MODLPOST** contains postprocessing code for the scoring function.

After you receive the scoring function from the analytical consultant, follow these steps to generate the scoring table:

1. With input from the analytical consultant, customize the %MODLPB macro for your site and data. (See “Customize the %MODLPB Macro” on page 140.)

You will have already modified the %VTAPB macro during the extraction process. For details, see "Modify the %VTAPB Macro" on page 135.

2. Make any necessary customizations to the %MODLPRE macro.

   Typically, the analytical consultant makes any necessary additions to this code. For details, see "Customize the %MODLPRE Macro" on page 143.

3. Paste the scoring function code from SAS Enterprise Miner into the %CODENODE macro and make minor customizations.

   See "Wrap the Scoring Function" on page 143.

4. Make any necessary customizations to the %MODLPOST macro.

   For details, see "Customize the %MODLPOST Macro" on page 144.

5. In SAS Data Integration Studio, run the hcm_140100_load_modelscores_table job.

The output, which is generated by the %MODLPOST macro, is MODELSCORES, the Voluntary Termination Scoring table. For details about this table, see "Contents of the Scoring Table" on page 145.

### Customize the %MODLPB Macro

The %MODLPB macro enables you to customize the data transformation without modifying the calling program. (In that respect, it is similar to the %PREBUILD macro.) To perform the analysis, the analytical consultant uses the default values. For producing scores data, you modify this macro with advice from the analytical consultant.

The following table lists the variables that are defined in the %MODLPB macro.

***Table 8.4*** *Variables in the %MODLPB Macro*

| Variable | Description |
|---|---|
| HRVA2LOC | Defines the location of the HRVANLY2 source table. For SAS Human Capital Management, this value should be set as follows:<br>`%let hrva2loc=&hcmlib;`<br>In a stand-alone SAS session, the analytical consultant has the option of specifying a LIBNAME path to this library. |
| PRE_PROG | Specifies the name of the macro (MODLPRE) that further transforms the HRVANLY2 table before processing in SAS Enterprise Miner. |
| BEGINYR | Defines the beginning year of the modeling observation period. The ENDYR macro variable defines the beginning year of the modeling prediction period. The analytical consultant, based on the client's data, determines the initial year in which predictive parameters are observed. It should be no fewer than 2 years before ENDYR, the prediction period.<br>Typically, the observation period begins 2–4 years before the current year (ensuring at least 2 years of observations), and the prediction period begins 1 year before the current year. For example, if the current year is 2008, then the observation period might begin in 2004 (BEGINYR) and the prediction period might begin in 2007 (ENDYR).<br>Example:<br>`%let beginyr=2004;` |

| Variable | Description |
|---|---|
| ENDYR | Defines the initial prediction year, which is typically the previous year (that is, January 2 of last year through January 1 of this year).<br>If you leave the value of ENDYR blank, it defaults to the previous year (**`year(today())-1`** ). Example:<br>`%let endyr=;`<br>Using the default simplifies future maintenance. Otherwise, you must update the ENDYR value each year.<br>If more than one year is being used for prediction, enter the initial year for the prediction period as the value of ENDYR, and calculate appropriate derivation for a custom voluntary termination parameter in the %MODLPRE macro.<br>A common example is a prediction period of “last year” through “available data this year.” In that case, you would leave ENDYR blank (signifying “last year”), and you would derive a custom parameter like this in the %MODLPRE macro:<br>`if vterm_f=1 and lstcondt gt mdy(1,1,&endyr.)`<br>`  then vterm_use=1;`<br>`  else vterm_use=0;`<br>In this example, VTERM_F and LSTCONDT are standard variables from the HRVANLY2 table. You would enter VTERM_USE as the value for the VTERMVAR parameter. |
| MODELSUB | Defines a population subset for the analysis. For model creation, the typical assignment is as follows:<br>`%let modelsub=where HIRE_DT le`<br>`  mdy(1,1,%eval(%sysfunc(year(%sysfunc(today())))-1));`<br>That statement subsets the data so that only employees who were active during the entire modeling observation period are used to develop the statistical model.<br>For producing scores data, this subset should be changed in order to derive predicted voluntary termination probabilities for as many employees as possible. The following subset should be sufficient in most situations. It allows voluntary termination probabilities to be derived for employees who were active for only a portion of the observation period:<br>`%let modelsub=where HIRE_DT le mdy(1,1, &endyr.);`<br>(Only full-time employees who did not involuntarily terminate are included in the HRVANLY2 data.) Because this parameter is called in a DATA step in the %MODLPRE macro, any valid DATA step code can be used (for example, the WHERE statement could be replaced by an IF statement). |
| PRE_DATA | Defines the name of the output table to be created from the %MODLPRE preprocessing macro. Typically, this is a WORK file. The default is HRVANLY3. Example:<br>`%let pre_data=HRVANLY3;` |
| EM_PROG | Defines the name of the macro that contains the scoring program that was generated by SAS Enterprise Miner (**`codenode`**). |

| Variable | Description |
|---|---|
| _SCORE | Defines the input table for the SAS Enterprise Miner program. The default is as follows:<br>`%let _score=&pre_data.;` |
| _PREDICT | Defines the output table from the SAS Enterprise Miner program. The default is as follows:<br>`%let _predict=&pre_data.;` |
| POST_PROG | Defines the name of the postprocessing macro (**`modlpost`**). After the SAS Enterprise Miner code processing., the %MODLPOST macro performs the final data transformations (for example, it calculates the voluntary termination probabilities). |
| BODDS, POINTS, and BASE | These parameters are defined by the analytical consultant. They control the derivation of a score from the voluntary termination probability. |
| VTERMVAR | Defines the name of the variable that identifies the voluntary termination flag for the model. The default value is **`vterm_f`**, which represents the previous year's voluntary terminations (for example, from January 2 last year until January 1 this year). The variable contains values of **`0`** (no) or **`1`** (yes). |
| VTGRP | Defines the method by which the risk of voluntary termination is categorized into low, medium, and high groups. The postprocessing macro contains the derivation, based on this parameter, which can have one of these possible values:<br>• **`PROB`** (the default) bases the categorization on the voluntary termination probabilities.<br>• **`SCORE`** bases the categorization on the scores that are derived from the voluntary termination probabilities.<br>• **`RANK`** bases the categorization in the RANK procedure, which divides the voluntary termination probabilities into three equal groups. |
| VTCUTS | Defines the cutoff values for either the PROB or SCORE methods that were described for the VTGRP variable. Either the site or the analytical consultant must define what is considered as a low, medium, or high risk of voluntary termination. Specify two decimal values, separated by a space, representing the upper end of the low range and the upper end of the medium range (inclusive).<br>This example (for the PROB method) defines low risk as less than or equal to 10%, and high risk as greater than 25%:<br>`%let vtcuts = 0.1 0.25;`<br>This example (for the SCORE method) defines low risk as a score less than or equal to 300, and high risk as a score greater than 400:<br>`%let vtcuts = 300 400;` |
| TRANSVARS | Contains the names of the categorical transform variables that are generated by SAS Enterprise Miner, so that these variables can be used later in reports. The analytical consultant supplies these names. |

| Variable | Description |
|---|---|
| MODELVARS | Defines the names of the statistically significant variables that are produced by SAS Enterprise Miner, so that you can use these variable names in further processing. The analytic consultant provides this list. SAS Enterprise Miner re-creates properly formatted variables from the analysis variables supplied. Example:<br>`%let modelvars=CHGAMT_CUR RNG_PENE06 YOS_ACT;` |

A PROC FORMAT statement defines the format for the three voluntary termination risk groups. Numeric values of **0**, **1**, and **2** are derived in the post processing macro (%MODLPOST). Example:

```
proc format;
value vtgroup 0='Low'
              1='Moderate'
              2='High';
run;
```

*Note:* An analytical consultant who is working in a stand-alone SAS session can specify the directory path and filename for the macros that are listed in this file, if the macro is not stored in a defined autocall library. For example, the path to the %MODLPRE macro might be **C:\myhcm\modlpre.sas**.

### Customize the %MODLPRE Macro

The %MODLPRE macro transforms the HRVANLY2 table for use by SAS Enterprise Miner. This macro uses values that are defined in the %VTAPB macro. The default transformations include a macro to rename the annual time-dependent parameters, based on the year of the prediction period. For example, if the prediction period year were 2007, then ANNUAL_SALARY05, ANNUAL_SALARY06, and ANNUAL_SALARY07 would be renamed as ANNUAL_SALARY_0, ANNUAL_SALARY_1, and ANNUAL_SALARY_2, respectively.

The macro also calls the population subset, which is defined in the MODELSUB macro variable in the %MODLPB macro.

Typically, you do not need to modify this code. Custom data transformations, which are usually added by the analytical consultant, should go below the commented area near the bottom of the default code.

### Wrap the Scoring Function

The scoring function code that is produced from SAS Enterprise Miner must be included in the %CODENODE macro, as follows:

1. Open the codenode.sas file for editing. Initially the %CODENODE macro contains these lines:

   ```
   %macro codenode;
   /**** PASTE ENTERPRISE MINER SCORE CODE HERE ****/
   %mend codenode;
   ```

2. Immediately after the **/**** PASTE ENTERPRISE MINER SCORE CODE HERE ****/** line, add the following statements:

   ```
   DATA &_PREDICT;
   SET &_SCORE;
   ```

3. Immediately after those two statements, paste the scoring function code. The analytical consultant will furnish this code.
4. (Optional) If the scoring function uses a logistic regression model, you can make additional changes to the scoring code and extract more in-depth data about the score. This data will tell you how much each input variable contributes to the score for each individual employee. With this data, you can create additional reports to provide information about which factors are increasing (or decreasing) employees likelihood of leaving. To capture this optional data, make the following additional changes to the code:
   a. Near the end of the file, find the section entitled **`Compute Linear Predictor`**.
   b. An example codenode.sas file is included with the SAS Human Capital Management sample data. Using this file as a model, add new code to capture each linear predictor. (These variables are listed in the MODELVARS macro variable in the %MODLPB macro.) We recommend that you name each new variable *original-variable*__**`LP`**. Here is an example for the CHGAMT_CUR variable:

      ```
      *** Effect: CHGAMT_cur;
      _TEMP = CHGAMT_cur;
      _LP0 = _LP0 + ( 0.01772224338353 * _TEMP);

      *** CHANGED HERE: Added following line;
      chgamt_cur_LP = _LP0;
      ```

   c. Scale the individual linear predictor variables: Using the sample codenode.sas file as a model, add new code to scale each linear predictor.

      We recommend that you name each new scaled variable as *original-variable*__**`SC`**. Here is an example for the CHGAMT_CUR variable:

      ```
      chgamt_cur_SC = round(chgamt_cur_LP*(X) + SHIFT, .001);
      ```

   d. Add the scaled variables together to create the voluntary termination score, as in this example:

      ```
      VTSCORE = round(chgamt_cur_SC + rng_pene06_SC + yos_act_SC, 1);
      ```

5. Save your changes.

*Note:* The %CODENODE macro is called by the %RUNMODEL macro. Make sure that the %CODENODE macro does not also contain a %CODENODE macro call. If it does, the macro is executed twice and does not produce the correct output.

### Customize the %MODLPOST Macro

The %MODLPOST macro applies final data transformations to the model that was created in SAS Enterprise Miner and produces the MODELSCORES table, which is registered in the HCM database. It also updates the SAS_HCMFORMATS table in the HCM database and the SAS formats catalog.

For customer implementations, update the %MODLPOST macro as advised by the analytical consultant. This DATA step within modlpost.sas generates the percentage measures for reporting. An example of this code follows, for model variables that are named CHGAMT_CUR, RNG_PENE06, and YOS_ACT.

```
*** Create Permanent Data ***;
data &hcmlib..modelscores;
  set modelscores;
```

```
  /* add percentage measures for WEBEIS reporting */
  attrib chgamt_cur_PERCT length=8. label = '% Due to Salary Change';
  attrib rng_pene06_PERCT length=8. label = '% Due to Range Penetration';
  attrib yos_act_PERCT    length=8. label = '% Due to Service Length';
  attrib TOT_PERCT        length=8. label = '% Due to All Factors';

  chgamt_cur_PERCT = (chgamt_cur_SC / VTSCORE);
  rng_pene06_PERCT = (rng_pene06_SC / VTSCORE);
  yos_act_PERCT = (yos_act_SC / VTSCORE);
  TOT_PERCT = sum(chgamt_cur_PERCT,rng_pene06_PERCT,yos_act_PERCT);

  format vtgroup vtgroup.;

  *** Call Modeling Variable Labels ***;
  %model_lbls;

run;
```

## Contents of the Scoring Table

A SAS Data Integration Studio job calls the %MODLPOST macro to generate the Voluntary Termination Scoring Table (MODELSCORES). This table lists the default columns in the scoring table:

***Table 8.5*** *Default Columns in the MODELSCORES Table*

| Column | Type | Description |
|---|---|---|
| EMPLOYEE_ID | Character | ID Attributes obtained from warehouse. |
| HIRE_DT | Date | Attributes obtained from warehouse. |
| LSTCONDT | Date | Derived as most recent contact date from ACTHMAST and WORK.SALHSUM. |
| &VTERMVAR | Numeric | Voluntary termination flag (0=no; 1=yes). The variable name is defined as the value of VTERMVAR in the %MODLPB macro. |
| P_&VTERMVAR.1 | Numeric | The derived predicted probability to voluntarily terminate. The variable name depends on the VTERMVAR macro variable, as described above. |
| VTSCORE | Numeric | Subjectively derived score based on the voluntary termination probability. |
| VTGROUP | Numeric | Categorization of predicted termination probabilities (0=low, 1=moderate, 2=high). |

| Column | Type | Description |
|---|---|---|
| *Statistically significant variables* | Numeric | All variables that are found to be statistically associated with voluntary termination. The scorecard function renames these variables and derives their value from variables in the HRVANLY2 table.<br>In order to appear in the MODELSCORES table, these variable names must appear in the definition of the MODELVARS or the TRANSVARS macro variable in the %MODLPB macro. |
| *Percentage measures* | Numeric | Percentage measures as computed in the %MODLPOST macro. |

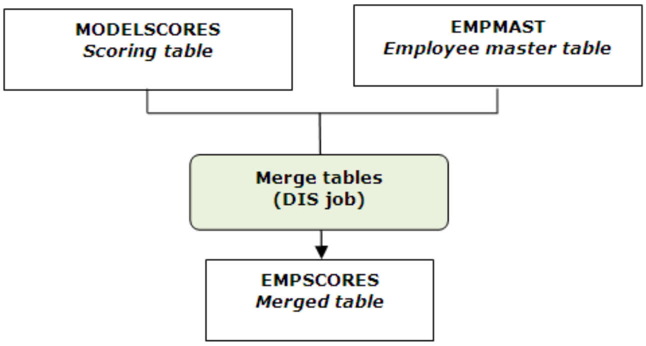
# Working with the Results

## Merging the Tables

### Overview

After you generate the Voluntary Termination Scoring table (MODELSCORES), you need to merge the table with the Employee Master table. This diagram illustrates the merge stage of the process:

**Figure 8.4** *Merge Stage of Retention Analysis Process*



The result of the merge is the Voluntary Termination and Employee Information Scoring table (EMPSCORES), which can be used in reports, in data explorations, in the Employee Browser, and in an organization analysis.

### Merge the Scoring Table and the Employee Master Table

A SAS Data Integration Studio job merges the MODELSCORES table with the Employee Master table (EMPMAST). The output is the EMPSCORES table. By default, three columns from the scoring table are kept: VTSCORE, VTGROUP, and the column specified by the voluntary termination flag (P_&TERMVAR.1). All columns from the Employee Master table are kept.

Follow these steps:

1. If necessary, modify the %VTSCORES macro, which is called by the SAS Data Integration Studio job.

2. In SAS Data Integration Studio, run the hcm_140150_load_empscores_table job. This job loads the EMPSCORES table in the HCM database, using code similar to the following:

   ```
   proc sql noprint;
        create table &hcmlib..empscores (drop=hcm_uniq_id) as
            select b.*, a.vtscore, a.vtgroup, a.p_&vtermvar.1
            from modelscores as a, EMPMAST as b
            where a.EMPLOYEE_ID = b.EMPLOYEE_ID;
   quit;
   ```

   It also registers the EMPSCORES table in the metadata repository, with HonorTableSecurity set to **Y**.

3. By default, the EMPSCORES table has entries in the SAS_HCMMETATABLE and SAS_HCMMETACOLUMN tables, with default formats for the standard columns. Modify the formats for the new columns in the EMPSCORES table. To make these changes in the Administration application:

   a. Log on to SAS Human Capital Management as a user with the HCM Administrator role.

   b. Click **Administration**.

   c. On the **Data** tab, select **Tables** ⇨ **EMPSCORES**.

   d. Modify the formats for the new table columns. For details, see "Modify Column Attributes" on page 14.

4. On the **Security** tab of the Administration application, create row-level filters for the EMPSCORES table. Create a filter for each of the roles (HCM Administrator, HCM Analyst, HCM User) that you want to have access to the EMPSCORES table. Set security appropriately for each role.

   For an overview of table security and detailed instructions, see "Securing Table Rows" on page 74.

### Create a Cube from the EMPSCORES Table

(Optional) Create a cube from the EMPSCORES table:

1. Log on to SAS Human Capital Management as a user with the HCM Administrator role.

2. Click **Administration**.

3. On the **Data** tab, select **New Cube**. For detailed instructions, see "Working with Cubes" on page 25.

## Using the EMPSCORES Table in a Report

After you make the EMPSCORES table available for use in SAS Human Capital Management, users can perform tasks such as the following:

*Note:* Security settings can restrict access to the table or the ability to perform certain tasks.

- select the table for display in the Employee Browser.
- use the table as the basis for an organization analysis or geographic analysis.
- create an information map from the table or cube.
- in SAS Web Report Studio, create a report that is based on the information map.

## Updating the Results

At various times in the future, the site might want to regenerate the MODELSCORES and EMPSCORES tables, based on new data. After refreshing the HCM data warehouse and updating the macros appropriately, you can rerun the jobs using the original code from SAS Enterprise Miner. (Typically, this code is useful for about two years. After that time, the code might need to be modified by the analytical consultant to reflect changes in the organization or other factors.)

# *Chapter 9*
# SAS for Workforce Planning & Budgeting

**About SAS for Workforce Planning & Budgeting . . . . . . . . . . . . . . . . . . . . . . . . 149**

**Administering SAS for Workforce Planning & Budgeting . . . . . . . . . . . . . . . . . . 150**

**Creating Planning Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150**
About Planning Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
Measures for Creating New Positions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
Measures for Supplemental Schedules . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
Measures That Depend on a Slicer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
Define a Planning Measure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
Define SAS Code for Target Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
Draw Information from Additional Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155

**Creating a Form Set with Supplemental Schedules . . . . . . . . . . . . . . . . . . . . . . . 156**
Designing a Form Set Template . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
Creating a Supplemental Schedule . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
What Happens at Run Time . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158

**Managing Plans . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158**

## About SAS for Workforce Planning & Budgeting

With SAS for Workforce Planning & Budgeting, HCM data is made available to the SAS Financial Management budgeting workflow. In a data-entry form, users have access to data from SAS Human Capital Management tables. This data, which is displayed in additional tables (supplemental schedules), might include details such as salary, bonus, travel expenses, and evaluations for each employee. In bottom-up workflows, data from the supplemental schedules is rolled up to the data-entry table and stored in SAS Financial Management.

This chapter contains an overview of the administrative process and details that are specific to SAS Human Capital Management. For detailed information about creating a form set with supplemental schedules, see the online Help for the SAS Financial Management Add-In for Microsoft Excel. For an overview of the end-user experience, see “Participating in the Budgeting Process” in the *SAS Human Capital Management: User's Guide*.

# Administering SAS for Workforce Planning & Budgeting

The basic administration process is as follows:

1. In the HCM Administration application, an HCM administrator defines one or more planning measures. Depending on their definition, these measures might be columns in a supplemental schedule or prompts in a New Position dialog box.
2. In SAS Financial Management Studio, a finance process administrator creates a form set.
3. In Microsoft Excel, the finance process administrator opens the form template and inserts one or more data-entry tables and supplemental schedules.
4. The finance process administrator edits the template (for example, applying formulas, headings, and styles).
5. The finance process administrator publishes the form set.

# Creating Planning Measures

## *About Planning Measures*

When finance process administrators create a form set template, they can insert two types of measures: FM measures and planning measures.

Planning measures are defined on the **Data** tab of the SAS Human Capital Management Administration application. Their values can come from HCM tables, from an end user's data entry, from a calculation, or from executing SAS code.

There are two types of planning measures:

- measures that can appear in a new-position dialog box. If an end user expects to be hiring new employees during the budget period, he can create one or more new positions as placeholders. These measures (such as job group and salary) are used to define the new position.
- measures that can appear in a supplemental schedule.

## *Measures for Creating New Positions*

At run time, during data entry, a user can define one or more anticipated positions. In the New Position dialog box, the following types of measures can appear:

- **Prompts** are text boxes in which the user types a value. For example, the user might be prompted to enter the required years of experience for a new position. Prompts appear only in aNew Position dialog box.
- **Lookups** appear as a drop-down list from which the user can select a value (such as job group). Lookups are populated by the source table and column (to a maximum of

100 distinct values). Like prompts, lookups appear only in a New Position dialog box.

When the user clicks **OK** to close the dialog box, SAS code is executed. It uses the values from these two field types to populate columns (target measures) in the supplemental schedules.

### *Measures for Supplemental Schedules*

The following types of measures can appear only in a supplemental schedule:

- **Target measures** are the targets of the SAS code that is executed when an end user creates a new position. If you want the user to be able to override the value that the SAS code provides, make the measure editable for new employees.

  Target measures can also contain information for existing employees. To display data from an HCM table, specify a source table and column and make the measure editable only for new employees. To enable the user to enter a value, leave the source table and column blank and make the measure always editable.

- **None** identifies a measure that appears only in a supplemental schedule but that is not a target measure. These measures might be used for comments or for reference.

  As an example of a reference measure, suppose that you want to create a planning measure with information that managers might use in calculating salary increases or bonuses. For example, you might want the planning form to display the number of years an employee has been working at the company. You create a measure called Tenure, which takes its value from the SERVICE_YEARS column of the Employee Master table (EMPMAST). Tenure has no meaning for new employees, and you do not want this measure to appear in the New Position dialog box. You set the **Field Type** to **None**, and you set **Editable** to **Never**, because this measure is for reference only.

### *Measures That Depend on a Slicer*

Measures can be designated as slicer-dependent or slicer-independent.

- **A slicer-dependent measure** changes its value when the slicer's value changes. A typical slicer-dependent measure might contain employee expense information that depends on a particular account. In a budgeting form, as a user changes the account slicer, a different value appears in the employee expense cells.

  Other slicer-dependent measures might be associated with a time period or another dimension.

- **A slicer-independent measure** does not change value when the slicer changes. A typical slicer-independent measure might contain a value such as an employee's start date or date of birth.

For existing employees, measures that are based on the standard HCM tables are always slicer-independent.

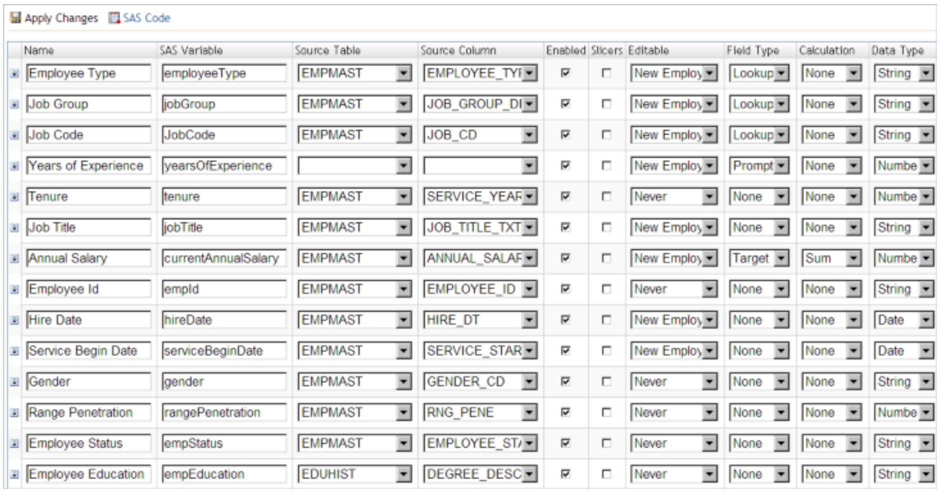
For new employees, measures that are based on the standard HCM tables can be designated slicer-dependent. In that case, the plan tables store separate values for the measure for each dependency.

If a measure is not based on a table (that is, if you leave both the **Source Table** and **Source Column** boxes empty), then the measure can be designated slicer-dependent and can contain information for both new and existing employees.

## Define a Planning Measure

You define planning measures in the Administration application of SAS Human Capital Management. Those measures are then available when you are designing the form set in SAS Financial Management.

To define a planning measure, log on to SAS Human Capital Management as an HCM administrator. In the Administration application, click the **Data** tab and select **Planning** ⇨ **Planning Measures**. The planning measures page displays measures that have already been defined.



| Name | SAS Variable | Source Table | Source Column | Enabled | Slicers | Editable | Field Type | Calculation | Data Type |
|---|---|---|---|---|---|---|---|---|---|
| Employee Type | employeeType | EMPMAST | EMPLOYEE_TYI | ☑ | □ | New Employ | Lookup | None | String |
| Job Group | jobGroup | EMPMAST | JOB_GROUP_DI | ☑ | □ | New Employ | Lookup | None | String |
| Job Code | JobCode | EMPMAST | JOB_CD | ☑ | □ | New Employ | Lookup | None | String |
| Years of Experience | yearsOfExperience | | | ☑ | □ | New Employ | Prompt | None | Numbe |
| Tenure | tenure | EMPMAST | SERVICE_YEAF | ☑ | □ | Never | None | None | Numbe |
| Job Title | jobTitle | EMPMAST | JOB_TITLE_TXT | ☑ | □ | New Employ | None | None | String |
| Annual Salary | currentAnnualSalary | EMPMAST | ANNUAL_SALAF | ☑ | □ | New Employ | Target | Sum | Numbe |
| Employee Id | empId | EMPMAST | EMPLOYEE_ID | ☑ | □ | Never | None | None | String |
| Hire Date | hireDate | EMPMAST | HIRE_DT | ☑ | □ | New Employ | None | None | Date |
| Service Begin Date | serviceBeginDate | EMPMAST | SERVICE_STAR | ☑ | □ | New Employ | None | None | Date |
| Gender | gender | EMPMAST | GENDER_CD | ☑ | □ | Never | None | None | String |
| Range Penetration | rangePenetration | EMPMAST | RNG_PENE | ☑ | □ | Never | None | None | Numbe |
| Employee Status | empStatus | EMPMAST | EMPLOYEE_ST/ | ☑ | □ | Never | None | None | String |
| Employee Education | empEducation | EDUHIST | DEGREE_DESC | ☑ | □ | Never | None | None | String |

To define a new measure, follow these steps:

1. In an empty row, complete these fields:

   **Name**
   Enter a unique name for this measure. The name is used as a column label in the supplemental schedule. It is the value that you see when you are selecting measures for a supplemental schedule.

   **SAS Variable**
   Give this measure a variable name (following the rules for SAS variable names). The variable name is used in SAS code that provides a value for target measures, and it must be unique in the list of planning measures. For more information, see "Define SAS Code for Target Measures" on page 154.

   **Source Table**, **Source Column**
   (Optional) Select the HCM table and column that will provide a value for this measure. These measures are not editable for existing employees.

   **Enabled**
   Select this check box to make the measure available for use in a planning form. You might leave a measure disabled temporarily, until you have finished defining it.

   **Slicers**
   Select this check box to specify that the measure is slicer-dependent. (See "Measures That Depend on a Slicer" on page 151.)

   **Editable**
   Specifies whether this measure can be modified. Select one of the following values:

- **Never** applies to historical measures (for example, last year's salary) and other measures that should not be changed by the user.
- **Always** applies to measures that need user input, such as planned bonus percentage.
- **New Employee** applies to measures that require user input only for new employees.

*Note:* If a measure has a source table and column, you can select **Never** or **New Employee**, but not **Always**.

**Field Type**

The field type determines the user interface for this measure in the New Position dialog box. Select one of the following values:

- **None** specifies a measure that does not appear in the New Position dialog box and is not a target measure.
- **Prompt** creates a text-box prompt in the New Position dialog box. Prompts do not appear in the supplemental schedule.
- **Lookup** creates a drop-down list in the New Position dialog box, using values from the source table (for example, job group descriptions). Lookups do not appear in the supplemental schedule.
- **Target** specifies a measure that appears only in the supplemental schedule, not in the New Position dialog box. For new positions, the value of a target measure is based on other measures in the New Position dialog box. When the user clicks **OK** to close the dialog box, SAS code is executed to provide the value for this measure. (See "Define SAS Code for Target Measures" on page 154.)

**Calculation**

The **Calculation** drop-down list defines a calculation that takes place on this measure at run time. (This option applies only to numeric measures.) Select one of the following values:

- **None** specifies no calculation.
- **Sum** specifies to create a sum for this measure.
- **Averaged** specifies to create an average for this measure.

**Data Type**

Select one of the following data types:

- **String** specifies a character string.
- **Number** specifies a numeric value.
- **Date** specifies a date field. If the **Field Type** is **Prompt**, the New Position dialog box contains a pop-up for date selection.

2. To add another measure, click **Add Row**.
3. Click **Apply Changes**.

Be sure to apply row-level security and column-level security appropriately to any tables that provide values for the planning measures.

Measures for employee IDs or employee names are handled differently than other measures:

- The New Position dialog box automatically has a prompt for the employee name. It never has a prompt for the employee ID, which is automatically generated and cannot be edited.
- If you create measures that get their value from the EMPLOYEE_NAME or EMPLOYEE_ID column, those measures are not editable in the supplemental schedule, and they do not appear in the New Position dialog box.
- To change the employee name for a new position, right-click the employee name and select **Change Description**.

## *Define SAS Code for Target Measures*

### *Overview*

On the Planning Measures page, click **SAS Code** to open a dialog box for code that is applied at run time to compute target planning measures. For example, you might want to calculate projected salary for new employees, based on typical pay for their level of experience and the type of job that they will be doing.

There is a single instance of the SAS code for target measures that applies to all supplemental schedules. If a form set does not use a particular target measure, that part of the code is ignored. The code is run when an end user clicks **OK** in the New Position dialog box, to add one or more new employees to a supplemental schedule. It does not apply to existing employees.

Here is an example of the contents of the SAS Code window:

```
%hcmlib;

/*
New Employee Suggested Salary Calculation
*/
%nuempsal(nexp=&yearsOfExperience, jobgrp=&jobGroup,
   emptype=&employeeType);

/*
New Employee Suggested Bonus Calculation
*/
%nuempbns(jobgrp=&jobGroup);

/*
New Employee Suggested Travel Calculation
*/
%nuemptvl(jobgrp=&jobGroup);
```

The SAS code has three parts:

1. First is a call to %HCMLIB, which sets the locale macro variables and contains the LIBNAME statement for the HCM database.
2. It is followed by code that produces values for the target planning measures (also known as target variables). For the sake of readability and maintenance, we recommend that you define a separate macro for each target variable. (See “Example Macro Code” on page 155.)

   If you look at the first of these macros, notice that it references three input macro variables:

   ```
   %nuempsal(nexp=&yearsOfExperience, jobgrp=&jobGroup, emptype=&employeeType);
   ```

These input variables come from the New Position dialog box, from measures that have a field type of **Lookup** or **Prompt**.

3. Optionally, you can add statements to write the variables to the log.

   *Note:* The workspace server log from the most recent operation is stored in the SAS_HCM_PLAN_SAS_CODE table in the HCM database.

When you finish updating the SAS Code dialog box, click **OK** to save your changes. (If the **OK** button is not visible, press the TAB key on your keyboard.)

### Example Macro Code

This sample macro (%NUEMPSAL) generates a suggested salary for a new employee based on the average annual salary of other employees with the same job group and employee type and with similar years of experience. You can treat SAS variables for the planning measures just as you would any other macro variable. In this case, the SAS variable for the target planning measure is **`currentAnnualSalary`**.

```
%macro nuempsal(nexp=, jobgrp=, emptype= );
   proc summary data=&hcmlib..empmast (where=
        (job_group_desc=&jobgrp and
          strip(put(employee_type_cd, $emptype.))=&emptype and
          service_years >=(&nexp-1) and service_years <=(&nexp+1)));
        var annual_salary;
        output out=newempsal mean=annual_salary;
   run;

   data _null_; set newempsal;
      call symput("currentAnnualSalary",round(annual_salary, 100));
   run;
%mend;
```

*Note:* If a lookup measure is from a formatted column, the formatted value is passed to the SAS code. The SAS code should include an appropriate INFORMAT statement.

The sample macros (%NUEMPSAL, %NUEMPBNS, and %NUEMPTVL) are stored in the **`!SASROOT\hrds\macros`** directory on the data tier. A site must modify these macros, or add other macros, to support the target measures that are defined for the site. Store any custom macros in the following location on the data tier: ***`SAS-config-dir`*** **`\Lev1\SASApp\SASEnvironment\HumanCapitalManagement\SASMacro`**.

## Draw Information from Additional Tables

If a site has additional tables or spreadsheets with information that would be helpful in the budgeting process, you can include those tables as input for the planning measures, as follows:

1. If the table will be used to populate a field in the supplemental schedule, make sure that the table contains an EMPLOYEE_ID column.

   If the table will be used only as a lookup in the New Position dialog box, the EMPLOYEE_ID column is not necessary.

2. On the **Data** tab of the Administration application, import the table to SAS Human Capital Management.

   Make sure that **Register table in metadata repository** is selected.

3. Click **Refresh Cache**.

4. Define planning measures that use columns from the table.

# Creating a Form Set with Supplemental Schedules

## Designing a Form Set Template

The online Help for the SAS Financial Management Add-In for Microsoft Excel has detailed instructions about creating form sets with supplemental schedules. Here are some points to keep in mind for SAS Human Capital Management:

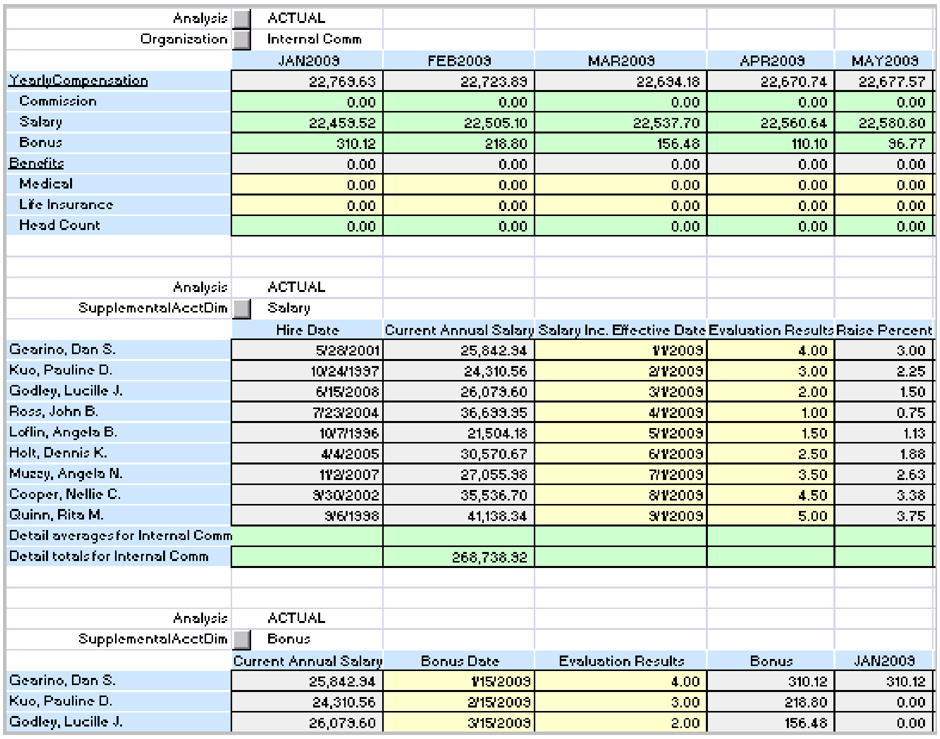
- Both bottom-up workflows and top-down workflows are supported in SAS for Workforce Planning & Budgeting.
- The model for data-entry forms must reference a financial cycle, not an operational planning cycle. The model must include the hierarchy that you plan to use as the target hierarchy in the form set. It must be an Organization hierarchy that has the same members in both SAS Human Capital Management and SAS Financial Management.
- Design the data-entry table carefully, to contain only the information that is needed. When you insert a supplemental schedule, you select its fields from the set of leaf members in the FM data-entry table and the set of HCM planning measures. This selection will be easier if you limit the members in the data-entry table.
- Do not copy or import a form set that contains a supplemental schedule. You must create a new form set instead.
- When you create a form set, you are prompted for the default currency, which applies only when you are editing the form template. It should match the currency that was configured for SAS Human Capital Management.

## Creating a Supplemental Schedule

You can insert one or more supplemental schedules in a form set template. A typical implementation is to design a data-entry form with the Time dimension in the columns, the Account dimension in the rows, and the Organization dimension as a slicer. Then you might insert multiple supplemental schedules, using the Time dimension in the columns and the Account dimension as a slicer. (In a supplemental schedule, the row headings are always Organization members.)

In this figure (which shows only part of a form template), each supplemental schedule is set to display a different slicer value (that is, a different account). With such a layout, each supplemental schedule could have different formulas that are appropriate for the selected account.

*Figure 9.1* *Example Supplemental Schedule*



| | | | | | |
|---|---|---|---|---|---|
| Analysis | ACTUAL | | | | |
| Organization | Internal Comm | | | | |
| | JAN2009 | FEB2009 | MAR2009 | APR2009 | MAY2009 |
| YearlyCompensation | 22,769.63 | 22,723.89 | 22,694.18 | 22,670.74 | 22,677.57 |
| Commission | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Salary | 22,459.52 | 22,505.10 | 22,537.70 | 22,560.64 | 22,580.80 |
| Bonus | 310.12 | 218.80 | 156.48 | 110.10 | 96.77 |
| Benefits | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Medical | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Life Insurance | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Head Count | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

| | | | | | |
|---|---|---|---|---|---|
| Analysis | ACTUAL | | | | |
| SupplementalAcctDim | Salary | | | | |
| | Hire Date | Current Annual Salary | Salary Inc. Effective Date | Evaluation Results | Raise Percent |
| Gearino, Dan S. | 5/28/2001 | 25,842.94 | 1/1/2009 | 4.00 | 3.00 |
| Kuo, Pauline D. | 10/24/1997 | 24,310.56 | 2/1/2009 | 3.00 | 2.25 |
| Godley, Lucille J. | 6/15/2008 | 26,079.60 | 3/1/2009 | 2.00 | 1.50 |
| Ross, John B. | 7/23/2004 | 36,699.35 | 4/1/2009 | 1.00 | 0.75 |
| Loflin, Angela B. | 10/7/1996 | 21,504.18 | 5/1/2009 | 1.50 | 1.13 |
| Holt, Dennis K. | 4/4/2005 | 30,570.67 | 6/1/2009 | 2.50 | 1.88 |
| Muzzy, Angela N. | 11/2/2007 | 27,055.98 | 7/1/2009 | 3.50 | 2.63 |
| Cooper, Nellie C. | 3/30/2002 | 35,536.70 | 8/1/2009 | 4.50 | 3.38 |
| Quinn, Rita M. | 3/6/1998 | 41,138.34 | 9/1/2009 | 5.00 | 3.75 |
| Detail averages for Internal Comm | | | | | |
| Detail totals for Internal Comm | | 268,738.32 | | | |

| | | | | | |
|---|---|---|---|---|---|
| Analysis | ACTUAL | | | | |
| SupplementalAcctDim | Bonus | | | | |
| | Current Annual Salary | Bonus Date | Evaluation Results | Bonus | JAN2009 |
| Gearino, Dan S. | 25,842.94 | 1/15/2009 | 4.00 | 310.12 | 310.12 |
| Kuo, Pauline D. | 24,310.56 | 2/15/2009 | 3.00 | 218.80 | 0.00 |
| Godley, Lucille J. | 26,079.60 | 3/15/2009 | 2.00 | 156.48 | 0.00 |

Alternatively, you might create a single supplemental schedule and allow the end user to select an account from the slicer and enter values for that account. As the user switches between accounts, the values are cached. When the user saves the supplemental schedule, the cached values are written to the plan table.

The following points are important to remember as you create supplemental schedules:

- When you add a planning measure to a supplemental schedule, the planning measure is locked for that form set. If the measure is later changed in the Administration application, the changes do not affect that form set template. You can still add other planning measures to the form set template, and you can delete measures. However, you cannot update measures after they have been added to a template.
- Each New Position dialog box, however, contains all measures that are currently defined as prompts and lookups. In addition, there is only one instance of the SAS code, that is run whenever the end user clicks **OK** to add one or more new positions.
- Slicers are not required.
- A supplemental schedule can have more than one slicer. However, if you associate a measure with one slicer dimension in a supplemental schedule, then you cannot associate the same measure with a different slicer dimension in another supplemental schedule in the same form set template.
- The following FM dimensions are not permitted as slicers: Currency, Trader, Source, RateType, ToCurrency, XRateType, and Frequency.
- You cannot use calls to the SAS Financial Management Add-In API for Microsoft Excel for a supplemental schedule. However, the same formatting options are available for supplemental schedules as for the FM table: **Format Members** for member formatting, and **Format Cells** for cell formatting. For details, see the online Help for the add-in.

When you select **Save**, a plan table is created in the HCM database. It contains an EMPLOYEE_ID column and measures from the supplemental schedule.

### *What Happens at Run Time*

At run time, users with the appropriate permissions enter data in the data-entry form and supplemental schedules. When they finish entering data, they select **Save All Supplemental Data** from the **SAS Solutions** menu. For each of the FM accounts in the supplemental schedules, the totals are rolled up to the data-entry table, and values from the supplemental schedules are saved in the plan table. Sums and averages are not stored. The standard HCM tables (such as the Employee Master table) provide input to the supplemental schedules, but they are not updated.

Here are some points to note about the end-user experience:

- Data entry and review must be performed in Microsoft Excel, not via Web data entry. By default, form sets are created to disallow Web data entry.
- Users cannot check out a data-entry form (for offline budgeting) that contains a supplemental schedule.
- If a numeric field has an error (for example, resulting from a failed calculation or a failed query from an HCM table), a zero value is displayed.
- If the data returned from the server has a null or empty value for a date field, Excel renders it as **1/0/1900**.

## Managing Plans

On the **Data** tab of the Administration application, you can view and delete existing plans. In the navigation tree, select **Planning** ⇨ **Plans**.

The display contains these columns:

| Column | Description |
|---|---|
| **Formset** | Name of the form set that is the basis of this plan. |
| **Table** | Name of the table that was created in the HCM database to contain supplemental schedule data. |

To delete a plan:

1. Click the action menu ▾ at the beginning of the row.
2. Select **Delete**.

# Part 2

# Data Administration

*Chapter 10*
**Introduction to Data Administration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *161*

*Chapter 11*
**Setting Up the SAS Data Integration Studio Environment** . . . . . . . . *169*

*Chapter 12*
**Using SAS Data Integration Studio to Supply Data to Solutions** . . *171*

*Chapter 13*
**Loading Language Codes and Data Locale Codes** . . . . . . . . . . . . . . . *181*

*Chapter 14*
**Loading Users and User Groups** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *185*

*Chapter 15*
**Creating a Dimension** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *187*

*Chapter 16*
**Loading Members and Hierarchies into a Dimension** . . . . . . . . . . . . . *193*

*Chapter 17*
**Registering Member Properties So That They Are Loaded into the SDM** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *205*

*Chapter 18*
**Exporting and Promoting Members and Hierarchies** . . . . . . . . . . . . . *209*

*Chapter 19*
**Adding a Dimension Type** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *215*

*Chapter 20*
**Loading Measures** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *253*

*Chapter 21*
**Loading Metrics** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *257*

*Chapter 22*
**Creating a Stored Process from a SAS Data Integration Studio Job** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *263*

*Chapter 23*
**Loading Data for SAS Human Capital Management** . . . . . . . . . . . . . . *265*

*Chapter 24*
**Modifying the Data Model for SAS Human Capital Management** . . *295*

*Chapter 25*
**Macro Variables in the PREBUILD.SAS Macro File** . . . . . . . . . . . . . . . *303*

*Chapter 26*
**Internal Formats** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *329*

*Chapter 10*

# Introduction to Data Administration

**The Mission of Data Administration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **161**

**New Data Administration Features in SAS Solutions Services** . . . . . . . . . . . . . . . **162**
Metadata Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163

**Survey of Tasks** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **164**
Preparatory Tasks for All Sites . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
Tasks to Consider for Any Site . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
Tasks to Support Scorecards . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
Tasks to Support SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . 166

**Server Configuration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **166**

**Documentation Conventions** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **166**

**Related Documentation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **167**

## The Mission of Data Administration

Your main mission as a SAS Solutions Services data administrator is to supply data to the SAS solution software. The relevant data spans a variety of content categories. These categories differ in the roles that they play and the times when they are needed:

- Data that belongs to certain content categories must be supplied initially in order to get the software working.
- Data that belongs to other content categories must be supplied periodically so that the software can produce timely output.
- Data that belongs to still other content categories might not be needed at all, depending on which solutions you are running and how you are using them.

Each content category involves unique considerations. However, there are some general themes:

- The ultimate destination of all the data that you supply is MySQL tables to which the solution software has access. These MySQL tables are grouped into three distinct areas—the Solutions Data Mart (SDM), the HCM Data Mart, and the SPM database.
- The main way of moving data from one table to another is by running SAS Data Integration Studio jobs.

- For many content categories, the data travels from its source to the MySQL tables through two sets of intermediate SAS tables—first the staging tables, and then the detail data store (DDS) tables. Where this is the case, the journey of the data has three main phases:
  1. Run a custom, site-specific job that extracts the data from its source and loads it into the staging table that is designed to hold it.
  2. Run the SAS Solutions Services job that moves the data from its staging table to its detail data store table.
  3. Run a SAS Solutions Services job or an equivalent wizard in the solution software to move the data from its detail data store table to its ultimate destination in a MySQL table.

## New Data Administration Features in SAS Solutions Services

The following data administration features are new as of the 5.x versions of SAS Solutions Services and the solutions.

### *Metadata Structure*

- The metadata folder structure and ETL shipped job content have a uniform look and feel for all solutions. ETL job names are lowercase and include sequence numbering to indicate the order of execution.
- By default, the solutions are installed in the Foundation Repository. For solutions that are migrated or installed in another repository, the LIBNAME statement searches across all repositories of metadata.
- The top-level folder for the jobs provided with the release is identified by the release number, such as **5.x Jobs**. When the product is migrated to the next release, a new folder is included with the release name and its jobs.
- Each SAS Library resides in an associated folder under its particular solution. Examples are the `CrossIndustryDDS`, `Error Data`, and `StageDDS` folders. Each folder contains the table metadata and the library object. The folder name and the library object name are the same. The libref is a SAS associated name. The DDS libref has been changed to CIND_DDS.
- Migrated SAS Data Integration Studio DDS jobs are located in the `/Products/Cross Industry Detail Data Store/Migrated Jobs` folder. All other DDS *X*.4 jobs under the **Inventory** tab have not been upgraded to SAS Solutions Services 5.3 and do not produce correct results.

### *Security*

Authentication has been added to the Solutions ETL jobs based on the user ID and password used to run the job (interactive or batch). The transformations have been updated to remove the previous options for user ID and password.

## *SAS Human Capital Management*

The following changes are of interest primarily to users of SAS Human Capital Management:

- These are the default metadata folder paths for SAS Human Capital Management:

  Tables
  **`/Products/SAS Human Capital Management/Data Sources/HCMData`**

  Information Maps
  **`/Products/SAS Human Capital Management/Data Sources/Information Maps`**

  OLAP Cubes
  **`/Products/SAS Human Capital Management/Data Sources/Cubes`**

  Jobs
  **`/Products/SAS Human Capital Management/5.2 Jobs`**

  Reports
  **`/Products/SAS Human Capital Management/5.2 Reports`**

- The names of HCM jobs have been changed to include sequence numbers.
- Three new forecasting jobs have been added to build three new forecasting tables:
  - hcm_126050_run_month_forecast (HCMMONTHFORECAST)
  - hcm_126100_run_quarter_forecast (HCMQTRFORECAST)
  - hcm_126150_run_year_forecast (HCMYEARFORECAST)

  • By default, the hcm_126050_run_month_forecast job forecasts New Hires, Churn, and Change in Headcount. The hcm_126100_run_quarter_forecast job forecasts Voluntary Terminations and Headcount. The hcm_126150_run_year_forecast job forecasts Involuntary Terminations.
- The Metrics process has been modified to store all metrics in the new table SAS_MEASURES. There are two new jobs to load these metrics:
  - hcm_128050_load_sas_measures_table
  - hcm_128100_load_sas_measures_table_with_org

  If a site has SAS Strategy Management, the site can load the metrics to the SDM using the following jobs:
  - hcm_128900_load_sdm_metric_table
  - hcm_128901_load_sdm_metric_table_with_org
- The HRVANLY2, MODELSCORES and EMPSCORES retention analysis data tables have been added. HRVANLY2 is a utility table that is not registered in SAS Human Capital Management. There are three new jobs to build these tables:
  - hcm_140050_load_hrvanly2_table
  - hcm_140100_load_modelscores_table
  - hcm_140150_load_empscores_table
- Default formats are now loaded at installation directly into the new SAS_HCM_FORMATS table. The DDS formats and Organizational Hierarchy

formats are loaded into SAS_HCM_FORMATS during the build. The formats in SAS_HCM_FORMATS are maintained via a Web user interface. The FORMATS catalog is updated from the formats in the SAS_HCM_FORMATS table.

- The data locales available for SAS Human Capital Management are restricted to the locales that are loaded into the detail data store CODE_LANGUAGE table. The job that loads the formats table, hcm_110050_load_formats_table, is modified so that it flags the detail data store CODE_LANGUAGE locales in the SAS_LOCALE_LIST table.
- For stored process reports, the column labels and report titles can now be displayed using the user's SAS Portal language preference if the corresponding locale properties files are available. If the properties files are not available, the locale set for SAS Human Capital Management will be used.
- A number of macros have been deleted or modified.
- A new job has been added to load the SAS_USER_EMPLOYEE table: hcm_110450_load_sas_user_employee_table. All active employee users that are members of the metadata group designated for SAS Human Capital Management users are loaded into the table.
- Two new Education Detail tables have been added, along with the jobs to build them:

  EDUHIST
  : This education history detail table contains the employee's education history.

  EDUVAL
  : This education assessment detail table contains the employee's educational metrics.
- A new job to refresh the SAS Human Capital Management cache, hcm_900000_refresh_cache, runs at the end of the build.
- The Education Enrollment jobs have been removed.

## Survey of Tasks

### Preparatory Tasks for All Sites

The set of relevant data administration tasks depends on your site-specific circumstances. At any site, perform the following tasks in the specified order before you begin to load data:

1. Complete the installation of SAS Solutions Services.
2. Prepare the SAS Data Integration Studio environment, following the instructions in "Setting Up the SAS Data Integration Studio Environment" on page 169.

### Tasks to Consider for Any Site

Tasks that are not specific to a single solution are grouped together in Part 1. Here is a summary of the most important cross-solution tasks:

- To support any solution, you must load user and user group data into the Solutions Data Mart (SDM). You must also take steps to ensure that the user and group data in the SDM always matches the user and group data in the metadata repository. For details, see "Loading Users and User Groups" on page 185.

- To support any solution, you must load measures, as described in "Loading Measures" on page 253.
- If the set of predefined dimension types is not adequate for your purposes, then you must define additional dimension types, as explained in "Adding a Dimension Type" on page 215.
- You must define dimensions, members, and hierarchies for the dimension types that you are using. To do this with SAS Data Integration Studio, see "Creating a Dimension" on page 187 and "Loading Members and Hierarchies into a Dimension" on page 193.
- At your discretion, you can widen the availability of any SAS Data Integration Studio job by converting it into a stored process. See "Creating a Stored Process from a SAS Data Integration Studio Job" on page 263.
- In order for a job to send a **Data Modified** notification, you must edit the transformation and specify the condition that generates the notification and the action to be taken. See "Send Notifications from Jobs" on page 170.

### *Tasks to Support Scorecards*

To support the use of scorecards in SAS Strategy Management or the KPI viewer, consider all of the following:

- If the set of predefined measures is not adequate for your purposes, then you must define and load additional measures. To review the set of predefined measures, view the data in the SAS_MEASURE table. For details, see "Loading Measures" on page 253.
- For each dimension type that is used in scorecards, you must make sure that it is properly stocked with dimensions, members, and hierarchies. For details, see "Creating a Dimension" on page 187 and "Loading Members and Hierarchies into a Dimension" on page 193.

  Every scorecard must use the TIME dimension type to indicate the time periods that numeric values apply to. The ANALYSIS dimension type can be used to distinguish results from forecasts and budgets. All other dimension types are also available for use in scorecards.
- The numeric values that are displayed in scorecards can be supplied in any of the following ways:
  - You can use SAS Data Integration Studio to load numeric values from an external source to a metric table, which scorecards can point to. For details, see "Loading Metrics" on page 257.
  - If you have SAS Human Capital Management installed, you can use SAS Data Integration Studio to load numeric values from the HCM Data Mart to a metric table, which scorecards can point to. For details, see "Loading HCM Metrics into a Metric Table" on page 291.
- You might want to load user-member associations that determine default read access to scorecards. See "Users Tab Data" on page 200.

### Tasks to Support SAS Human Capital Management

See "Loading Data for SAS Human Capital Management" on page 266.

## Server Configuration

There are three types of servers that you might work with:

- The metadata server is the server machine on which the SAS Metadata Server software is running. SAS must be available on this same machine.
- The data-tier server is the server machine on which SAS runs data-handling programs (including the logical servers for workspace and stored process servers). Transformations, error tables, and jobs are installed on the data-tier server.

  The same machine is often used as both the data-tier server and the metadata server.
- The middle-tier server is the server machine on which the managed servers and SAS Remote Services run. Certain activities require you to start or stop the managed servers and SAS Remote Services, as explained in the System Administration part of this book.

## Documentation Conventions

This book uses the following documentation conventions to identify paths in the solutions configuration:

***Table 10.1*** *Conventions for Pathnames*

| Path | Refers to | Example |
|---|---|---|
| **!SASROOT** | Path to the SAS root directory | Windows: **C:\Program Files\SAS \SASFoundation\9.2**<br>UNIX: **/usr/local/SAS/SASFoundation/9.2** |
| *SAS-config-dir* | Path to the SAS configuration directory | Windows: **C:\SAS\Config**<br>UNIX: **/usr/local/SAS/Config** |
| *MySQL-install-dir* | Path to the MySQL installation directory | Windows: **C:\mysql**<br>UNIX: **/usr/local/mysql** |

*Note:*

- The name of the configuration directory and the level number might be different at your site.
- If your configuration is the result of a migration from the previous release of SAS Solutions Services, the **SASApp** directory might be called **SASMain** instead (for example, **C:\SAS\Config\Lev1\SASMain** rather than **C:\SAS\Config**

**\Lev1\SASApp**). Please make the appropriate substitutions as you read this book.

- File system pathnames are typically shown with Windows separators (“\”); for UNIX, substitute a forward slash .

The data-tier server can be either a Windows server or a UNIX server. For a Windows server, this book assumes that the installation drive is the C drive.

This book uses the following abbreviations:

- DDS—Detail Data Store
- HCM—SAS Human Capital Management
- SDM—Solutions Data Mart

## Related Documentation

See the following site for related documentation:

- **SAS Human Capital Management:** `http://support.sas.com/documentation/onlinedoc/hcm`

This site is password-restricted. You can find the user name and password in the preinstallation checklist or by calling Technical Support.

*Chapter 11*

# Setting Up the SAS Data Integration Studio Environment

**Overview of Setup Tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169**

**Access Settings for the Data-Tier Server . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169**

Protecting Data Directories . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169

Giving Server Access to Users of SAS Data Integration Studio . . . . . . . . . . . . . . 169

Groups and Roles for Data Administrators . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170

**Copy Jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170**

**Send Notifications from Jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170**

## Overview of Setup Tasks

This chapter describes setup tasks that you must perform after you install SAS Solutions Services and before you start using SAS Data Integration Studio to load data. These setup tasks consist of establishing access settings for the data-tier server.

## Access Settings for the Data-Tier Server

### *Protecting Data Directories*

For information about operating-system protection for files and folders, see “Post-Configuration Steps” in the System Administration part of this book.

### *Giving Server Access to Users of SAS Data Integration Studio*

#### *Overview*

Each user of SAS Data Integration Studio must have a user ID and password for the data-tier server.

This user must not be the unrestricted user. If you log on as the unrestricted user, then you cannot attach the libraries that are necessary to run SAS Data Integration Studio.

The user must have the following rights and permissions:

- the `Log on as a batch job` right.

The recommended way to grant this right to a user is to place the user in the SAS Server Users group and grant the right to this group. For more information, see “Windows Privileges” in the *SAS Intelligence Platform: Security Administration Guide*.

- Read/Write/Update access to the ***SAS-config-dir*\Lev1\Data** directory and all its subdirectories.

### Groups and Roles for Data Administrators

For information about group and role requirements for data administrators, see “Assigning Groups and Roles” in the System Administration part of this book.

## Copy Jobs

There are some jobs that you might need many copies of so that you can customize the copies in different ways. There are two ways to obtain another copy of a job:

- Use **Copy** and **Paste**: select the job, select **Copy** from the pop-up menu, select the folder in which you want to place the copy, select **Paste** from the pop-up menu.
- For a very simple job that consists only of a single transformation, you can create a blank job and then drag and drop the required transformation onto the job’s process diagram.

## Send Notifications from Jobs

When a job is run, no notifications are sent automatically. In order for a job to send a **Data Modified** notification, you must edit the transformation and specify the condition that generates the notification and the action to be taken. For example, a site might want to send notifications for the Table Loader and SCD Type 2 Loader transformations. An action might be an e-mail message or a custom action that is defined at a site.

For more information, see “Managing the Status of Jobs and Transformations” in the *SAS Data Integration Studio: User's Guide* (available at **support.sas.com/documentation/**).

# Chapter 12
# Using SAS Data Integration Studio to Supply Data to Solutions

**Overview of the Main Data Pathway** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **171**

**Data Encodings** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **172**

**Moving Data from Its Source to the Staging Tables** . . . . . . . . . . . . . . . . . . . . . . . . . **173**

**Loading the Detail Data Store Tables from the Staging Tables** . . . . . . . . . . . . . . . **174**

Survey of the Detail Data Store Jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174

Setting a Valid Time Range for Data Records . . . . . . . . . . . . . . . . . . . . . . . . . . 175

Modifying the Jobs That Load the Detail Data Store Tables . . . . . . . . . . . . . . . . . 176

Testing a Detail Data Store Job . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177

Scheduling a Detail Data Store Job . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177

**Moving Data from the Detail Data Store to the SDM** . . . . . . . . . . . . . . . . . . . . . . . **177**

Overview of Moving Data from the Detail Data Store to the SDM . . . . . . . . . . . . 177

Testing a Job That Loads Data into the SDM . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177

Scheduling an SDM Job . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177

**Moving Data from the Detail Data Store to the HCM Data Mart** . . . . . . . . . . . . . **178**

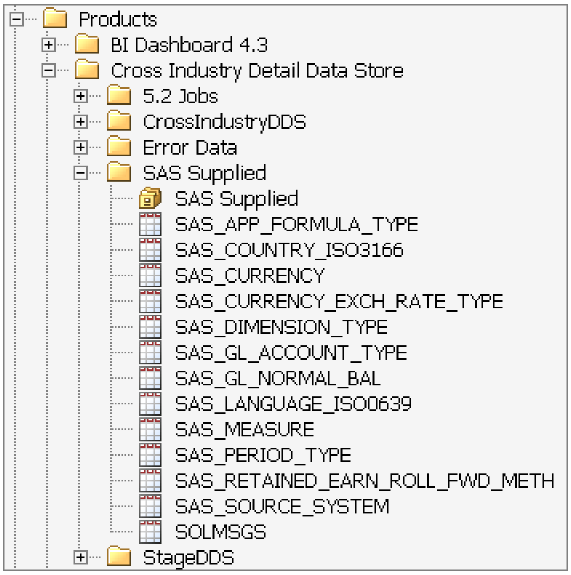
**Overview of Other Data Pathways** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **178**

**Extending the Detail Data Store** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **178**

## Overview of the Main Data Pathway

Most data moves from its source, through the detail data store staging area and the detail data store (DDS), to a destination data mart.

In general, the sources of data are transactional systems or databases that are outside the SAS environment. However, there are some source tables of predefined data that are installed with the SAS Solutions Services software. The predefined source tables are the SAS_ tables in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **SAS Supplied** folder on the **Folders** tab of SAS Data Integration Studio.



- Products
  - BI Dashboard 4.3
  - Cross Industry Detail Data Store
    - 5.2 Jobs
    - CrossIndustryDDS
    - Error Data
    - SAS Supplied
      - SAS Supplied
      - SAS_APP_FORMULA_TYPE
      - SAS_COUNTRY_ISO3166
      - SAS_CURRENCY
      - SAS_CURRENCY_EXCH_RATE_TYPE
      - SAS_DIMENSION_TYPE
      - SAS_GL_ACCOUNT_TYPE
      - SAS_GL_NORMAL_BAL
      - SAS_LANGUAGE_ISO0639
      - SAS_MEASURE
      - SAS_PERIOD_TYPE
      - SAS_RETAINED_EARN_ROLL_FWD_METH
      - SAS_SOURCE_SYSTEM
      - SOLMSGS
    - StageDDS

There are two destination data marts:

- Solutions Data Mart (SDM)
- HCM Data Mart

Each destination data mart is a MySQL database. The predefined source tables, the staging tables, and the detail data store tables are all SAS tables.

The complete path that the data follows consists of the following main steps:

1. Using jobs that you write, extract data from your source systems and load it into the appropriate staging tables.

   This step does not apply to the predefined source tables.

2. For each staging table, run the job that loads data from it into the corresponding detail data store table. The jobs that move data from the staging tables to the detail data store tables do some standard processing, including providing retained keys and time stamps for all the records.

   For each predefined source table, a single job moves the data from the predefined source table to the staging table and then from the staging table to the detail data store table.

3. Load the data from the detail data store into the appropriate data mart.

   You can always perform this step by running the appropriate job in SAS Data Integration Studio.

## Data Encodings

The MySQL databases that hold solution data must be set up at installation time to use the UTF-8 encoding.

Unless you are using double-byte SAS with a UTF-8 SAS session encoding, jobs that load data from the detail data store to the SDM must transcode the data from the SAS session encoding to UTF-8. Conversely, jobs that export data from the SDM to staging tables must transcode the data from UTF-8 to the SAS session encoding. To facilitate this transcoding, your SAS session encoding must be specified at installation time.

## Moving Data from Its Source to the Staging Tables

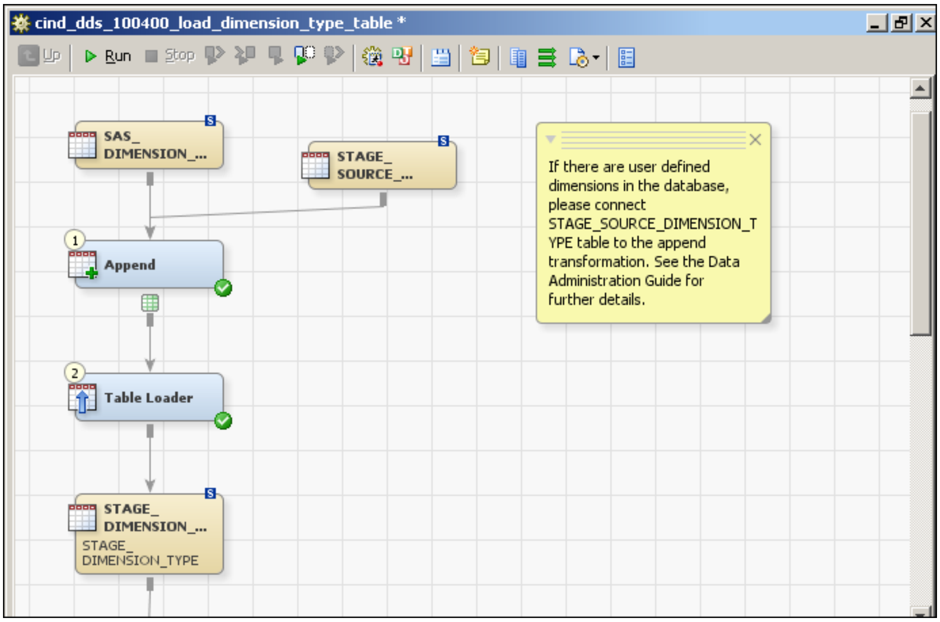
The trip from source to staging table has one form when the source is outside the SAS environment and another form when the source is a SAS_ table of predefined data:

- If the data source is outside the SAS environment, then you can load the appropriate staging table from the data source in any way that you want. For example, you can write a separate job to load each staging table or you can write jobs that load groups of related staging tables. You can run the jobs in any order. You can store the jobs in any folder. The one requirement is that your jobs must place the right data in the right columns of the right staging tables. If they achieve that result, then the jobs that load the detail data store tables from the staging tables can perform the next step.

  If one of your data sources is SAP, then you can use the SAS Solutions Adapter for SAP. See *SAS Solutions Adapter for SAP: User's Guide*.

  If you are running SAS under 64-bit Windows and the source files are on a machine running 32-bit Windows, then you must use SAS PC Files Server to configure the data sources. For instructions, see “Post-Configuration Steps” in the System Administration part of this book.

- If the data source is a SAS_ table of predefined data, then the job that loads the corresponding detail data store table does all the work. The job first moves the data from the SAS_ source table to the corresponding staging table, and then moves the data from the staging table to the corresponding detail data store table. Most of the jobs that load a detail data store table begin at the staging table, but the jobs that handle predefined data begin at the SAS_ source table, as illustrated here by the job that loads dimension types:



If there are user defined dimensions in the database, please connect STAGE_SOURCE_DIMENSION_TYPE table to the append transformation. See the Data Administration Guide for further details.

As this process diagram shows, these jobs enable you to supplement the predefined data with additional data from another source. You never need to write an additional job.

Before you can write a job to extract data from an external source and load it into a staging table, you must understand all the data columns in the relevant staging table. For each column, you must either determine the data source or else verify that it is appropriate to leave the column empty.

This manual discusses the structure of some of the staging tables. The Data Model part of this book shows the structure of every detail data store table. Each detail data store table is identical to the corresponding staging table except for the presence of additional columns for such things as time stamps and automatically generated key values.

In writing a job to extract data from an external source and load it into a staging table, you might be able to use the following:

- The User Written Code transformation.
- Registering tables. Right-click a metadata folder and select **Register Tables** to register your data sources in the metadata repository, so that they show up as icons in SAS Data Integration Studio. You can then use the icons in the Process Designer.

## Loading the Detail Data Store Tables from the Staging Tables

### *Survey of the Detail Data Store Jobs*

On the **Folders** tab, the jobs that load the detail data store tables are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder:

On the **Inventory** tab, all the jobs are in the **Jobs** folder.

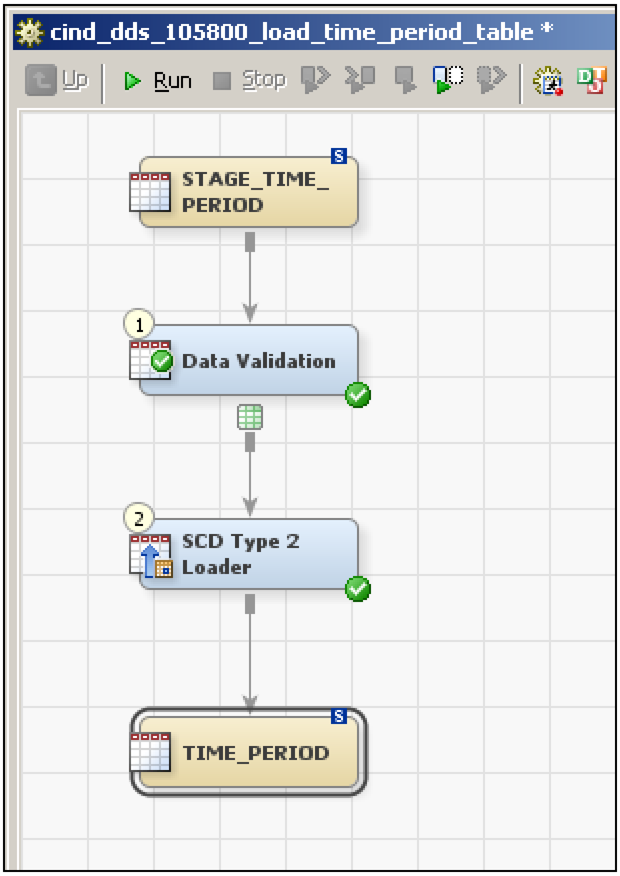
The names of most jobs contain a six-digit sequence number. The purpose of these sequence numbers is to indicate the order in which you should run the jobs. You can load the staging tables in any order, but you must load the detail data store tables in an order that enables each job to validate the values that it loads against other already-loaded tables. Certain variations on the order that is indicated by the sequence numbers can also work, but the safe course is to follow the sequence numbers.

Jobs that define additional dimension types do not have sequence numbers,as explained in “Adding a Dimension Type” on page 215. You need a separate copy of these jobs for each additional dimension type that you define. When you rename these copies, you can include appropriate sequence numbers along with the name of the relevant dimension type.

More basic than the question of job order is the question of which jobs you need to run. You should first determine which jobs you need to run, and then run those jobs in the order that is indicated by their sequence numbers. For example, if you are supplying data to one solution but not to another, then you should ignore the jobs that are specific to the other solution. If you are doing a specific maintenance task such as updating the members and hierarchies of a certain dimension, then you should run only the jobs that contribute to that task.

### *Setting a Valid Time Range for Data Records*

Many staging tables include a Valid From Datetime column and a Valid To Datetime column. If a staging table includes these columns, then the accompanying detail data store job includes the SCD Type 2 Loader transformation, as shown in the following example:



The Valid From Datetime and Valid To Datetime columns are the basis of the so-called Slowly Changing Dimension (SCD) capability. Each record is recognized as valid by the solution software only during the time range that begins with its Valid From Datetime value and ends with its Valid To Datetime value. This enables the software to maintain bundles of two or more records that have different times of validity but otherwise identical key values. The records in such a bundle represent different time-dependent versions of the same entity.

There is a very simple way to manage these two columns that is likely to meet your needs: do nothing. If you load nothing into these columns in the staging tables, then the job that loads the corresponding detail data store table automatically generates the following values for the corresponding columns in the detail data store table:

- Valid From Datetime—the second at which the job begins to load records into the detail data store
- Valid To Datetime—January 1, 5999:00:00:00

When a record is loaded from the detail data store into a data mart, it might encounter a pre-existing record that has the following set of characteristics:

- the same key value, apart from its time of validity, as the record being loaded

- a Valid From Datetime that is earlier than the Valid From Datetime of the record being loaded
- a Valid To Datetime of January 1, 5999:00:00:00

In this case, the Valid To Datetime of the pre-existing record is changed to be one second earlier than the Valid From Datetime of the new record.

If these automatically generated values do not meet your needs, then you can load any values that you want into the Valid From Datetime and Valid To Datetime columns of a staging table. You can use either of the two ways in which SAS data sets can represent time values:

- A count of seconds, starting with January 1, 1960:00:00:00 as the first second.

  If you load numbers that count seconds, then make sure that the Format Type for Dates option of the job's SCD Type 2 Loader transformation has the default value, **Source begin and end column values are to flow to the target without change**:

  This causes the counts of seconds in the staging table to be copied without change to the corresponding detail data store table.

- A count of days, starting with January 1, 1960 as day 1.

  If you load numbers that count days, then make sure that the Format Type for Date option of the job's SCD Type 2 Loader transformation has the value DATE. This causes the counts of days in the staging table to be converted to counts of seconds in the corresponding detail data store table.

If you do not load values into the Valid From Datetime and Valid To Datetime columns of the staging table, then the value of the Format Type for Date option has no effect. The job automatically generates counts of seconds no matter what the value of the Format Type for Date option is.

### *Modifying the Jobs That Load the Detail Data Store Tables*

In most cases, the jobs that load the detail data store tables do not need to be modified. If you load the correct input into the staging table that feeds a job and then run the job, the job loads the correct output into the corresponding detail data store table. However, there are a couple of modifications that you might want to make in certain cases.

The first processing step in most detail data store jobs is a Data Validation transformation, which validates certain columns of data. To see which columns this transformation validates and how it validates them:

1. In the process diagram, select the Data Validation transformation.
2. Right-click and select **Properties** from the pop-up menu. The Data Validation Properties window appears.
3. In the Data Validation Properties window, select the **Invalid Values** tab. This tab lists the validated columns and shows which column of which other table each validated column is validated against.

On the **Invalid Values** tab, pay special attention to the **Blanks are valid** column. The value in this column must be YES for any row that represents a column that you are not loading with data.

In some jobs the last processing step is an SCD Type 2 Loader transformation. In other jobs the last processing step is a Table Loader transformation. For any Table Loader transformation, you can use the **Load Style** field on the **Load Technique** tab to select one of the following load styles:

- Append to Existing—Leave all existing records in place and load all new records.
- Replace—Delete all existing records, and then load all new records.
- Update/Insert—Overwrite records that have matching keys, leave in place records that are not overwritten, and add records with new keys.

The Replace load style is the default. In certain cases, you might want to make a different selection.

### *Testing a Detail Data Store Job*

After you modify a detail data store job, test it by running the job and checking the log to make sure that the job ran cleanly and produced the desired results.

### *Scheduling a Detail Data Store Job*

For some categories of data, you might want to run the relevant detail data store job regularly according to a defined schedule.

After you complete and successfully test a detail data store job, you can deploy it for scheduling. This generates the code for the job and stores the resulting code files on a server, ready for scheduling. Use the scheduling tool of your choice to schedule the job to run on a regular basis.

## Moving Data from the Detail Data Store to the SDM

### *Overview of Moving Data from the Detail Data Store to the SDM*

The jobs that load data into the SDM from tables in the detail data store require minor modifications. They use transformations that are in the **Solutions Transforms** folder on the **Transformations** tab.

### *Testing a Job That Loads Data into the SDM*

After you modify an SDM job, test it by running the job and checking the log to make sure that the job ran cleanly and produced the desired results.

### *Scheduling an SDM Job*

For some categories of data, you might want to run the relevant SDM job regularly according to a defined schedule.

After you complete and successfully test an SDM job, you can deploy it for scheduling. This generates the code for the job and stores the resulting code files on a server, ready for scheduling. Use the scheduling tool of your choice to schedule the job to run on a regular basis.

## Moving Data from the Detail Data Store to the HCM Data Mart

See "Moving Data from the Detail Data Store to the HCM Data Mart" on page 279 for a detailed discussion.

## Overview of Other Data Pathways

The following categories of data travel over pathways that do not involve the detail data store:

- User and user group data travels instead through the metadata repository. It is loaded first into the metadata repository, and then into the SDM from the metadata repository. The jobs that load this data into the SDM are in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder on the **Folders** tab of SAS Data Integration Studio. There are three of these jobs:
  - solnsvc_1300_load_users loads the user definitions.
  - solnsvc_1400_load_groups loads the group definitions.
  - solnsvc_1500_load_user_x_group loads the information about which users belong to which groups.

  For details, see "Loading Users and User Groups" on page 185.
- Metrics are loaded directly into the SDM metric tables from source tables that you build. For details, see "Loading Metrics" on page 257.
- Dimensions can be created in the detail data store or directly in the SDM. For details, see "Creating a Dimension" on page 187.
- Certain tables in the HCM Data Mart must be loaded directly, using jobs that you write. For details, see "Loading Certain HCM Data Mart Tables Directly" on page 285.

The following categories of data travel through the detail data store, but give you a choice of two ways to load them from the detail data store into the SDM:

- Members and hierarchies for an existing dimension. See "Loading Members and Hierarchies into a Dimension" on page 193.

## Extending the Detail Data Store

You can extend the detail data store in two general ways:

- Add more detail data store tables to the detail data store tables that are installed as part of SAS Solutions Services.
- Add columns to installed tables.

In general, if you extend the detail data store, the additional data cannot be loaded into a predefined data mart; in order to make use of the additional data, you must load it into tables in a separate location that is accessible by an appropriate application. There are three important exceptions:

- You can extend the HCM Data Mart by adding new tables or adding columns to existing tables. The HCM Data Mart is used by SAS Human Capital Management. For details, see "Modifying the Data Model for SAS Human Capital Management" on page 295.
- You can add a column that represents a custom property to the primary member table of any dimension type. You can load the values of a custom member property into the SDM by following the procedure that is described in "Registering Member Properties So That They Are Loaded into the SDM" on page 205.

If you add detail data store tables that are to be used to load non-data-mart tables, be sure to do all of the following:

1. Create the detail data store tables.
2. Create corresponding staging tables.
3. Create the non-data-mart target tables if they do not already exist.

   *Note:* Do not write an application that accesses detail data store tables.
4. Right-click a metadata folder and select **Register Tables** to register the metadata of all the new tables, including the staging tables, the detail data store tables, and the target tables.
5. Create jobs that load the staging tables.
6. Create jobs that load the detail data store tables from the staging tables.
7. Create jobs that load the non-data-mart target tables from the detail data store tables.

The result of the preceding set of steps is a data pathway to the non-data-mart target tables that is analogous to the main data pathway to the data marts. The main data pathway is described in "Overview of the Main Data Pathway" on page 171 .

If you add columns to existing detail data store tables that are to be used to load non-data-mart target tables, be sure to do all of the following:

1. Add the columns to the detail data store tables.
2. Add corresponding columns to the corresponding staging tables.
3. Use **Tools** ⇨ **Update Table Metadata** to register the metadata of all the modified tables, including the staging tables and the detail data store tables.
4. Modify the jobs that load the staging tables.
5. Modify the jobs that load the detail data store tables from the staging tables.
6. Create the non-data-mart target tables, if they do not already exist.

   *Note:* Do not write an application that accesses detail data store tables.
7. Right-click a metadata folder and select **Register Tables** to register the metadata of the new non-data-mart target tables.
8. Create jobs that load the non-data-mart target tables from the detail data store tables.

If you add a column to a member table for the purpose of loading an additional member property into the SDM, do the first five of the preceding steps in order to provide for the trip into the detail data store, and then provide for the trip from the detail data store to the SDM as described in "Registering Member Properties So That They Are Loaded into the SDM" on page 205.

*Chapter 13*

# Loading Language Codes and Data Locale Codes

**Overview of Languages and Data Locales** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 181

**Loading the Staging Table for Language and Locale Data** . . . . . . . . . . . . . . . . . . 181

**Data Locales with Predefined Text** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182

**Loading the Detail Data Store Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183

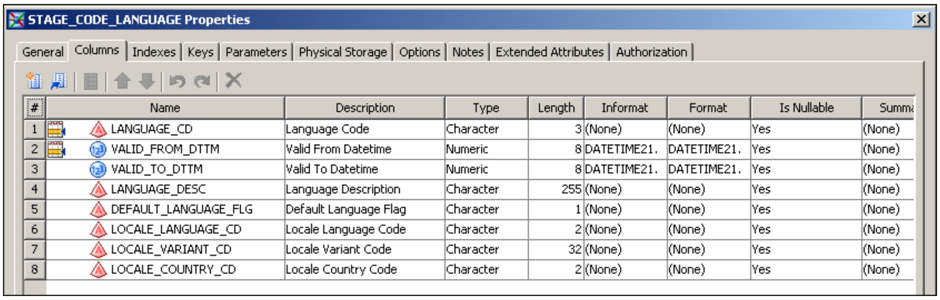
**Loading Data Locale Codes into the SDM** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 183

## Overview of Languages and Data Locales

Language codes and data locale codes are used to identify the language in which associated textual data is expressed. The language codes are used in the staging tables and the detail data store tables. The data locale codes are used in the SDM. You can view them by selecting **Tools** ⇨ **Data Locales** in the SAS Solutions Dimension Editor.

## Loading the Staging Table for Language and Locale Data

You must write and run a job that loads all the language codes and data locale codes that your site requires into the STAGE_CODE_LANGUAGE table:



| # | Name | Description | Type | Length | Informat | Format | Is Nullable | Summa |
|---|---|---|---|---|---|---|---|---|
| 1 | LANGUAGE_CD | Language Code | Character | 3 | (None) | (None) | Yes | (None) |
| 2 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DATETIME21. | DATETIME21. | Yes | (None) |
| 3 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DATETIME21. | DATETIME21. | Yes | (None) |
| 4 | LANGUAGE_DESC | Language Description | Character | 255 | (None) | (None) | Yes | (None) |
| 5 | DEFAULT_LANGUAGE_FLG | Default Language Flag | Character | 1 | (None) | (None) | Yes | (None) |
| 6 | LOCALE_LANGUAGE_CD | Locale Language Code | Character | 2 | (None) | (None) | Yes | (None) |
| 7 | LOCALE_VARIANT_CD | Locale Variant Code | Character | 32 | (None) | (None) | Yes | (None) |
| 8 | LOCALE_COUNTRY_CD | Locale Country Code | Character | 2 | (None) | (None) | Yes | (None) |

Each record in this table defines a language code and a three-part data locale code and associates the language code with the data locale code. In populating this table, note the following:

- Load only those languages and data locales that are used by your data. If all your data is in a single data locale, then you need to load only one record into this table.

  SAS Human Capital Management always uses only one data locale.

- Language Code is the language code that is used in staging tables and detail data store tables.

  In general, this language code should be one of the two-character codes in the ISO0639_LANGUAGE_CD column of the SAS_LANGUAGE_ISO0639 table. You need to make an exception only if you need two or more records that represent variants of the same language. For example, if you have a record for French as used in France and another record for French as used in Canada, then you might use language codes **`frf`** and **`frc`**, respectively.

  Do not use the same language code in two records.

- Language Description is a description of the language or language variant that Language Code designates.

  For example, you might specify **`French`** or **`Canadian French`**.

- Default Language Flag must be Y for exactly one record and N for all other records. Y marks the language code for the language that is used in all the primary member tables. Be careful to coordinate the language that you mark here as the default language with the language that you use in the primary member tables. For a detailed discussion of primary and secondary member tables, see “Loading Members and Hierarchies from the Staging Tables to the Detail Data Store” on page 201.

- Locale Language Code and Locale Country Code work together to identify the SDM data locale that is associated with the detail data store language code.

  - Locale Language Code must be one of the two-character codes in the ISO0639_LANGUAGE_CD column of the SAS_LANGUAGE_ISO0639 table.

  - Locale Country Code must be one of the two-character codes in the ISO3166_COUNTRY_CD column of the SAS_COUNTRY_ISO3166 table.

  In many cases, Locale Language Code can be the same two-character code as Language Code, and the other locale columns can remain empty.

  In general, the SDM data locale in the record that has a Default Language Flag of Y should be the data locale that is set in the SDM by the SAS Solutions Services installation.

  Do not use the same combination of locale language code and locale country code in two records.

- Valid From Datetime and Valid To Datetime define the lifespan of the record. See “Setting a Valid Time Range for Data Records” on page 175.

## Data Locales with Predefined Text

The installed software includes the names and descriptions of the predefined dimension types and predefined dimensions in all the following data locales:

- da (Danish)
- de (German)
- en (English)

- es (Spanish)
- fr (French)
- it (Italian)
- ja (Japanese)
- ko (Korean)
- pl (Polish)
- ru (Russian)
- zh_CN (simplified Chinese)
- zh_TW (traditional Chinese)

If you load any of these data locales into STAGE_CODE_LANGUAGE and carry them through into the SDM, then the associated predefined text will be available in the SAS Solutions Dimension Editor.

## Loading the Detail Data Store Table

To load the CODE_LANGUAGE table from the STAGE_CODE_LANGUAGE table, run the cind_dds_100200_load_code_language_table job. On the **Folders** tab, this job is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder.

## Loading Data Locale Codes into the SDM

To load data locale codes from the CODE_LANGUAGE table into the SDM, run the solnsvc_1200_import_locales job. On the **Folders** tab, this job is in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder. On the **Inventory** tab, it is in the **Jobs** folder.

Run the job and then review the log.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

*Chapter 14*

# Loading Users and User Groups

**Overview of Users and Groups** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185

**Ways to Load User and Group Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185

## Overview of Users and Groups

Definitions of users and groups are maintained in the metadata repository.

The SAS Solutions Services installation procedure defines a number of default users, groups, and roles (see the System Administration part of this book). You must define all the users at your site as well as their group and role memberships. You can provide this information through a bulk-load process or interactively though SAS Management Console.

Whenever a user logs in to the solution software, the authentication process consults the user data in the metadata repository. However, there are other uses of the user data that require it to be present in the SDM. Whenever changes are made to the user data in the metadata repository, the user data in the SDM must be updated to reflect those same changes.

## Ways to Load User and Group Data

The following jobs load data from the metadata repository to the SDM:

- solnsvc_1300_load_users
- solnsvc_1400_load_groups
- solnsvc_1500_load_user_x_group

These jobs are on the **Folders** tab in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder of SAS Data Integration Studio.

Best practice is to run these three jobs according to a regular schedule. For example, you might schedule a batch job to run each night. The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

The user account in which these jobs run must have Read and Write permissions to the ***SAS-config-dir*\Lev1** directory on the metadata server.

There is a stored process that includes all three of these jobs. For information about running the Import Users and Groups stored process, see “Assigning Groups and Roles” in the System Administration part of this book.

*Chapter 15*

# Creating a Dimension

**Dimension Types, Dimensions, Hierarchies, and Members** . . . . . . . . . . . . . . . . . . 187
**Ways to Create a Dimension** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
**Using the Create Dimension Transformation** . . . . . . . . . . . . . . . . . . . . . . . . . . . 188
**Starting from a Staging Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 189

## Dimension Types, Dimensions, Hierarchies, and Members

Before you perform any task that involves dimension types, dimensions, hierarchies, or members, make sure that you understand how these four concepts are related.

A dimension type represents a category of information. Examples are CURRENCY, and TIME—two of the predefined dimension types.

Each dimension type can contain many dimensions. Each dimension contains members and at least one hierarchy that is built from some or all of its members. The dimensions within a dimension type are like folders that enable you to separate the hierarchies and members into different groups.

Dimension types can and typically do have multi-level hierarchies. Here is an example:

- The members of a TIME dimension are time periods of different lengths. In a typical time hierarchy, years, quarters, and months are on different levels.

The CURRENCY dimension type can have only flat, single-level hierarchies.

This chapter is about creating a new, empty dimension. You create a dimension within a dimension type, which must already exist. If you need to create a dimension type, see "Adding a Dimension Type" on page 215. After you create a dimension, you must place members and hierarchies in it. "Loading Members and Hierarchies into a Dimension" on page 193 explains how to do that.

## Ways to Create a Dimension

A dimension is defined by a single dimension code, but it can have names and descriptions in any number of data locales. There are three ways to create a dimension:

- In the SAS Solutions Dimension Editor, select **New Dimension** and use the New Dimension wizard. Do this for each dimension that you need to create.

  If you are using several data locales, use the **Identification** tab of the dimension properties window in the SAS Solutions Dimension Editor to add names and descriptions in data locales other than the current data locale.

- In SAS Data Integration Studio, run the solnsvc_2200_create_dimension job or a job that you create that uses the create_dimension transformation. Do this for each dimension that you need to create.

  If you are using several data locales, use the **Identification** tab of the dimension properties window in the SAS Solutions Dimension Editor to add names and descriptions in data locales other than the current data locale.

- In SAS Data Integration Studio, define the dimension in the STAGE_APP_DIMENSION staging table. Move the dimension definition from the staging area to the SDM through the standard series of steps.

  With this method, you can define any number of dimensions in any number of data locales all at once by placing all the necessary specifications in the STAGE_APP_DIMENSION table.

The relative convenience of these three methods can vary with the number of dimensions that you are creating and the number of data locales for each dimension.

## Using the Create Dimension Transformation

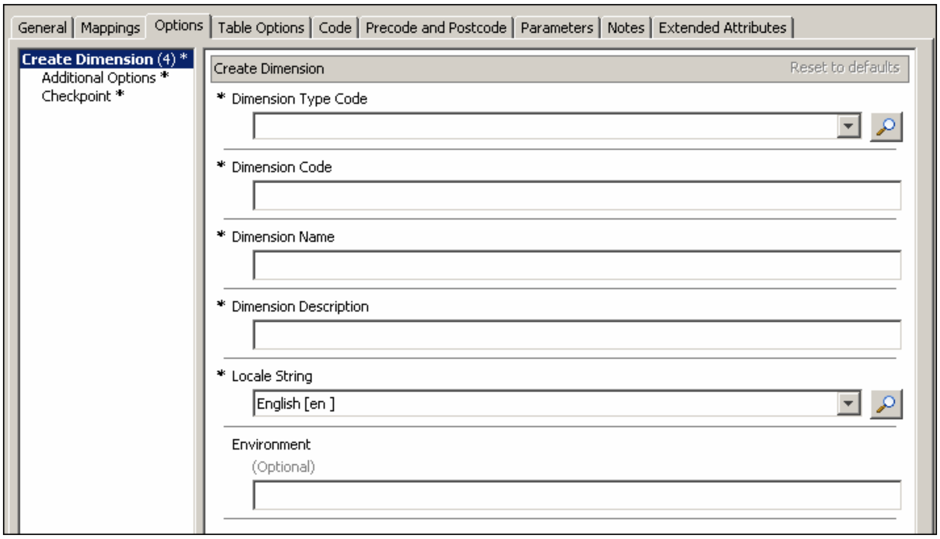
Make a copy of the solnsvc_2200_create_dimension job, using one of the methods in "Copy Jobs" on page 170.

Continue as follows:

1. Double-click the job to display its process diagram.

2. In the process diagram, select the Create Dimension transformation, and then select **Properties** from the pop-up menu.

3. In the Properties window, select the **Options** tab:



Locale String: English [en ]

4. Provide values for the following options:

- **Dimension Type Code** is the code of the dimension type within which the new dimension will be created. To check the spelling of dimension type codes, use the SAS Solutions Dimension Editor.
- **Dimension Code** is a unique code that will identify the new dimension. You must use this code whenever you load members and hierarchies into the dimension (as explained in "Moving Member and Hierarchy Data from the Detail Data Store to the SDM" on page 202) and whenever you load metrics that are associated with members of the dimension (as explained in "Preparing Jobs to Load Metric Data" on page 259).

  A dimension code that is used in a metric table must not be a MySQL reserved word. See "MySQL Reserved Words" on page 735.
- **Dimension Name** and **Dimension Description** identify the new dimension in a way that users will find useful.

  The name and description that you supply here will be associated with the data locale that you specify with the Locale String option. After you create the dimension, you can enter names and descriptions for other data locales using the SAS Solutions Dimension Editor.
- **Locale String** is a string that identifies the data locale that will be associated with the name and description that you specify with the Dimension Name and Dimension Description options. The string concatenates the language and country components of the data locale, using underscores as separators. Here are some examples of well-formed data locale strings:
  - `en` has only a language component.
  - `en_US` concatenates language and country components.
  - `zh_CN` concatenates language and country components.
  - `zh_CN_MAC` concatenates language and country components.

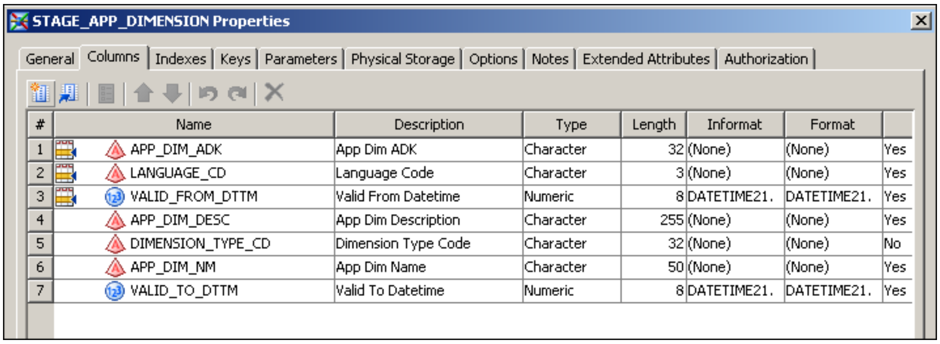
  The data locale that you specify here must already be defined in the SDM at the time that you run this job. For details about loading data locale codes, see "Loading Language Codes and Data Locale Codes" on page 181.
- Run the job and then review the log.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

## Starting from a Staging Table

To create one or more dimensions using a staging table:

1. Write and run a job to load the necessary data into the STAGE_APP_DIMENSION table:



| # | Name | Description | Type | Length | Informat | Format | |
|---|---|---|---|---|---|---|---|
| 1 | APP_DIM_ADK | App Dim ADK | Character | 32 | (None) | (None) | Yes |
| 2 | LANGUAGE_CD | Language Code | Character | 3 | (None) | (None) | Yes |
| 3 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DATETIME21. | DATETIME21. | Yes |
| 4 | APP_DIM_DESC | App Dim Description | Character | 255 | (None) | (None) | Yes |
| 5 | DIMENSION_TYPE_CD | Dimension Type Code | Character | 32 | (None) | (None) | No |
| 6 | APP_DIM_NM | App Dim Name | Character | 50 | (None) | (None) | Yes |
| 7 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DATETIME21. | DATETIME21. | Yes |

On the **Folders** tab, this table is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder.

In building records for this table, note the following:

- App Dim ADK is a unique code that will identify the new dimension. You must use this code whenever you load members and hierarchies into the dimension (as explained in "Moving Member and Hierarchy Data from the Detail Data Store to the SDM" on page 202) and whenever you load metrics that are associated with members of the dimension (as explained in "Preparing Jobs to Load Metric Data" on page 259).

  A dimension code that is used in a metric table must not be a MySQL reserved word. See "MySQL Reserved Words" on page 735.

- Language Code is a Language Code value that is in the CODE_LANGUAGE table. The data locale that the specified language code is associated with in the CODE_LANGUAGE table is the data locale that will be associated with the name and description that you specify in the App Dim Name and App Dim Description columns.

- App Dim Name and App Dim Description identify the new dimension in a way that users will find useful.

  The name and description that you supply here will be associated with the data locale that you specify indirectly with the Language Code. To specify names and descriptions in several data locales for the same dimension, create several records that have the same App Dim ADK value but different Language Code values.

- Dimension Type Code is the code of the dimension type within which the new dimension will be created. To check the spelling of dimension type codes, use the SAS Solutions Dimension Editor.

- Valid From Datetime and Valid To Datetime define the lifespan of the record. See "Setting a Valid Time Range for Data Records" on page 175.

2. Run the cind_dds_101200_load_app_dimension_table job to move the data into the detail data store.

3. Run the solnsvc_2100_create_application_dimension job to load the dimension definitions into the SDM.

   On the **Folders** tab, this job is located in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder.

   The data locales for which you are loading dimension names and descriptions must already be defined in the SDM at the time that you run this job. For details about loading data locale codes, see "Loading Language Codes and Data Locale Codes" on page 181.

4. Run the job and then review the log.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

*Chapter 16*

# Loading Members and Hierarchies into a Dimension

**Ways to Modify the Content of a Dimension . . . 193**

**Moving Member and Hierarchy Data from Its Source to the Staging Tables . . . 194**
Tables for Each Dimension Type . . . 194
Requirements for All or Most Dimension Types . . . 196
Special Requirements for the Currency Dimension Type . . . 197
Special Requirements for the Organization Dimension Type . . . 198
Special Requirements for the Time Dimension Type . . . 200
Users Tab Data . . . 200
Security Tab Data . . . 201

**Loading Members and Hierarchies from the Staging Tables to the Detail Data Store . . . 201**

**Moving Member and Hierarchy Data from the Detail Data Store to the SDM . . 202**
Overview of Moving Member and Hierarchy Data from the Detail Data Store to the SDM . . . 202
Using a Job . . . 202
Using the Load Dimension Wizard . . . 203
Summary of Results . . . 204

## Ways to Modify the Content of a Dimension

There are two ways to modify the members and hierarchies of a dimension, interactively or via an ETL job. Each approach has two variations, as seen in the following table.

***Table 16.1*** *Modifying the Content of a Dimension*

| Method | Details |
|---|---|
| Interactive: editing members and hierarchies directly in the SDM | SAS Solutions Dimension Editor: Select **Members** and use the Members window to work with members and hierarchies. |
| In SAS Data Integration Studio, using the detail data store | Load the members and hierarchies into the detail data store from a third-party software product or another external source. |

This chapter discusses the process of loading members and hierarchies through the detail data store. You can use this method for only one dimension per dimension type. Therefore, for any dimension type that includes more than one dimension, you must choose which dimension to supply with members and hierarchies through the detail data store. All other dimensions that belong to the same dimension type must be supplied with members and hierarchies interactively, using the SAS Solutions Dimension Editor.

*Note:* If only SAS Human Capital Management is installed, loading to the SDM is not required. The only dimension that Human Capital Management requires to be populated in the staging and detail data store tables is the INTERNAL_ORG dimension.

## Moving Member and Hierarchy Data from Its Source to the Staging Tables

### Tables for Each Dimension Type

To load members and hierarchies into a given dimension, use the set of tables for the dimension type to which that dimension belongs. You can use the detail data store to load members and hierarchies into only one dimension per dimension type.

On the **Folders** tab, the staging tables that you must use to load members and hierarchies into a dimension are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder. This folder contains a set of tables for each dimension type. For most dimension types, there are four tables. For the currency and item category dimension types, there are only three tables. The set of staging tables for a dimension type includes the following:

- The primary member table is the table that has the shortest name. For example, the primary member table for the organization dimension type is STAGE_INTERNAL_ORG.

  For most dimension types, the primary member table must contain a row for each member that you are loading, with text in the detail data store default language. For the currency and item category dimension types, the primary member table must contain all the member records that you are loading, regardless of language. The currency and item category primary member tables have a Language Code column. This column identifies the language used in each record. The primary member tables for other dimension types do not have a Language Code column. Therefore, all their records must use the detail data store default language.

  The columns of the primary member tables differ from one dimension type to another, because the members of different dimension types are characterized by different properties. The sections of this chapter on the account dimension type, the organization dimension type, and the time dimension type include illustrations of the primary member tables for those dimension types.

  You can add other columns that represent custom properties to any primary member table, as explained in "Overview of Member Properties " on page 205.

- The secondary member table is the table whose name ends with NLS. For example, the secondary member table for the organization dimension type is STAGE_INTERNAL_ORG_NLS.

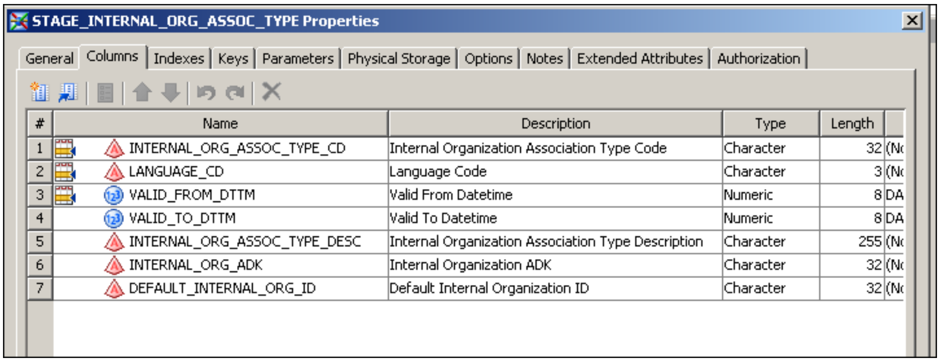
  For most dimension types, the secondary member table is the table in which to place any member records that use languages other than the default detail data store language. For the currency and item category dimension types, there is no secondary member table because their primary member tables can accommodate records in all languages.

If you are loading member records in only one language, then you can ignore the secondary member table. If you use a secondary member table, then any member that you place in it must also be in the associated primary member table.

- The hierarchy identification table is the table whose name ends with ASSOC_TYPE. For example, the hierarchy identification table for the organization dimension is STAGE_INTERNAL_ORG_ASSOC_TYPE.

  The hierarchy identification table must contain a row for each hierarchy that you are loading into the target dimension. If you are loading hierarchy descriptions in more than one language, then this table must contain additional rows that describe the hierarchies in the other languages.

  The hierarchy identification tables have the same column structure for all dimension types, because identifying a hierarchy involves the same considerations regardless of dimension type. Some of the column names differ from one table to another to reflect the different dimension types, but the number, order, and characteristics of the columns are the same. Here are the columns of the STAGE_INTERNAL_ORG_ASSOC_TYPE table:

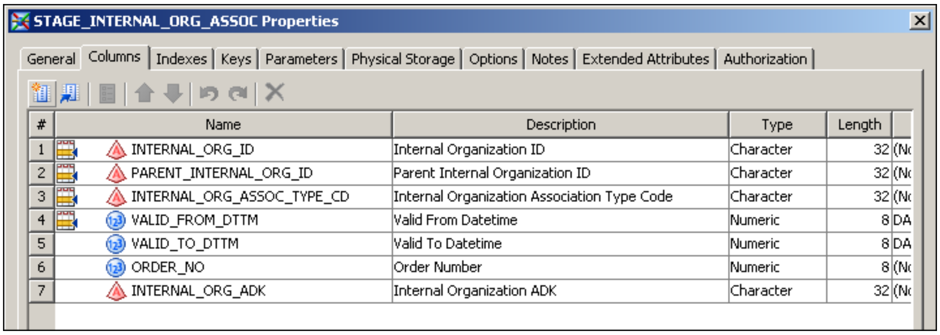


| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | INTERNAL_ORG_ASSOC_TYPE_CD | Internal Organization Association Type Code | Character | 32 | (N |
| 2 | LANGUAGE_CD | Language Code | Character | 3 | (N |
| 3 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DA |
| 4 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DA |
| 5 | INTERNAL_ORG_ASSOC_TYPE_DESC | Internal Organization Association Type Description | Character | 255 | (N |
| 6 | INTERNAL_ORG_ADK | Internal Organization ADK | Character | 32 | (N |
| 7 | DEFAULT_INTERNAL_ORG_ID | Default Internal Organization ID | Character | 32 | (N |

- The hierarchy structure table is the table whose name ends with ASSOC. For example, the hierarchy structure table for the organization dimension is STAGE_INTERNAL_ORG_ASSOC.

  The hierarchy structure table must contain a row for each parent-child relationship within each hierarchy that you are loading into the target dimension. Each row of this table identifies a member, its parent member, and the hierarchy that the relationship is a part of. It also specifies the display position of the member in a fully expanded display of the hierarchy in either the SAS Solutions Dimension Editor.

  The hierarchy structure tables have the same column structure for all dimension types because detailing a hierarchical structure involves the same considerations regardless of dimension type. Some of the column names differ from one table to another to reflect the different dimension types, but the number, order, and characteristics of the columns are the same. Here are the columns of the STAGE_INTERNAL_ORG_ASSOC table:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | INTERNAL_ORG_ID | Internal Organization ID | Character | 32 | (Nc |
| 2 | PARENT_INTERNAL_ORG_ID | Parent Internal Organization ID | Character | 32 | (Nc |
| 3 | INTERNAL_ORG_ASSOC_TYPE_CD | Internal Organization Association Type Code | Character | 32 | (Nc |
| 4 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DA |
| 5 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DA |
| 6 | ORDER_NO | Order Number | Numeric | 8 | (Nc |
| 7 | INTERNAL_ORG_ADK | Internal Organization ADK | Character | 32 | (Nc |

In loading these tables, there are many points to keep in mind. Points that apply across all or most dimension types are discussed in "Requirements for All or Most Dimension Types " on page 196. Points that are specific to a particular dimension type are discussed in subsequent sections.

## Requirements for All or Most Dimension Types

For any dimension type, the data that goes into the member tables, the hierarchy identification table, and the hierarchy structure table must meet the following conditions:

- If the primary member table has a Roll Up to Parent Flag column, then this column must have a value of either **Y** or **N**.
- The hierarchy identification table must contain at least one record.
- In the hierarchy identification table, you can either specify a default member in each record or leave this column blank. If you leave the column blank, then a default member is designated automatically for each hierarchy when the hierarchies are loaded into the SDM. The automatically designated default member is the member in the first record of the hierarchy structure table that describes the relevant hierarchy and that makes a member its own parent.

  All the default members that you specify must also be in the primary member table. If you have several records for the same hierarchy in different languages, then either specify the same default member in all of them or leave them all blank.
- A member can be used in a hierarchy only if it is in the dimension to which the hierarchy belongs. In other words, any member that is in a parent-child record in the hierarchy structure table must also be in the primary member table.
- In the subset of the hierarchy structure table that describes a given hierarchy as of a given moment, each member that occurs as either a parent or a child must occur as a child in exactly one record:
  - If the member has a parent in that hierarchy at that moment, then that one record indicates which member is its parent.
  - If the member has no parent in that hierarchy at that moment, then that one record names the member as its own parent. This is how top-level members are identified.
- The Order Number column of the hierarchy structure table holds integers that determine the top-to-bottom display order of each parent's children in the SAS Solutions Dimension Editor. Among each parent's children, the child with the lowest order number is displayed first, the child with the next lowest order number is displayed second, and so on.

  One approach to assigning order numbers is to assign a unique order number to every record in the table. Another approach is to start a fresh count for the children of each parent. The first approach gives the software more information than it needs, because the hierarchical structure already determines that each member will be displayed as subordinate to its parent. However, you might find that a table with unique order numbers is easier to maintain than a table that reuses the same low numbers many times.

With either approach, it is not necessary to use consecutive integers. For example, by numbering initially with multiples of ten you can provide room to insert new members without having to renumber old members.

If you leave this column blank in all the records of a hierarchy structure table, then the software assigns default order numbers that reflect the order of the records in the table.

*Note:* The Order Number column does not affect the display of organization hierarchies in SAS Human Capital Management.

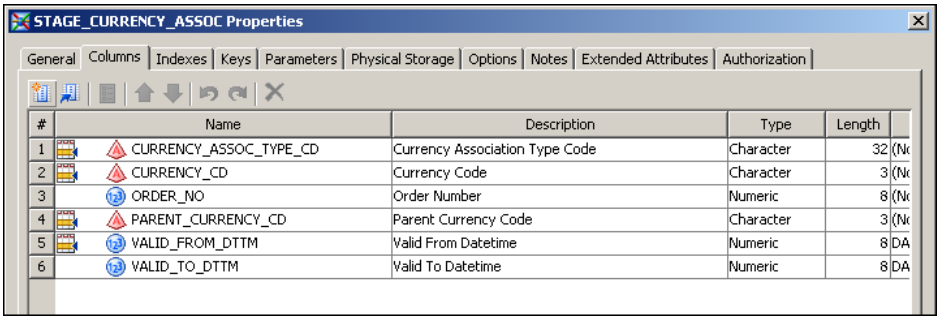
- The records of the hierarchy structure table can occur in any order, but it is a good idea to load this table so that the records are grouped by hierarchy.
- Each table includes a Valid From Datetime column and a Valid To Datetime column, which define the lifespans of its records. See "Setting a Valid Time Range for Data Records" on page 175.

As indicated by the sequence numbers of the jobs, the tables must be loaded in the following order:

1. The primary member table must be loaded first. This is because the members in it are used to validate the default members that are specified in the hierarchy identification table and all the members that are specified in parent-child relations in the hierarchy structure table.
2. The hierarchy identification (ASSOC_TYPE) table must be loaded second. This is because the hierarchies that are identified are used to validate the hierarchies that are fully specified in the hierarchy structure (ASSOC) table.
3. The hierarchy structure (ASSOC) table must be loaded third. Both the primary member table and the hierarchy identification table are used to validate it.
4. The secondary member (NLS) table must be loaded last, if there is one and if you need to use it.

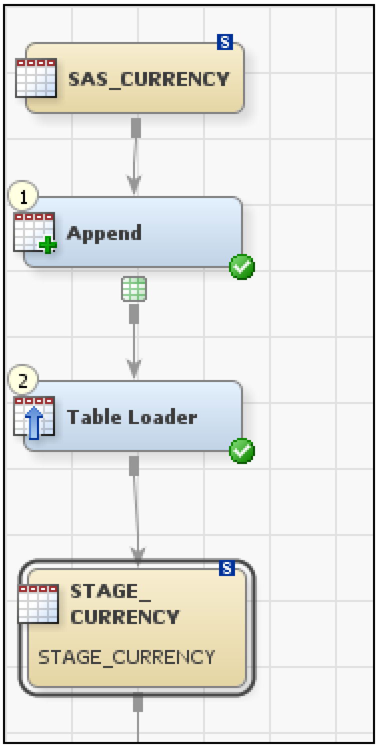
## Special Requirements for the Currency Dimension Type

In the Currency dimension type, every hierarchy must be flat. Every record that you load into the STAGE_CURRENCY_ASSOC hierarchy structure table must have the same currency code in the Currency Code and Parent Currency Code columns:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | CURRENCY_ASSOC_TYPE_CD | Currency Association Type Code | Character | 32 | (N |
| 2 | CURRENCY_CD | Currency Code | Character | 3 | (N |
| 3 | ORDER_NO | Order Number | Numeric | 8 | (N |
| 4 | PARENT_CURRENCY_CD | Parent Currency Code | Character | 3 | (N |
| 5 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DA |
| 6 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DA |

The STAGE_CURRENCY table is loaded from the SAS_CURRENCY table of predefined data by the cind_dds_102400_load_currency_table job:



This is the only dimension staging table for which you do not need to write your own job.

## Special Requirements for the Organization Dimension Type

If you are using SAS Human Capital Management (HCM), then the following macro variables in the prebuild.sas file must have values that correctly describe the organization dimension and the organization hierarchies in the INTERNAL_ORG detail data store tables that are used by SAS Human Capital Management:

- HIERS
- NUMBER_OF_HIERS

For details about these HCM macro variables, see “Macro Variables in the PREBUILD.SAS Macro File” on page 304.

The rest of this section is relevant no matter what solution or combination of solutions you are using.

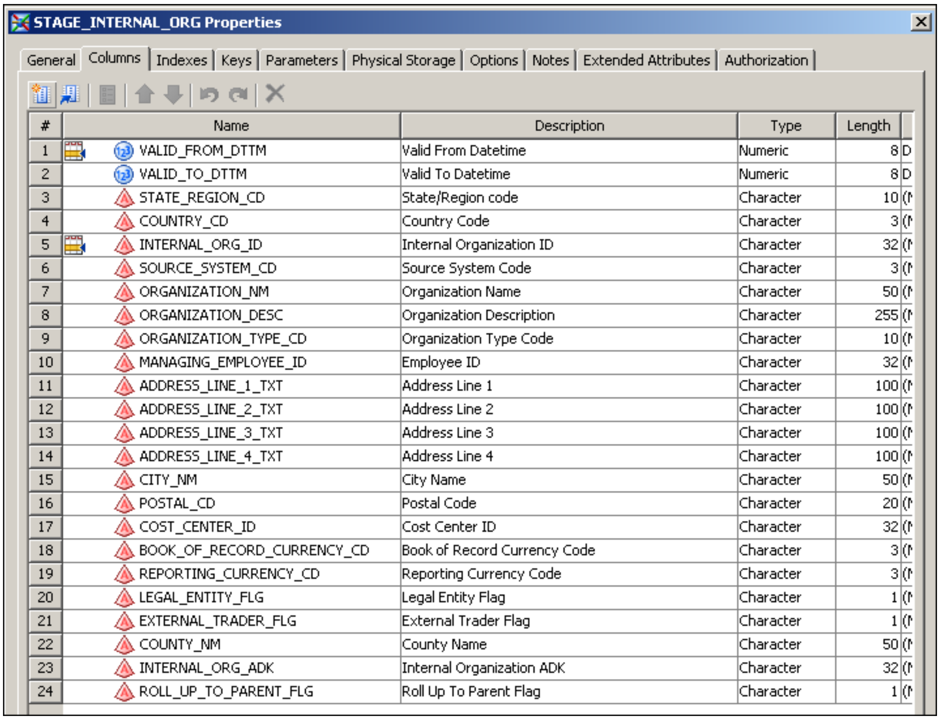
The STAGE_INTERNAL_ORG table must contain two special members, which are not visible in the solution software. One special member is defined by an Internal Organization ID of ALL. The other special member is defined by an Internal Organization ID of EXT.

The ALL and EXT members must be part of every hierarchy that is defined in the STAGE_INTERNAL_ORG_ASSOC table. In every organization hierarchy, ALL must be the unique top member, and EXT must be a leaf that is directly under ALL. The formal constraints are as follows:

- ALL must not have a parent. This is indicated by a record in which ALL is its own parent.

- ALL must be the only member of the hierarchy that does not have a parent.
- ALL must be the parent of EXT.
- EXT must not be the parent of any member.

Each member of an internal organization dimension has properties that correspond to the columns of the STAGE_INTERNAL_ORG table:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | D |
| 2 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | D |
| 3 | STATE_REGION_CD | State/Region code | Character | 10 | ( |
| 4 | COUNTRY_CD | Country Code | Character | 3 | ( |
| 5 | INTERNAL_ORG_ID | Internal Organization ID | Character | 32 | ( |
| 6 | SOURCE_SYSTEM_CD | Source System Code | Character | 3 | ( |
| 7 | ORGANIZATION_NM | Organization Name | Character | 50 | ( |
| 8 | ORGANIZATION_DESC | Organization Description | Character | 255 | ( |
| 9 | ORGANIZATION_TYPE_CD | Organization Type Code | Character | 10 | ( |
| 10 | MANAGING_EMPLOYEE_ID | Employee ID | Character | 32 | ( |
| 11 | ADDRESS_LINE_1_TXT | Address Line 1 | Character | 100 | ( |
| 12 | ADDRESS_LINE_2_TXT | Address Line 2 | Character | 100 | ( |
| 13 | ADDRESS_LINE_3_TXT | Address Line 3 | Character | 100 | ( |
| 14 | ADDRESS_LINE_4_TXT | Address Line 4 | Character | 100 | ( |
| 15 | CITY_NM | City Name | Character | 50 | ( |
| 16 | POSTAL_CD | Postal Code | Character | 20 | ( |
| 17 | COST_CENTER_ID | Cost Center ID | Character | 32 | ( |
| 18 | BOOK_OF_RECORD_CURRENCY_CD | Book of Record Currency Code | Character | 3 | ( |
| 19 | REPORTING_CURRENCY_CD | Reporting Currency Code | Character | 3 | ( |
| 20 | LEGAL_ENTITY_FLG | Legal Entity Flag | Character | 1 | ( |
| 21 | EXTERNAL_TRADER_FLG | External Trader Flag | Character | 1 | ( |
| 22 | COUNTY_NM | County Name | Character | 50 | ( |
| 23 | INTERNAL_ORG_ADK | Internal Organization ADK | Character | 32 | ( |
| 24 | ROLL_UP_TO_PARENT_FLG | Roll Up To Parent Flag | Character | 1 | ( |

In building records for this table, note the following:

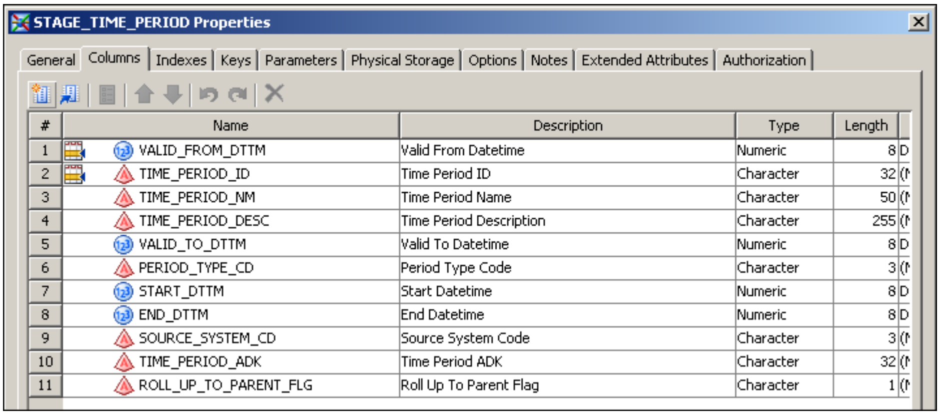
- Valid From Datetime and Valid To Datetime define the lifespan of the record. See "Setting a Valid Time Range for Data Records" on page 175.
- Employee ID must have a value if you are using SAS Human Capital Management. Otherwise, leave this column blank. Each value that you use must be defined in the DDS EMPLOYEE table.
- Reporting Currency Code corresponds to the Functional Currency property in SAS Financial Management Studio. You must provide a valid currency code for each organization, including ALL and EXT.

  If you are not using SAS Financial Management, then you can specify any currency code for each organization.
- Book of Record Currency Code is not used.
- Legal Entity Flag corresponds to the Reporting Entity property in SAS Financial Management Studio. Use Y for any organization that is a reporting entity and N for any organization that is not a reporting entity. For ALL and EXT, use N.

  If you are not using SAS Financial Management, then you can leave this column bank. In this case, the value N is supplied automatically for every record.
- Internal Organization ADK is not used.

## *Special Requirements for the Time Dimension Type*

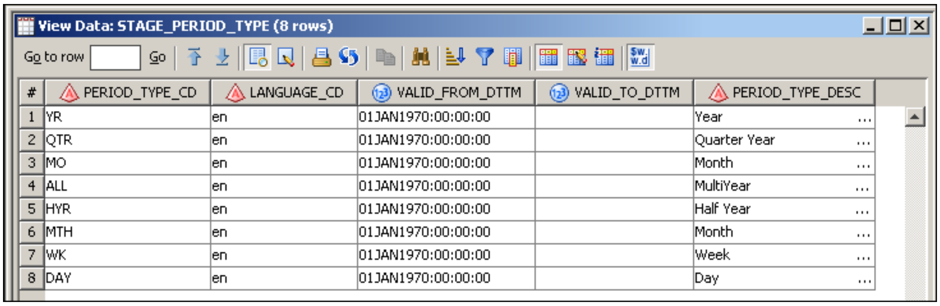
Each member of a time dimension has properties that correspond to the columns of the STAGE_TIME_PERIOD table:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | D |
| 2 | TIME_PERIOD_ID | Time Period ID | Character | 32 | ( |
| 3 | TIME_PERIOD_NM | Time Period Name | Character | 50 | ( |
| 4 | TIME_PERIOD_DESC | Time Period Description | Character | 255 | ( |
| 5 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | D |
| 6 | PERIOD_TYPE_CD | Period Type Code | Character | 3 | ( |
| 7 | START_DTTM | Start Datetime | Numeric | 8 | D |
| 8 | END_DTTM | End Datetime | Numeric | 8 | D |
| 9 | SOURCE_SYSTEM_CD | Source System Code | Character | 3 | ( |
| 10 | TIME_PERIOD_ADK | Time Period ADK | Character | 32 | ( |
| 11 | ROLL_UP_TO_PARENT_FLG | Roll Up To Parent Flag | Character | 1 | ( |

For each record in this table, note the following:

- Valid From Datetime and Valid To Datetime define the lifespan of the record. See "Setting a Valid Time Range for Data Records" on page 175.
- Start Date and End Date define the time period that the member represents. You must place counts of seconds from January 1, 1960:00:00:00 in both these columns. This is so even though the solution software shows only calendar dates. Do not put counts of days from January 1, 1960 in these columns.
- Period Type Code must be one of the codes in the SAS_PERIOD_TYPE table:



| # | PERIOD_TYPE_CD | LANGUAGE_CD | VALID_FROM_DTTM | VALID_TO_DTTM | PERIOD_TYPE_DESC |
|---|---|---|---|---|---|
| 1 | YR | en | 01JAN1970:00:00:00 | | Year ... |
| 2 | QTR | en | 01JAN1970:00:00:00 | | Quarter Year ... |
| 3 | MO | en | 01JAN1970:00:00:00 | | Month ... |
| 4 | ALL | en | 01JAN1970:00:00:00 | | MultiYear ... |
| 5 | HYR | en | 01JAN1970:00:00:00 | | Half Year ... |
| 6 | MTH | en | 01JAN1970:00:00:00 | | Month ... |
| 7 | WK | en | 01JAN1970:00:00:00 | | Week ... |
| 8 | DAY | en | 01JAN1970:00:00:00 | | Day ... |

- Ignore Time Period ADK. It is not used.

## *Users Tab Data*

For dimensions of every dimension type except analysis, currency, and time, the member properties window in the SAS Solutions Dimension Editor includes a **Users** tab. You can load the user-member associations that can be viewed and edited with this tab.

These user-member associations can serve the following purposes:

- In the KPI Viewer, a user who has a **Users** tab user-member association with a certain dimension member has default Read access to any scorecard that is assigned to that dimension member.

To load **Users** tab information, use the STAGE_APP_USER_X_MEMBER staging table. For information about the columns of this table, see the Data Model part of this book.

### *Security Tab Data*

For dimensions of every dimension type, the member properties window in the SAS Solutions Dimension Editor includes a **Security** tab. You can load the user-member and group-member associations that can be viewed and edited with this tab.

To load **Security** tab user-member associations, use the STAGE_APP_USER_ACTIONS staging table. To load **Security** tab group-member associations, use the STAGE_APP_GROUP_ACTIONS staging table. For information about the columns of these tables, see the Data Model part of this book.

## Loading Members and Hierarchies from the Staging Tables to the Detail Data Store

On the **Folders** tab, the jobs for loading members and hierarchies into the detail data store are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder. There is a separate job for loading each detail data store table from the corresponding staging table. For example, loading members and hierarchies into an organization dimension involves the following four jobs:

- cind_dds_106200_load_internal_org_table (all members in the detail data store default language)
- cind_dds_106230_load_internal_org_nls_table (members in other supported languages)
- cind_dds_106210_load_internal_org_assoc_type_table (identities of all hierarchies)
- cind_dds_106220_load_internal_org_assoc_table (parent-child structure of all hierarchies)

You will need to run jobs that handle specific types of member properties across dimension types if you meet one of the following conditions:

- You are loading original **Users** tab data as described in “Users Tab Data” on page 200.
- You are loading original **Security** tab data as described in “Security Tab Data” on page 201.

To load exported information about formulas for any dimension type that supports formulas, run the following jobs:

- cind_dds_108200_load_app_formula_target_table
- cind_dds_108210_load_app_formula_table
- cind_dds_108220_load_app_formula_write_member_table
- cind_dds_108230_load_app_formula_read_member_table

To load exported or original **Users** tab information for any dimension type except analysis, currency, and time, run the following job:

- cind_dds_108300_load_app_user_x_member_table

To load exported or original **Security** tab information for any dimension type, run the following jobs:

- cind_dds_108310_load_app_user_actions_table
- cind_dds_108320_load_app_group_actions_table

On the **Folders** tab, these cross-dimension-type jobs are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder.

# Moving Member and Hierarchy Data from the Detail Data Store to the SDM

## *Overview of Moving Member and Hierarchy Data from the Detail Data Store to the SDM*

You can load members and hierarchies into a dimension in the SDM using either one of the following:

- a SAS Data Integration Studio job that uses the Import Dimension transformation
- the Load Dimension wizard in the SAS Solutions Dimension Editor

*Note:* You do not need to load members and hierarchies into a dimension in the SDM if only SAS Human Capital Management is installed at your site. If your site has SAS Financial Management or SAS Strategy Management, and you want to view HCM metrics in SAS Strategy Management, you can load members and hierarchies.

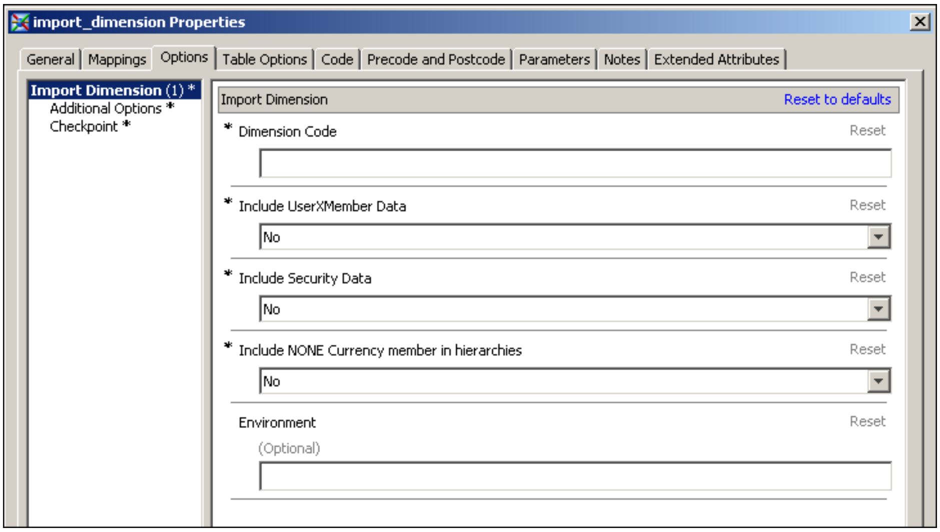
Typically, you can handle your dimensions in any order. The only exception is that you must load currencies into a currency dimension before you load organizations into an organization dimension.

The data locales for which you are loading member and hierarchy names and descriptions must be defined in the SDM before you load the member and hierarchy data. For details about loading data locales, see "Loading Language Codes and Data Locale Codes" on page 181.

## *Using a Job*

To use a SAS Data Integration Studio job, first prepare the job in the following way:

1. In the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder on the **Folders** tab, make a copy of the solnsvc_3200_load_dimension job, using one of the methods described in "Copy Jobs" on page 170.
2. In the process diagram, select the Import Dimension transformation, and then select **Properties** from the pop-up menu.
3. Select the **Options** tab:



import_dimension Properties

General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
Import Dimension (1) *
Additional Options *
Checkpoint *
Reset to defaults
Reset

Import Dimension

* Dimension Code

* Include UserXMember Data: No

* Include Security Data: No

* Include NONE Currency member in hierarchies: No

Environment (Optional)

4. Provide values for the following options:

   - **Dimension Code** is the code of the target dimension. You can look this up in the SAS Solutions Dimension Editor.

   - **Include UserXMember Data** is a Yes/No flag. Select **Yes** in order to import the user-member associations that can be viewed in SAS Financial Management Studio on the **Users** tab of the member properties window. Select **No** if you are not importing this information.

     If you select **Yes**, then all information of this type for the target dimension is deleted from the SDM before the new information is imported.

   - **Include Security Data** is a Yes/No flag. Select **Yes** in order to import the user-member associations that can be viewed in SAS Financial Management Studio on the **Security** tab of the member properties window. Select **No** if you are not importing this information.

     If you select **Yes**, then all information of this type for the target dimension will be deleted from the SDM before the new information is imported.

5. Save the job.

Run the job and then review the log. The log lists the location of an HTML report of the results.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

## Using the Load Dimension Wizard

To use the Load Dimension wizard in the SAS Solutions Dimension Editor:

1. Select the target dimension from the displayed list of dimensions.
2. Select **Load Dimension** to launch the Load Dimension wizard.
3. Proceed through the Load Dimension wizard, referring to the online Help as necessary.

When the load process is complete, a window appears from which you can view an HTML report of the results.

### Summary of Results

Whether you load members and hierarchies using a job or using the Load Dimension wizard, the results are the same:

- All the detail data store data in the dimension-type-specific tables for the relevant dimension type is loaded. This includes the data in the primary and secondary member tables, the hierarchy identification table, and the hierarchy structure table. For an account dimension, it also includes the data in the SOURCE_GL_ACCOUNT table.
- Each member that you load replaces the existing member that has the same code, if there is one. Any existing member that is not replaced by a newly loaded member remains in the target dimension.
- For each member that you load, any associated formula data is also loaded. Associated **Security** tab data and **User** tab data is loaded only if you set the relevant flags to **Y**. Any dimension that is used in a formula must be loaded before the dimension with which the formula is associated.
- Each hierarchy that you load replaces the entire existing hierarchy that has the same code, if there is one. Any existing hierarchy that is not replaced by a newly loaded hierarchy remains in the target dimension.

# *Chapter 17* Registering Member Properties So That They Are Loaded into the SDM

**Overview of Member Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
**Member Properties That Are Preregistered** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
**Defining New Member Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
**Registering Member Properties** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
**Using Member Properties That You Have Registered** . . . . . . . . . . . . . . . . . . . . . . 208

## Overview of Member Properties

Some of the columns in a primary member table contain information that is common across all or most dimension types, while other columns contain information that is specific to the dimension type in question. The columns that contain dimension-type-specific information represent member *properties*. Examples are Account Type for the account dimension type and Functional Currency for the organization dimension type. The generic columns represent member attributes that are not classified as member properties. Examples are Code, Name, Description, Valid from Datetime, Valid to Datetime, and Roll Up to Parent Flag.

When you load members into a dimension in the SDM, the information that is loaded includes all the generic columns and the values of those member properties that are registered to be loaded. Many but not all member properties are preregistered in the software. You can register more member properties, including member properties that you add to the detail data store and member properties that are predefined in the detail data store but not preregistered.

## Member Properties That Are Preregistered

For the account and time dimension types, all predefined member properties are preregistered.

For the organization dimension type, the following predefined member properties are preregistered:

- Reporting Currency Code
- Legal Entity Flag

## Defining New Member Properties

For any dimension type, you can define new member properties in the detail data store by doing the following:

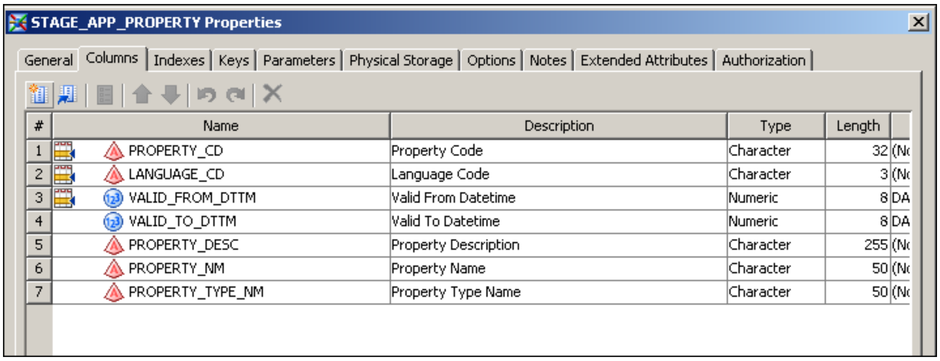
1. Add a column for the new property to the relevant staging primary member table.
2. Modify the job that you wrote to load the staging primary member table so that it loads values into the new column.
3. Add a matching column for the new property to the corresponding detail data store primary member table.
4. Modify the relevant detail data store job so that it moves values from the new column of the staging primary member table to the matching new column of the detail data store primary member table.

*Note:* After you add a column to a table, right-click the table name and select **Update Metadata**.

## Registering Member Properties

To register a predefined member property or a member property that you have added to the detail data store, do the following:

1. In the STAGE_APP_PROPERTY table, add a row that describes the property. This table is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder on the **Folders** tab. It has the following columns:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | PROPERTY_CD | Property Code | Character | 32 | (N |
| 2 | LANGUAGE_CD | Language Code | Character | 3 | (N |
| 3 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DA |
| 4 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DA |
| 5 | PROPERTY_DESC | Property Description | Character | 255 | (N |
| 6 | PROPERTY_NM | Property Name | Character | 50 | (N |
| 7 | PROPERTY_TYPE_NM | Property Type Name | Character | 50 | (N |

Each row that you add to this table must satisfy the following constraints:

- The Property Code column must contain the same value as the Property Code column of the corresponding record in the STAGE_APP_MEMBER_PROPERTY_MAP table. Do not use any of the following reserved property codes:

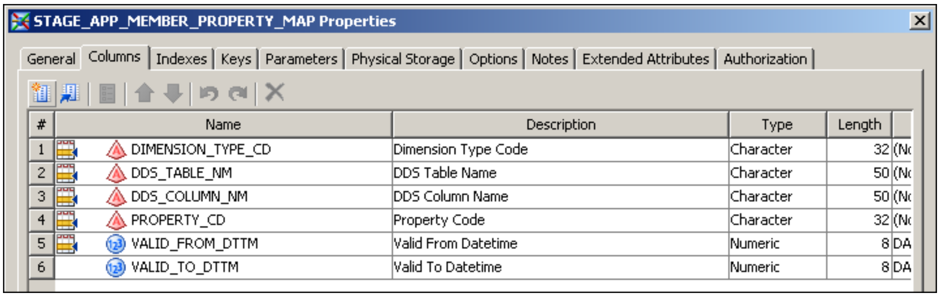
```
AccountBehavior
AccountType
BalanceType
BasisData
BookCurrency
```

```
EndDate
ExchangeRateType
Formula
FormulaId
FormulaPrecedence
FormulaScope
FormulaType
FunctionalCurrency
Intercompany
Level
ReportingEntity
RollForwardMethod
SourceAccounts
StartDate
TotalAfterImport
```

- The Property Type Name column identifies the data type of the property's values. It must contain one of the following strings:

```
boolean
date
double
integer
string
```

2. Add to the STAGE_APP_MEMBER_PROPERTY_MAP table a row that associates the property with the DDS column that holds its values. This table is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder. It has the following columns:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | DIMENSION_TYPE_CD | Dimension Type Code | Character | 32 | (N |
| 2 | DDS_TABLE_NM | DDS Table Name | Character | 50 | (N |
| 3 | DDS_COLUMN_NM | DDS Column Name | Character | 50 | (N |
| 4 | PROPERTY_CD | Property Code | Character | 32 | (N |
| 5 | VALID_FROM_DTTM | Valid From Datetime | Numeric | 8 | DA |
| 6 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DA |

Each row that you add to this table must satisfy the following constraints:

- The DDS Column Name column must contain the name of the column that contains the values of the property.

- The Property Code column must contain the same value as the Property Code column of the corresponding record in the STAGE_APP_PROPERTY table.

3. Run the cind_dds_101300_load_app_property_table job.

4. Run the cind_dds_101400_load_app_member_property_map_table job.

## Using Member Properties That You Have Registered

After additional member properties have been loaded into the SDM, you can view their values using either the Members view or the Hierarchies view in the SAS Solutions Dimension Editor. You can view information about the properties themselves in the Custom Properties view.

If you want the solution software to do anything else with the values of custom properties, talk to your SAS consultant about customizing the solution software.

# *Chapter 18*
# Exporting and Promoting Members and Hierarchies

**Overview of Exporting Members and Hierarchies** . . . . . . . . . . . . . . . . . . . . . . . . . 209
**Using a Job to Export Members and Hierarchies** . . . . . . . . . . . . . . . . . . . . . . . . . 210
**Using the Export Dimension Wizard to Export Members and Hierarchies** . . . . . 212
**Details of the Result** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
**Possible Obstacles to Exporting a Dimension** . . . . . . . . . . . . . . . . . . . . . . . . . . . 213

## Overview of Exporting Members and Hierarchies

The need to export members and hierarchies from the SDM is most likely to arise for users of SAS Financial Management. It can arise for either of two reasons.

When you export members and hierarchies, you can choose the export destination. You can also choose whether to export **Users** tab information and **Security** tab information for the exported members. Your reason for performing the export operation has implications for both choices.

Here are the two reasons and their implications:

- You have created members using the Dimensions workspace of SAS Financial Management Studio. You now want to use these members in base accounting facts to be loaded through the GL_TRANSACTION_SUM and GL_JRNL_DETAILS tables of the detail data store. This requires that the members be in the appropriate detail data store dimension tables.

  In this case, the appropriate export destination is the staging area that serves the SDM that you are exporting from. From that staging area, you load the information into the detail data store dimension tables, as explained in "Loading Members and Hierarchies from the Staging Tables to the Detail Data Store" on page 201.

  There is no need to export **Users** tab and **Security** tab information, because this information is not used in the process of loading base accounting facts.

- You have created or modified members and hierarchies using the Dimensions workspace of SAS Financial Management Studio. You now want to promote these members and hierarchies to a test system or to a production system.

  In this case, the appropriate export destination is a Base SAS library other than the staging area that serves the SDM that you are exporting from. After you export the members and hierarchies to this library, you must move them to the staging area that serves the system that is your promotion target. From that staging area, you load the members and hierarchies into the detail data store dimension tables for your promotion target, as explained in "Loading Members and Hierarchies from the Staging Tables to the Detail Data Store" on page 201. From the detail data store dimension tables for your promotion target, you load the members and hierarchies into the SDM for your promotion target, as explained in "Moving Member and Hierarchy Data from the Detail Data Store to the SDM" on page 202.

Depending on how you manage **Users** tab and **Security** tab information across the two systems, you might or might not want to export **Users** tab and **Security** tab information as part of the promotion operation.

Export members and hierarchies only for those dimensions that you load through the detail data store. Remember that you must choose a single dimension per dimension type to load with members and hierarchies through the detail data store.

You can export the members and hierarchies of a dimension in two ways:

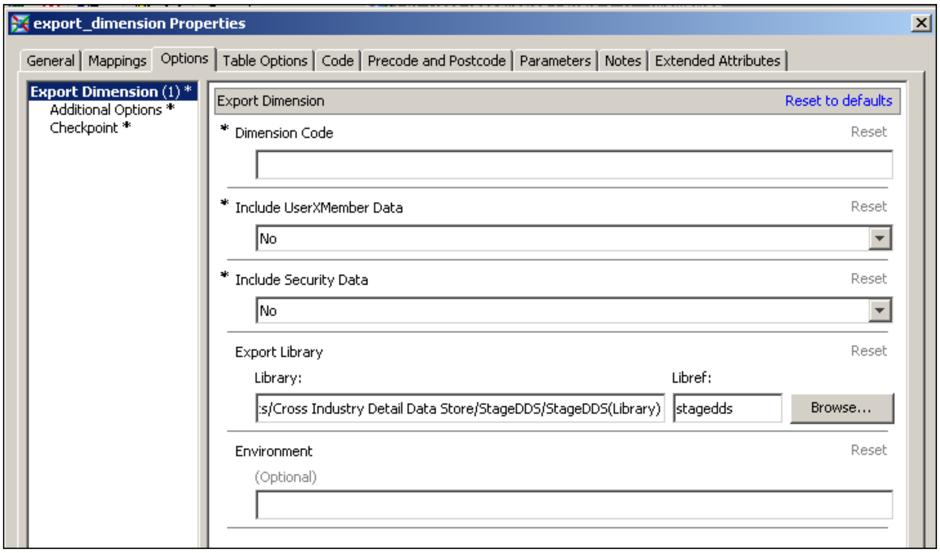
- Run a SAS Data Integration Studio job that uses the Export Dimension transformation.
- Run the Export Dimension wizard in the SAS Solutions Dimension Editor.

Both methods yield the same result. Both methods are available regardless of the reason for the export operation.

## Using a Job to Export Members and Hierarchies

To use a SAS Data Integration Studio job, first prepare the job in the following way:

1. On the **Folders** tab, in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder, make a copy of the solnsvc_4100_export_dimension job, using one of the methods described in "Copy Jobs" on page 170.
2. In the process diagram, select the export_dimension transformation, and then select **Properties** from the pop-up menu. In the Properties window, select the **Options** tab:


export_dimension Properties

Export Dimension

* Dimension Code

* Include UserXMember Data: No

* Include Security Data: No

Export Library
Library: :s/Cross Industry Detail Data Store/StageDDS/StageDDS(Library)
Libref: stagedds

Environment
(Optional)

3. Provide values for the following options:
   - **Dimension Code** is the code of the source dimension. You can look this up in the SAS Solutions Dimension Editor.

- **Include UserXMember Data** is a Yes/No flag. Select **Yes** in order to export the user-member associations that can be viewed in SAS Financial Management Studio on the **Users** tab of the member properties window. Select **No** in order to withhold these user-member associations from the exported information. **Yes** is appropriate only if you are exporting members and hierarchies in order to promote them to another system.

- **Include Security Data** is a Yes/No flag. Select **Yes** to export the user-member associations that can be viewed in SAS Financial Management Studio on the **Security** tab of the member properties window. Select **No** to withhold these user-member associations from the exported information. **Yes** is appropriate only if you are exporting members and hierarchies in order to promote them to another system.

- **Export Library** is the name of the Base SAS data library that you are exporting the data to. Click **Browse** to select a library. For example, select `stagedds` if you are exporting members and hierarchies to the staging area. If you specify a target library other than `stagedds`, then make sure that the target library satisfies the following conditions:

  *Note:* Do not export data to the CrossIndustryDDS library.

  - It is on a machine that uses the same operating system as the machine that holds the source SDM.

  - The Solutions Host User has operating system read and write access to it.

  - It contains copies of all the staging tables that are needed to receive the exported data. These include the following:

    - dimension-type-specific tables for each dimension type that you are working with. For the account dimension type, you need copies of the following five tables: STAGE_GL_ACCOUNT, STAGE_GL_ACCOUNT_ASSOC_TYPE, STAGE_GL_ACCOUNT_ASSOC, STAGE_GL_ACCOUNT_NLS, STAGE_SOURCE_GL_ACCOUNT. For most other dimension types, you need the counterparts of the first four of these tables. For the currency and item category dimension types, you need the counterparts of the first three.

    - tables that contain formula information across all dimension types that support formulas: STAGE_APP_FORMULA, STAGE_APP_FORMULA_TARGET, STAGE_APP_FORMULA_READ_MEMBER, STAGE_APP_FORMULA_WRITE_MEMBER.

    - tables that contain **Security** tab data across all dimension types: STAGE_APP_GROUP_ACTIONS, STAGE_APP_USER_ACTIONS.

    - the table that contains **User** tab data across all dimension types except analysis, currency, and time (which do not support **User** tab data): STAGE_APP_USER_X_MEMBER.

  *Note:* To define additional Base SAS data libraries, use SAS Management Console.

4. Save the job.

5. Run the job and then review the log.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

## Using the Export Dimension Wizard to Export Members and Hierarchies

To export the members and hierarchies of a selected dimension using the Export Dimension wizard:

1. In the SAS Solutions Dimension Editor, select the source dimension.
2. Select **Export this dimension** to launch the Export Dimension wizard.
3. Proceed through the wizard, referring to the online Help as necessary.

   If you specify an export library other than `stagedds`, then the export library must satisfy all the conditions that are listed in "Using a Job to Export Members and Hierarchies " on page 210 .

## Details of the Result

The two methods of exporting members and hierarchies produce the same result. Characteristics of this result include the following:

- All the data in the target dimension-type-specific tables is deleted, and then the data that you are exporting is placed in them. At the end of the process, these tables contain only the data that you have just exported.

  If the SDM contains member or hierarchy names and descriptions in more than one data locale, then the export includes names and descriptions in each data locale that is defined in the Detail Data Store CODE_LANGUAGE table. For a detailed discussion of this table, see "Loading Language Codes and Data Locale Codes" on page 181. The names and descriptions for the data locale that is associated with the detail data store default language are exported to the primary member table. The names and descriptions for all other data locales are exported to the secondary member table. For a detailed discussion of these tables, see "Tables for Each Dimension Type " on page 194.
- All the data in the target formula tables for the dimension type that you are working with is deleted, and then the formula data that you are exporting is placed in them. At the end of the process, these tables contain only the freshly exported formula data for the dimension type that you are working with plus the previously present formula data for all other dimension types.
- If you choose to export **Security** tab data, all the data in the target **Security** tab tables for the dimension type that you are working with is deleted, and then the **Security** tab data that you are exporting is placed in them. In this case, at the end of the process, these tables contain only the freshly exported **Security** tab data for the dimension type that you are working with plus the previously present **Security** tab data for all other dimension types.

  If you choose to not export **Security** tab data, then the export operation does not change the target **Security** tab tables in any way.

- If you choose to export **User** tab data, all the data in the target **User** tab table for the dimension type that you are working with is deleted. The **User** tab data that you are exporting is placed in the table. At the end of the process, this table contains only the freshly exported **User** tab data for the dimension type that you are working with, plus the previously present **User** tab data for all other dimension types.

  If you choose to not export **User** tab data, then the export operation does not change the target **User** tab table in any way.

## Possible Obstacles to Exporting a Dimension

The solnsvc_4100_export_dimension job and the Export Dimension wizard can encounter various obstacles that prevent them from successfully exporting the members and hierarchies of the selected dimension. Possible obstacles include the following:

- The Solutions Host User does not have operating system read and write access to the target data library.
- A target table does not exist. If the target data library is the staging area, this can happen if a table was accidentally deleted or if the staging tables for the relevant dimension type were never created. For a detailed discussion of the process of creating a dimension type, see "Adding a Dimension Type" on page 215.

  For a target data library other than the staging area, this can happen if you neglected to copy one of the necessary tables into the target library.
- A column is either misnamed or missing from a target table. This can happen if the target tables were not created correctly.
- One of the target tables is open and locked. This can happen if someone is working with the table.
- The CODE_LANGUAGE table in the detail data store does not have one and only one record marked with a Default Language Flag value of Y. For a detailed discussion of this table, see "Loading Language Codes and Data Locale Codes" on page 181.
- The CODE_LANGUAGE table in the detail data store does not have a record for one of the languages that is used in the member and hierarchy data that you want to export.

If the job or the wizard encounters any of these obstacles, an appropriate message is displayed.

*Chapter 19*

# Adding a Dimension Type

**Overview of Adding a Dimension Type** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 215

**Run the Job That Creates a New Dimension Type in the Staging Tables** . . . . . . . 216

**Write Jobs to Load the New Staging Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220

**Create the Jobs That Load the New Detail Data Store Tables** . . . . . . . . . . . . . . . . 220

**Customize the Job That Loads the Detail Data Store Primary Member Table** . . 221

**Customize the Job That Loads the Detail Data Store Hierarchy Identification Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 227

**Customize the Job That Loads the Detail Data Store Hierarchy Structure Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 236

**Customize the Job That Loads the Detail Data Store Secondary Member Table** 247

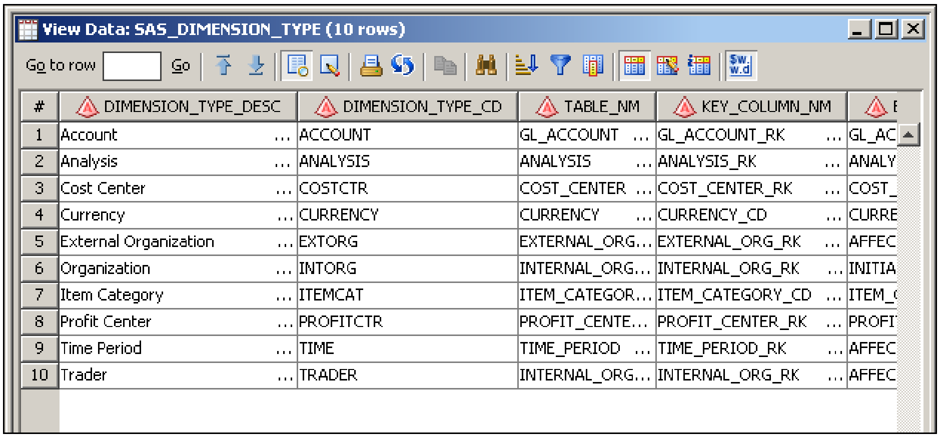
**Loading New Dimension Types into the SDM** . . . . . . . . . . . . . . . . . . . . . . . . . . . 251

**Creating Dimensions in a New Dimension Type** . . . . . . . . . . . . . . . . . . . . . . . . . 251

**Loading Members and Hierarchies into a Dimension That Belongs to a New Dimension Type** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251

## Overview of Adding a Dimension Type

The installed SAS Solutions Services software includes the dimension types that are defined in the SAS_DIMENSION_TYPE table:



| # | DIMENSION_TYPE_DESC | DIMENSION_TYPE_CD | TABLE_NM | KEY_COLUMN_NM | E |
|---|---|---|---|---|---|
| 1 | Account | ACCOUNT | GL_ACCOUNT | GL_ACCOUNT_RK | GL_AC |
| 2 | Analysis | ANALYSIS | ANALYSIS | ANALYSIS_RK | ANALY |
| 3 | Cost Center | COSTCTR | COST_CENTER | COST_CENTER_RK | COST_ |
| 4 | Currency | CURRENCY | CURRENCY | CURRENCY_CD | CURRE |
| 5 | External Organization | EXTORG | EXTERNAL_ORG... | EXTERNAL_ORG_RK | AFFEC |
| 6 | Organization | INTORG | INTERNAL_ORG... | INTERNAL_ORG_RK | INITIA |
| 7 | Item Category | ITEMCAT | ITEM_CATEGOR... | ITEM_CATEGORY_CD | ITEM_ |
| 8 | Profit Center | PROFITCTR | PROFIT_CENTE... | PROFIT_CENTER_RK | PROFI |
| 9 | Time Period | TIME | TIME_PERIOD | TIME_PERIOD_RK | AFFEC |
| 10 | Trader | TRADER | INTERNAL_ORG... | INTERNAL_ORG_RK | AFFEC |

This predefined set of dimension types might or might not meet your needs. This chapter provides instructions for adding another dimension type. You must work through the entire chapter in order, without skipping any steps. To add two or more dimension types, repeat the steps that are described here as many times as necessary.

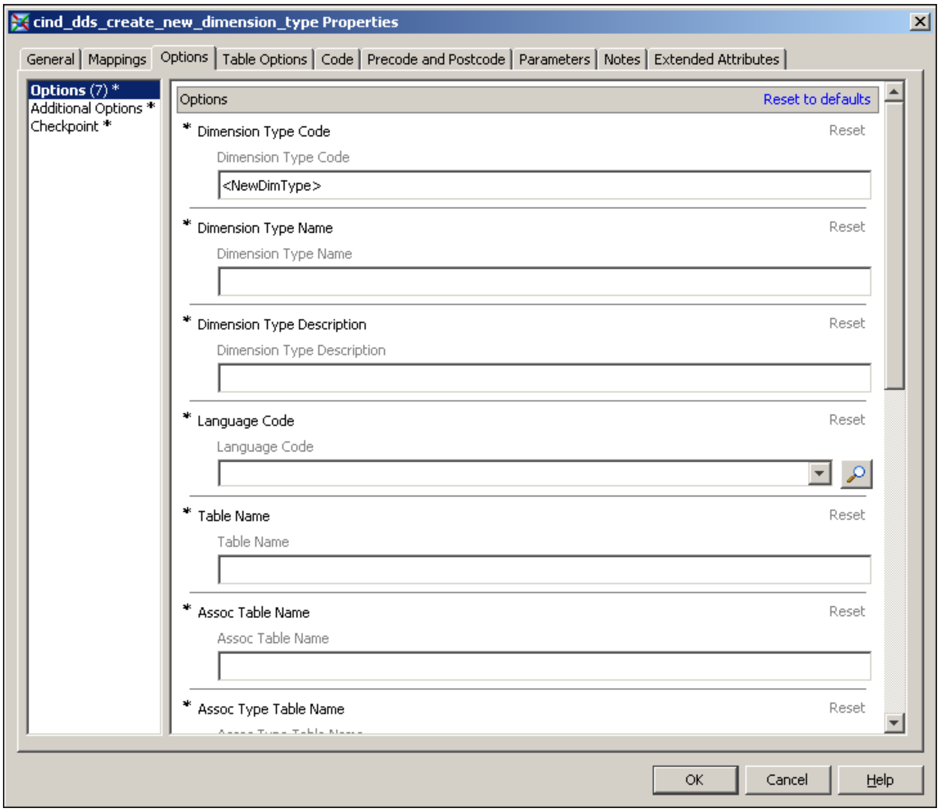
## Run the Job That Creates a New Dimension Type in the Staging Tables

The cind_dds_create_a_new_dimension_type_in_dds_stagedds_tables job does the following:

- It places a row that describes a specified new dimension type in the STAGE_SOURCE_DIMENSION_TYPE table. The physical table name is SOURCE_DIMENSION_TYPE and the metadata name is STAGE_SOURCE_DIMENSION_TYPE.
- It creates the four detail data store tables and the four corresponding staging tables that will be used to load members and hierarchies into a dimension that belongs to the new dimension type. For information about loading members and hierarchies into a dimension, see "Loading Members and Hierarchies into a Dimension" on page 193.

Before you run this job, set its options as follows:

1. On the **Folders** tab, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder.
2. Double-click the cind_dds_create_a_new_dimension_type_in_dds_stagedds_tables job. The process diagram for the job appears on the right.
3. In the process diagram, right-click the cind_dds_create_new_dimension_type transformation and select **Properties** from the pop-up menu.
4. In the properties window, select the **Options** tab:



<NewDimType>

5. On the **Options** tab, provide values for the following options:

   - **Dimension Type Code** is the code of the new dimension type. This code can be up to 32 characters long, and it can include special characters. (A backslash is treated as an escape character.) If you use the naming convention of the predefined dimension types, this selection embeds the dimension type code in certain table and column names. In this instance, the code must be 16 characters or less and it cannot include special characters.
   - **Dimension Type Name** is the name of the new dimension type.
   - **Dimension Type Description** is the description of the new dimension type.
   - **Language Code** is one of the language codes in the CODE_LANGUAGE table. Select the appropriate language code for the dimension name and description that you have provided. For information about loading language codes, see "Loading Language Codes and Data Locale Codes" on page 181.
   - **Table Name** is the name of the primary member table for the new dimension type. To use the naming convention of the predefined dimension types, make this table name identical to the dimension type code.
   - **Assoc Table Name** is the name of the hierarchy structure table for the new dimension type. To use the naming convention of the predefined dimension types, make this table name identical to *code*_ASSOC, where *code* is the dimension type code. The table name must be 32 characters or less, and it cannot contain special characters.
   - **Assoc Type Table Name** is the name of the hierarchy identification table for the new dimension type. To use the naming convention of the predefined dimension types, make this table name identical to *code*_ASSOC_TYPE, where *code* is the dimension type code. The table name must be 32 characters or less, and it cannot contain special characters.

- **NLS Table Name** is the name of the secondary member table for the new dimension type. To use the naming convention of the predefined dimension types, make this table name identical to *code*_NLS, where *code* is the dimension type code. The table name must be 32 characters or less, and it cannot contain special characters.
- **Business ID Column Name** is the name of the column that will be added to the STAGE_GL_TRANSACTION_SUM, STAGE_GL_JRNL_DETAILS, , STAGE_MISC_RATE, STAGE_CURRENCY_COMPLEX_EXCH_RATE, and STAGE_SASOP_DETAIL staging tables to hold member codes that belong to the new dimension type. To use the naming convention of the predefined dimension types, make this table name identical to *code*_ID, where *code* is the dimension type code. The column name must be 32 characters or less, and it cannot contain special characters.

  If you select **No** for the **Add Dimension Type to Fact Tables** option, then leave this option blank.
- **Base Fact Column Name** is the name of the column that will be added to the GL_TRANSACTION_SUM, GL_JRNL_DETAILS, MISC_RATE, CURRENCY_ COMPLEX_EXCH_RATE, and SASOP_DETAIL tables to hold the retained keys that are generated from member codes in the staging tables. To use the naming convention of the predefined dimension types, make this table name identical to *code*_RK, where *code* is the dimension type code. The column name must be 32 characters or less, and it cannot contain special characters.

  If you select **No** for the **Add Dimension Type to Fact Tables** option, then leave this option blank.
- **Source Location** is the physical location on the data-tier server of the STAGE_SOURCE_DIMENSION_TYPE table. The physical table name is SOURCE_DIMENSION_TYPE and the metadata name is STAGE_SOURCE_DIMENSION_TYPE.

  If the STAGE_SOURCE_DIMENSION_TYPE table already exists, then the job adds a row of data to it. If this table does not exist, then the job creates the table and places a row of data in it. If you create more than one dimension type, you should always specify the same physical location here, so that all the source data is kept in a single source table.

  The default location STAGEDDS is a good choice. You can specify any alternative location that you have defined.
- **Source Tree** is the location in the tree of folders on the **Folders** tab where the STAGE_SOURCE_DIMENSION_TYPE table is displayed.

  If you select Staging Tables or Source Tables, the STAGE_SOURCE_DIMENSION_TYPE table is displayed in the **Products ⇨ Cross Industry Detail Data Store ⇨ StageDDS** folder.

  The first time you create a dimension type, this option establishes a permanent display location. If you create additional dimension types, the display location for the STAGE_SOURCE_DIMENSION_TYPE table remains the same, and this option is ignored.
- **Add Dimension Type to Fact Tables** is a Yes/No flag. If you select **Yes**, then a column for the new dimension type is added to the GL_TRANSACTION_SUM and GL_JRNL_DETAILS tables and their corresponding staging tables. In this case, you must specify names for these columns using the Base Fact Column Name and Business ID Column Name options. If you select **No**, then no column

is added to these tables. Select **Yes** if and only if the new dimension type will be used to describe financial accounting data for use in SAS Financial Management.

- **Format/Informat for Timestamp Columns** determines the format that is used for time stamps in the four detail data store tables that will hold member and hierarchy data for the new dimension type.

6. Click **OK** to close the Properties window.
7. Select **File** ⇨ **Save**.

The following table contains a set of option values for creating a dimension type that has the code PRODUCT and that uses the naming conventions of the predefined dimensions.

| **Option** | **Value** |
|---|---|
| Dimension Type Code | PRODUCT |
| Dimension Type Name | Product |
| Dimension Type Description | Products and product categories |
| Language Code | en |
| Table Name | PRODUCT |
| Assoc Table Name | PRODUCT_ASSOC |
| Assoc Type Table Name | PRODUCT_ASSOC_TYPE |
| NLS Table Name | PRODUCT_NLS |
| Business ID Column Name | PRODUCT_ID |
| Base Fact Column Name | PRODUCT_RK |
| Source Location | STAGEDDS |
| Source Tree | Staging Tables |
| Add Dimension Type to Fact Table | Yes |

The rest of this chapter uses this set of options.

Run the job and then review the log. Select **View** ⇨ **Refresh** to refresh the metadata so that the tables for the new dimension type appear.

If you selected Staging Tables as the value of the Source Tree option, then you should see the updated STAGE_SOURCE_DIMENSION_TYPE table in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder.

If you specified PRODUCT as the value of the Dimension Type Code option, then you should see the following tables in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder on the **Folders** tab:

- PRODUCT
- PRODUCT_ASSOC

- PRODUCT_ASSOC_TYPE
- PRODUCT_NLS

You should also see the following tables in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder on the **Folders** tab:

- STAGE_PRODUCT
- STAGE_PRODUCT_ASSOC
- STAGE_PRODUCT_ASSOC_TYPE
- STAGE_PRODUCT_NLS

All these tables are also visible on the **Inventory** tab, in the **Table** folder.

## Write Jobs to Load the New Staging Tables

Write jobs to load member and hierarchy data into the staging tables for the new dimension type:

- STAGE_PRODUCT
- STAGE_PRODUCT_ASSOC
- STAGE_PRODUCT_ASSOC_TYPE
- STAGE_PRODUCT_NLS

These jobs are subject to the same requirements as the jobs that load the staging tables for a predefined dimension type. For details, see "Moving Member and Hierarchy Data from Its Source to the Staging Tables" on page 194 .

## Create the Jobs That Load the New Detail Data Store Tables

On the **Folders** tab, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder. Copy the following existing jobs. In the new job names, replace **`new_dimension`** with the code of the dimension type that you are adding, as in this example:

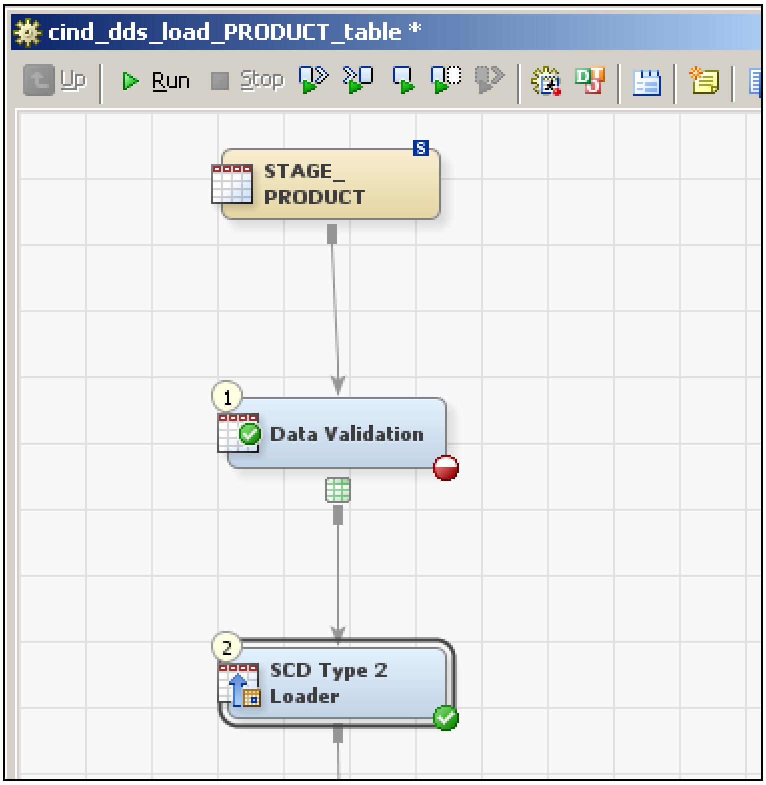
| Existing Jobs | Copied Jobs |
|---|---|
| cind_dds_load_new_dimension_table | cind_dds_load_PRODUCT_table |
| cind_dds_load_new_dimension_assoc_table | cind_dds_load_PRODUCT_assoc_table |
| cind_dds_load_new_dimension_assoc_type_table | cind_dds_load_PRODUCT_assoc_type_table |
| cind_dds_load_new_dimension_nls_table | cind_dds_load_PRODUCT_nls_table |

## Customize the Job That Loads the Detail Data Store Primary Member Table

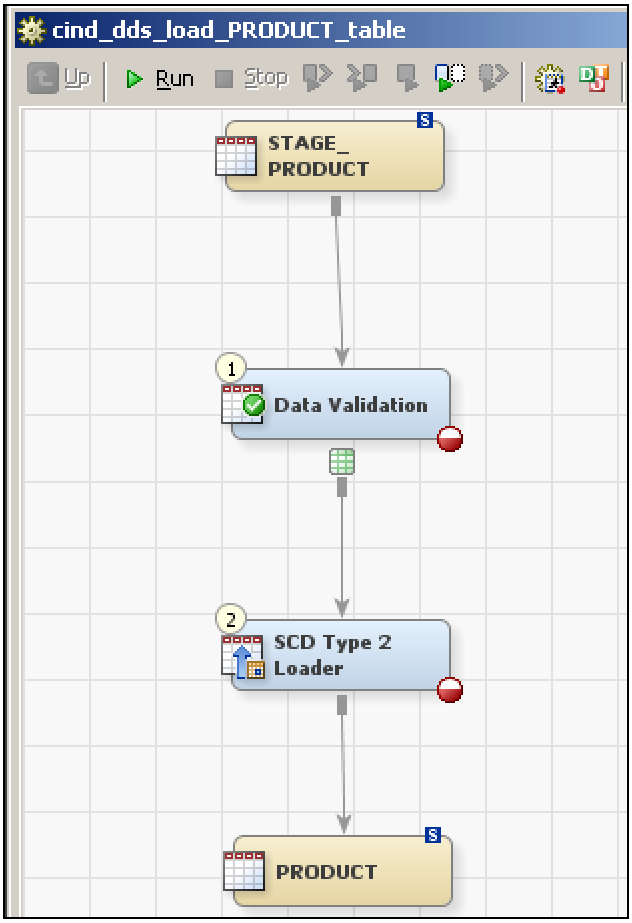
The primary member table is the table whose name consists only of the dimension type code. Each row in this table identifies a member using the detail data store default language.

To customize the job that loads the primary member table:

1. On the **Folders** tab, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder. Double-click the cind_dds_load_PRODUCT_table job to open it.
2. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder, drag and drop the STAGE_PRODUCT table onto the process diagram and connect the table to the Data Validation transformation:



3. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder, drag and drop the PRODUCT table onto the process diagram and connect it to the SCD Type 2 Loader node.

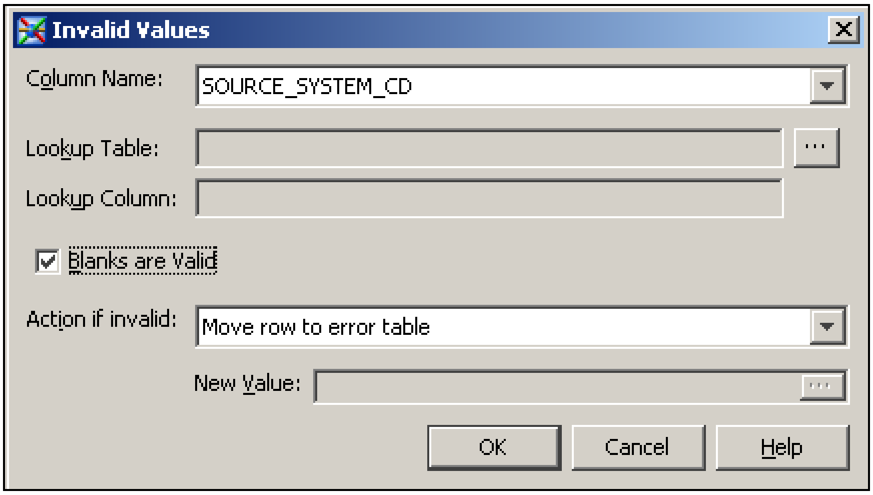


4. In the process diagram, select the Data Validation transformation. Right-click to display the pop-up menu and select **Propagate Columns** ⇨ **To Selected Transformation's Targets** ⇨ **From Sources**.

5. Right-click again to display the pop-up menu and select **Properties**.

6. In the Data Validation Properties window, select the **Invalid Values** tab. Click to display the Invalid Values window.

7. In the Invalid Values window, select SOURCE_SYSTEM_CD in the **Column Name** field and select the **Blanks are Valid** check box:



Invalid Values

Column Name: SOURCE_SYSTEM_CD

Lookup Table:

Lookup Column:

☑ Blanks are Valid

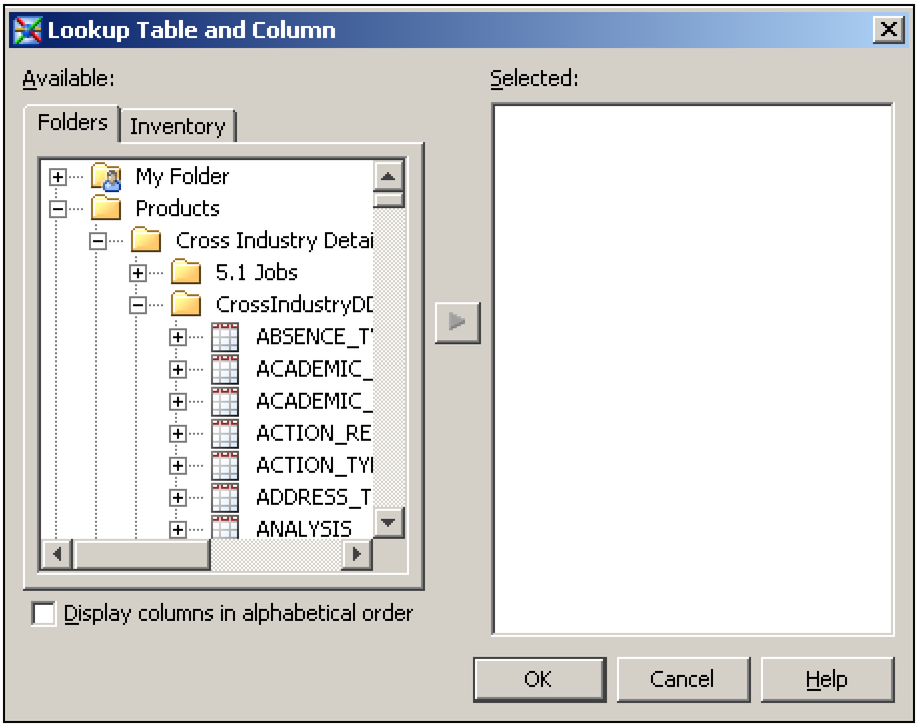
Action if invalid: Move row to error table

New Value:

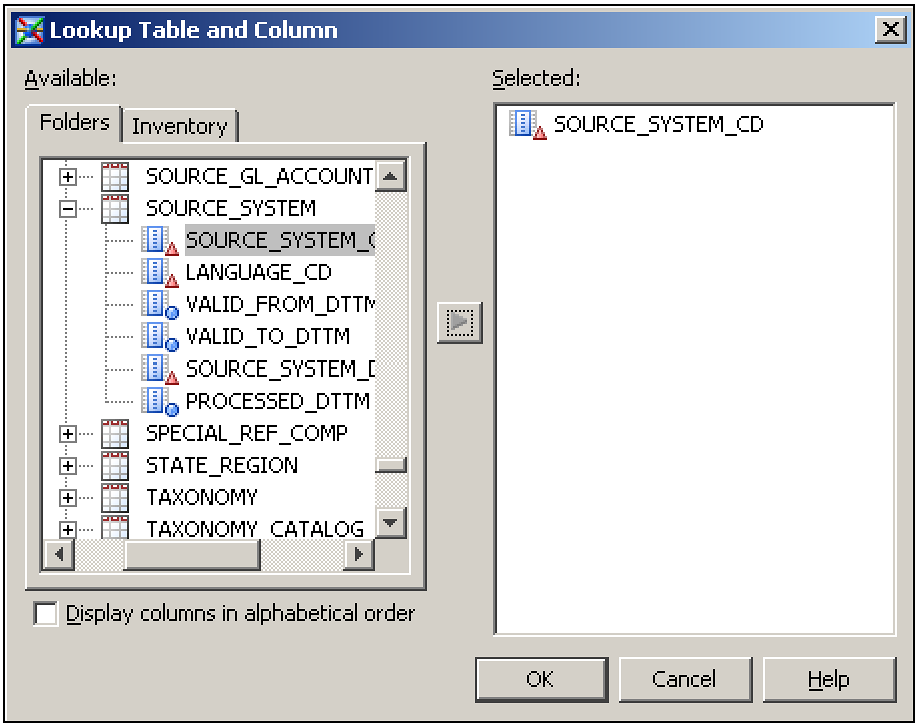
8. Click the button that is next to the **Lookup Table** field.

   The Lookup Table and Column window appears.

9. In the Lookup Table and Column window, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder:

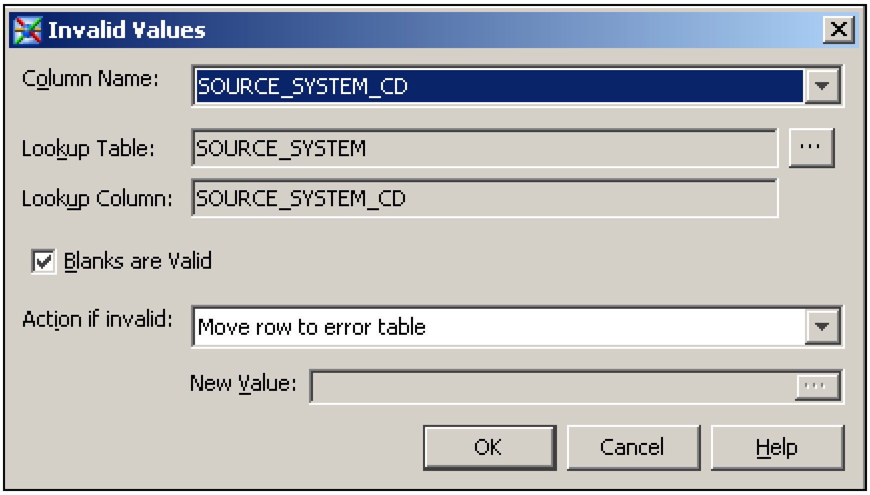


10. Scroll down to the SOURCE_SYSTEM table. Click the + sign next to the SOURCE_SYSTEM table to display the list of columns. Select the SOURCE_SYSTEM_CD column and move it to the **Selected** region:



11. Click **OK** to close the Lookup Table and Column window.

    The Invalid Values window now looks like this:



Column Name: SOURCE_SYSTEM_CD

Lookup Table: SOURCE_SYSTEM

Lookup Column: SOURCE_SYSTEM_CD

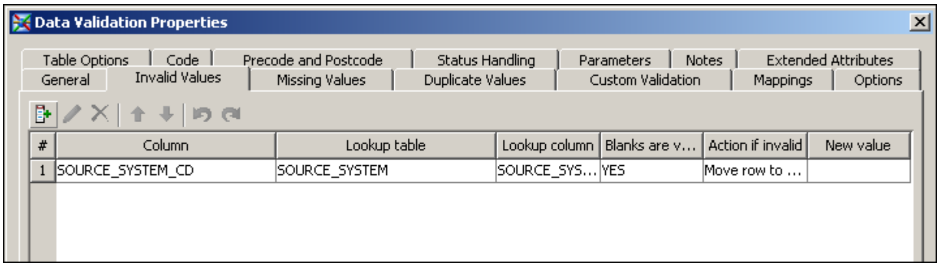
☑ Blanks are Valid

Action if invalid: Move row to error table

New Value:

12. Click **OK** to close the Invalid Values window.

    The **Invalid Values** tab of the Data Validation Properties window now looks like this:

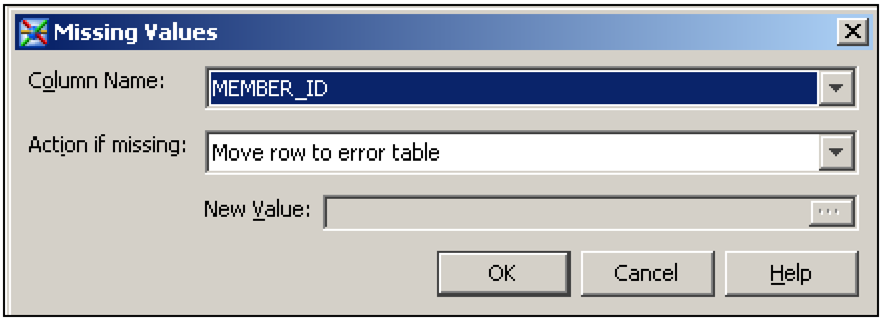


| # | Column | Lookup table | Lookup column | Blanks are v... | Action if invalid | New value |
|---|---|---|---|---|---|---|
| 1 | SOURCE_SYSTEM_CD | SOURCE_SYSTEM | SOURCE_SYS... | YES | Move row to ... | |

13. Select the **Missing Values** tab. Click to display the Missing Values window.

14. Use the Missing Values window to add these checks for missing values:

    - If MEMBER_ID is missing, then move the record to the error table:

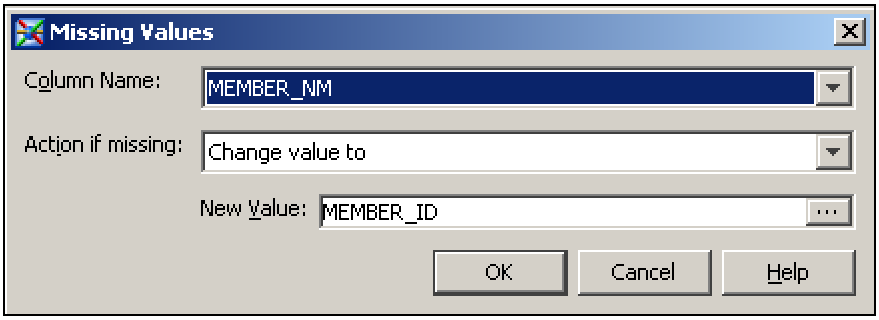


Column Name: MEMBER_ID

Action if missing: Move row to error table

New Value:

- If MEMBER_NM is missing, then give it the same value as MEMBER_ID:



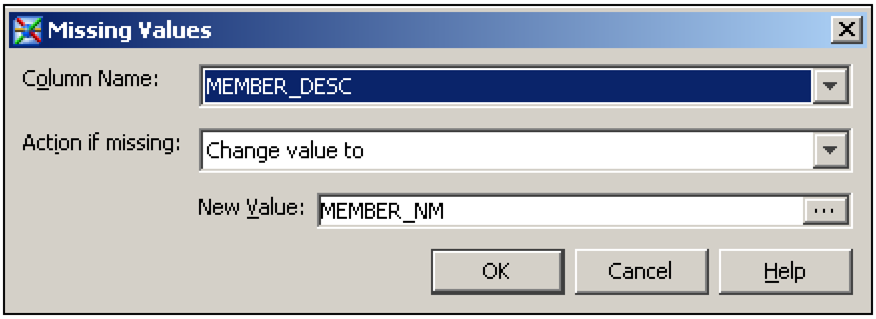
Column Name: MEMBER_NM

Action if missing: Change value to

New Value: MEMBER_ID

You can either type the column name in the **New Value** field or click the button that is next to the **New Value** field and use the Expression Builder window to select the column name. In the Expression Builder window, the column names are available on the **Data Sources** tab.

- If MEMBER_DESC is missing, then give it the same value as MEMBER_NM:



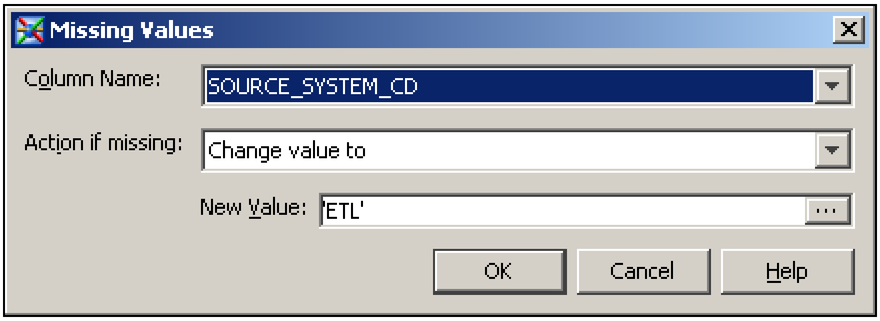
Column Name: MEMBER_DESC

Action if missing: Change value to

New Value: MEMBER_NM

You can either type the column name in the **New Value** field or click the button that is next to the **New Value** field and use the Expression Builder window to select the column name. In the Expression Builder window, the column names are available on the **Data Sources** tab.

- If SOURCE_SYSTEM_CD is missing, then change the value to “ETL”. This string must be in quotation marks and all uppercase:

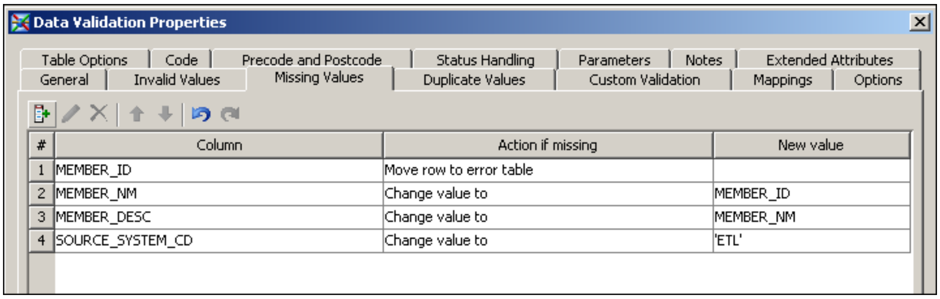


Column Name: SOURCE_SYSTEM_CD

Action if missing: Change value to

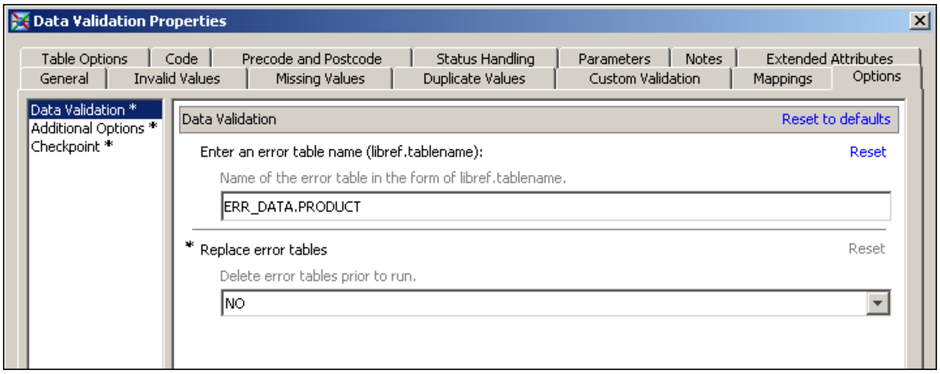
New Value: 'ETL'

These checks for missing values appear as follows on the **Missing Values** tab of the Data Validation Properties window:



| # | Column | Action if missing | New value |
|---|---|---|---|
| 1 | MEMBER_ID | Move row to error table | |
| 2 | MEMBER_NM | Change value to | MEMBER_ID |
| 3 | MEMBER_DESC | Change value to | MEMBER_NM |
| 4 | SOURCE_SYSTEM_CD | Change value to | 'ETL' |

15. Select the **Options** tab. Type ERR_DATA.PRODUCT as the value of the Error Table option:



Enter an error table name (libref.tablename):

Name of the error table in the form of libref.tablename.

ERR_DATA.PRODUCT

* Replace error tables

Delete error tables prior to run.

NO

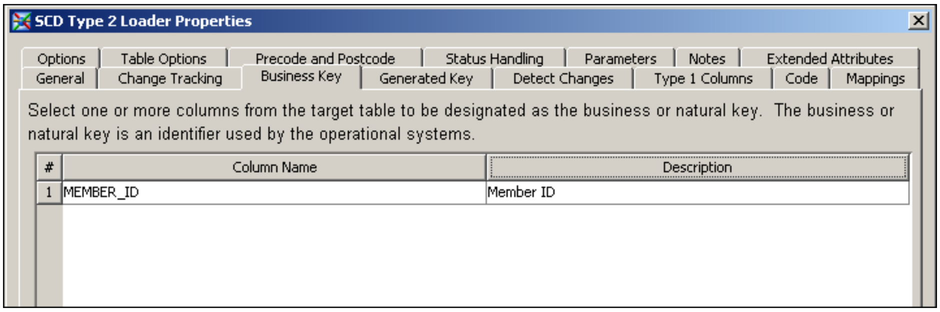
16. Click **OK** to save your changes.

17. In the process diagram, select the SCD Type 2 Loader transformation.

18. Right-click and select **Properties** from the pop-up menu.

19. Select the **Business Key** tab. Click **New** to add the MEMBER_ID column.

The **Business Key** tab of the SCD Type 2 Loader Properties window now looks like this:



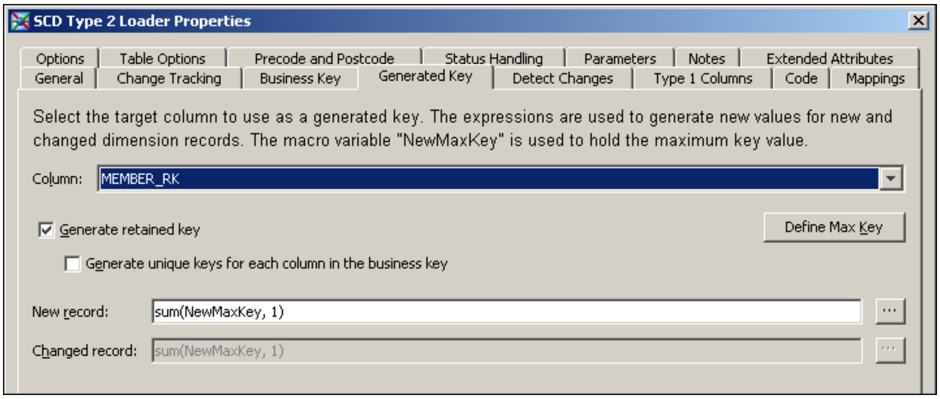
Select one or more columns from the target table to be designated as the business or natural key. The business or natural key is an identifier used by the operational systems.

| # | Column Name | Description |
|---|---|---|
| 1 | MEMBER_ID | Member ID |

20. Select the **Generated Key** tab.

21. Using the drop-down list for the **Column** field, select MEMBER_RK. Also select the **Generate retained key** check box.

The **Generated Key** tab of the SCD Type 2 Loader Properties window now looks like this:



Select the target column to use as a generated key. The expressions are used to generate new values for new and changed dimension records. The macro variable "NewMaxKey" is used to hold the maximum key value.

Column: MEMBER_RK

☑ Generate retained key

□ Generate unique keys for each column in the business key

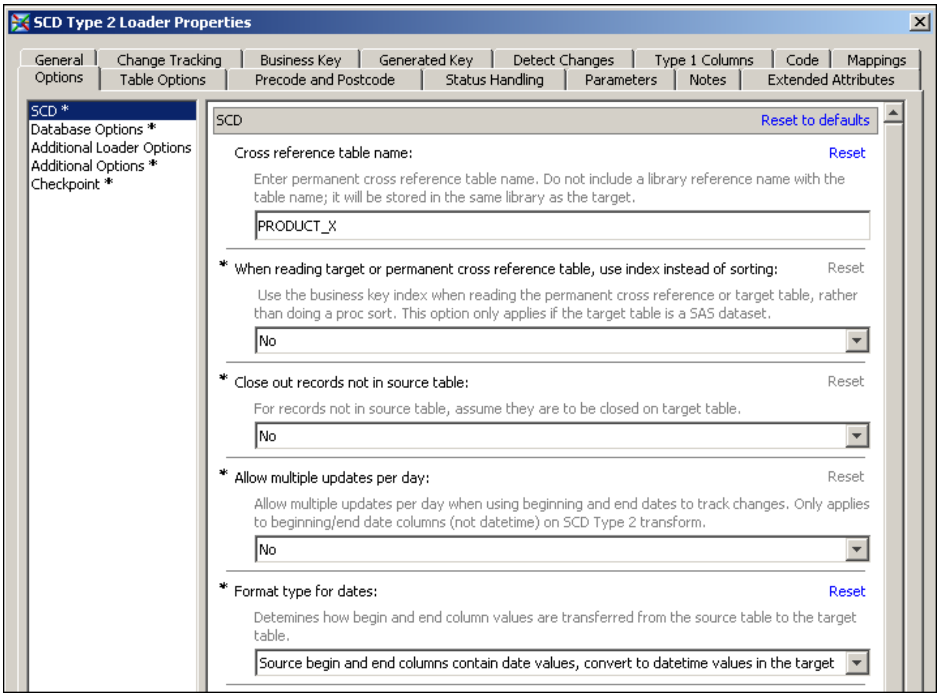
New record: sum(NewMaxKey, 1)

Changed record: sum(NewMaxKey, 1)

22. Select the **Options** tab. Provide values for some of the options as follows:

- **Cross-reference table name** is PRODUCT_X.

- **Format type for dates** depends on the values that you have stored in your STAGE_PRODUCT table. See "Setting a Valid Time Range for Data Records" on page 175.
- Select **Additional loader options**. The value for Load Time Column should contain PROCESSED_DTTM.



Cross reference table name: Reset

Enter permanent cross reference table name. Do not include a library reference name with the table name; it will be stored in the same library as the target.

PRODUCT_X

* When reading target or permanent cross reference table, use index instead of sorting: Reset

Use the business key index when reading the permanent cross reference or target table, rather than doing a proc sort. This option only applies if the target table is a SAS dataset.

No

* Close out records not in source table: Reset

For records not in source table, assume they are to be closed on target table.

No

* Allow multiple updates per day: Reset

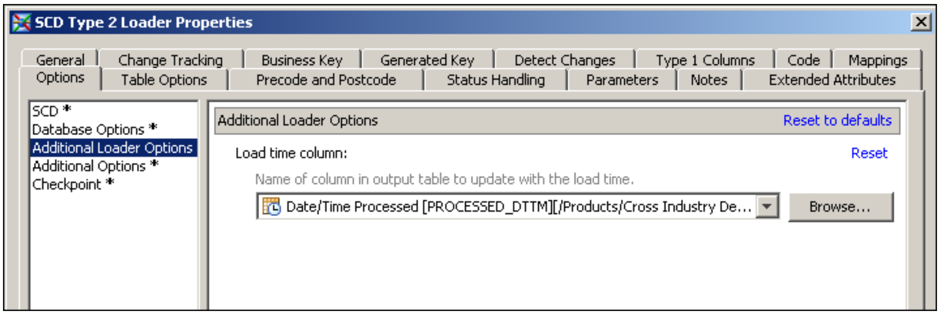
Allow multiple updates per day when using beginning and end dates to track changes. Only applies to beginning/end date columns (not datetime) on SCD Type 2 transform.

No

* Format type for dates: Reset

Detemines how begin and end column values are transferred from the source table to the target table.

Source begin and end columns contain date values, convert to datetime values in the target



Load time column: Reset

Name of column in output table to update with the load time.

Date/Time Processed [PROCESSED_DTTM][/Products/Cross Industry De...

23. Click **OK** to close the SCD Type 2 Loader Properties window.
24. Select **File** ⇨ **Save** to save the contents of the job, and then close the job.

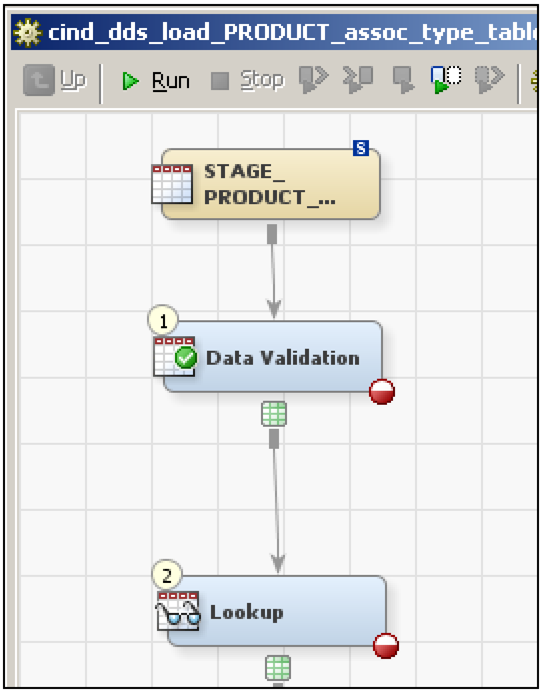
## Customize the Job That Loads the Detail Data Store Hierarchy Identification Table

The hierarchy identification table is the table whose name ends in ASSOC_TYPE. Each row in this table identifies a hierarchy.

To customize the job that loads the hierarchy identification table:

1. On the **Folders** tab, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder. Double-click the cind_dds_load_PRODUCT_assoc_type_table job to open it.

2. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder, drag and drop the STAGE_PRODUCT_ASSOC_TYPE table onto the process diagram, and connect it to the Data Validation transformation:



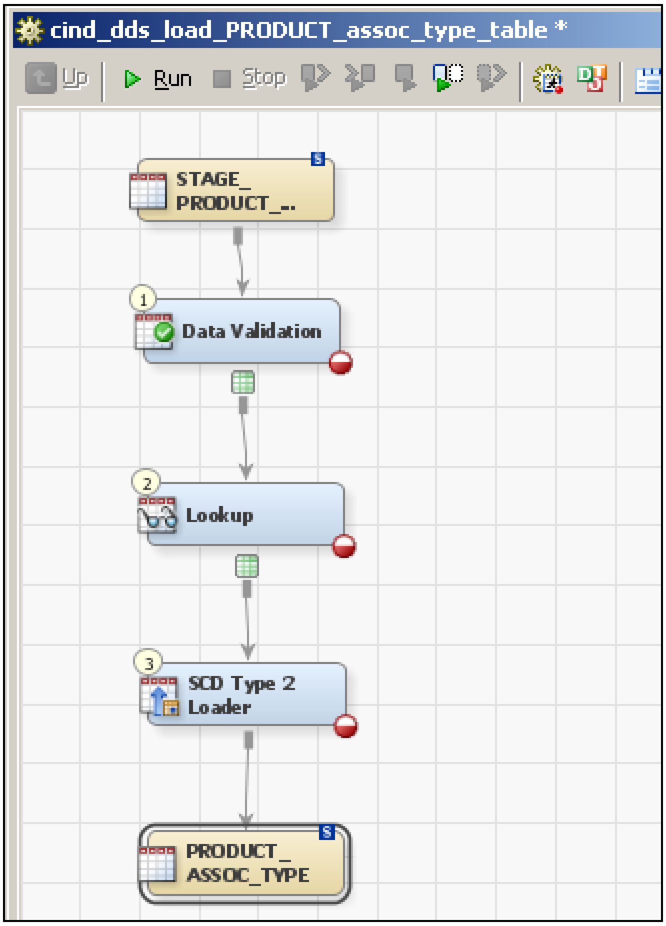
The Lookup transformation must be replaced by a Lookup transformation from the Transformations tab.

3. Select the Lookup transformation in the process diagram.

4. Right-click to display the pop-up menu and select **Delete**.

5. On the **Transformations** tab, drag the Lookup transformation from the Data folder onto the process diagram. Connect the Lookup transformation to the Data Validation and SCD Type 2 Loader transformations.

6. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder, drag and drop the PRODUCT_ASSOC_TYPE table onto the process diagram, and connect it to the SCD Type 2 Loader transformation:

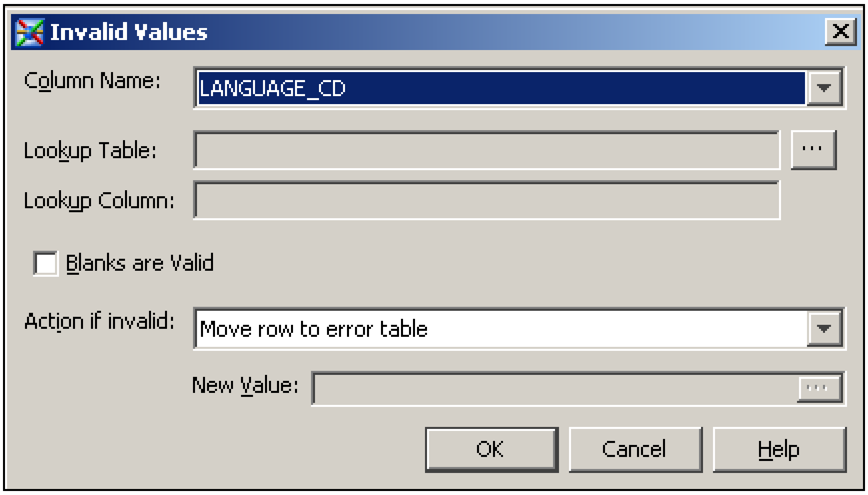


7. In the process diagram, select the Data Validation transformation. Right-click to display the pop-up menu and select **Propagate Columns** ⇨ **To Selected Transformation's Targets** ⇨ **From Sources**.

8. Right-click again to display the pop-up menu and select **Properties**.

9. In the Data Validation Properties window, select the **Invalid Values** tab. Click to display the Invalid Values window.

10. In the Invalid Values window, select LANGUAGE_CD in the **Column Name** field:



Column Name: LANGUAGE_CD

Lookup Table:

Lookup Column:

☐ Blanks are Valid

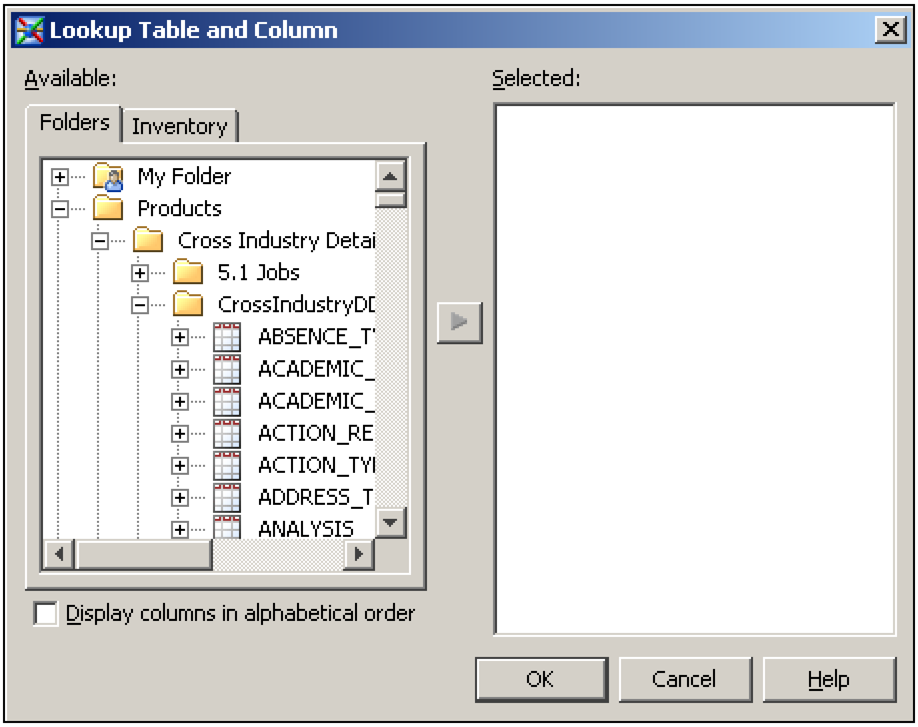
Action if invalid: Move row to error table

New Value:

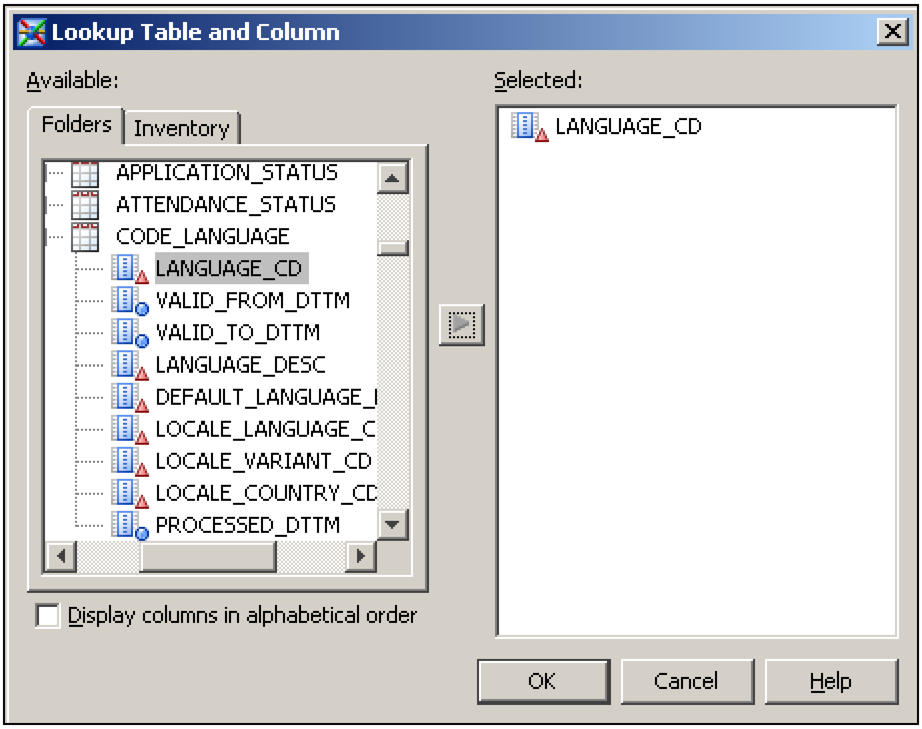
11. Click the button that is next to the **Lookup Table** field.

The Lookup Table and Column window appears.

12. In the Lookup Table and Column window, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder:

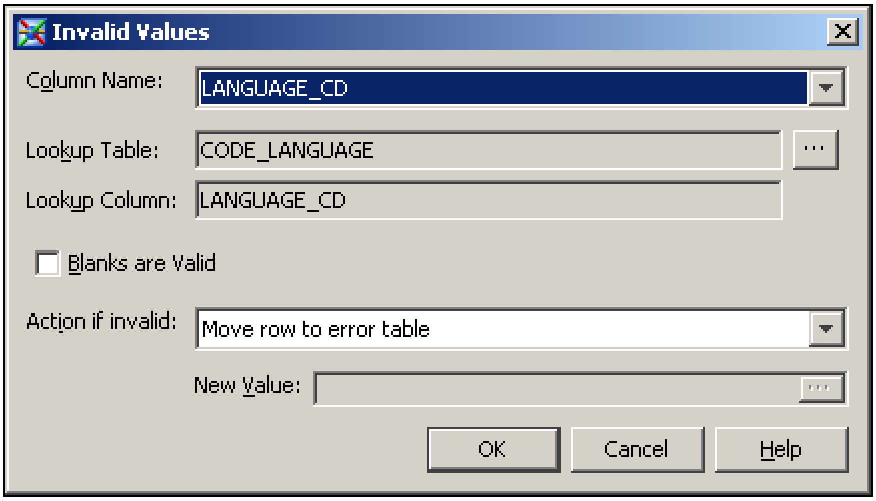


13. Scroll down to the CODE_LANGUAGE table. Click the + sign next to the CODE_LANGUAGE table to expand the list of columns. Select the LANGUAGE_CD column and move it to the **Selected** region:



14. Click **OK** to close the Lookup Table and Column window.

The Invalid Values window now looks like this:



**Invalid Values**

Column Name: LANGUAGE_CD

Lookup Table: CODE_LANGUAGE

Lookup Column: LANGUAGE_CD

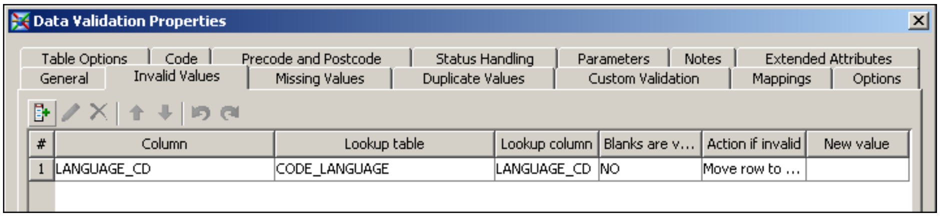
□ Blanks are Valid

Action if invalid: Move row to error table

New Value:

15. Click **OK** to close the Invalid Values window.

The **Invalid Values** tab of the Data Validation Properties window now looks like this:



**Data Validation Properties**

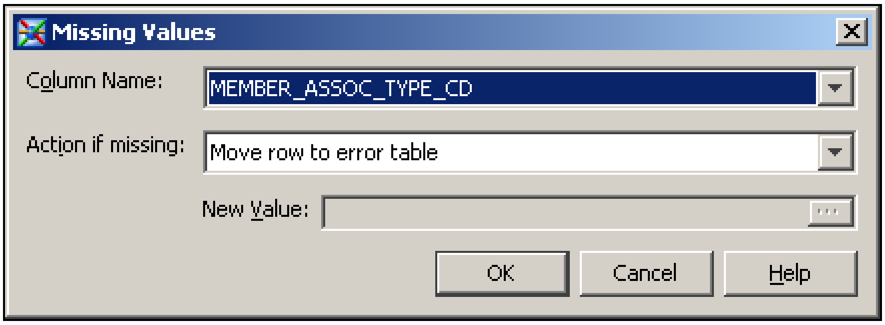
| # | Column | Lookup table | Lookup column | Blanks are v... | Action if invalid | New value |
|---|---|---|---|---|---|---|
| 1 | LANGUAGE_CD | CODE_LANGUAGE | LANGUAGE_CD | NO | Move row to ... | |

16. Select the **Missing Values** tab.

17. Click ⊞ to display the Missing Values window.

18. Use the Missing Values window to add these checks for missing values:

- If MEMBER_ASSOC_TYPE_CD is missing, then move the record to the error table:



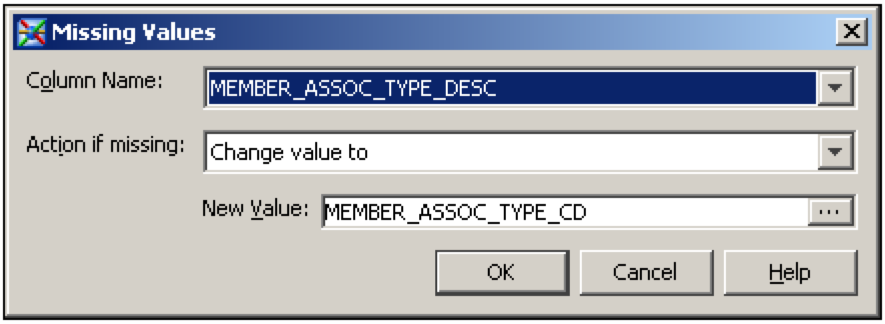
**Missing Values**

Column Name: MEMBER_ASSOC_TYPE_CD

Action if missing: Move row to error table

New Value:

- If MEMBER_ASSOC_TYPE_DESC is missing, then give it the same value as MEMBER_ASSOC_TYPE_CD:

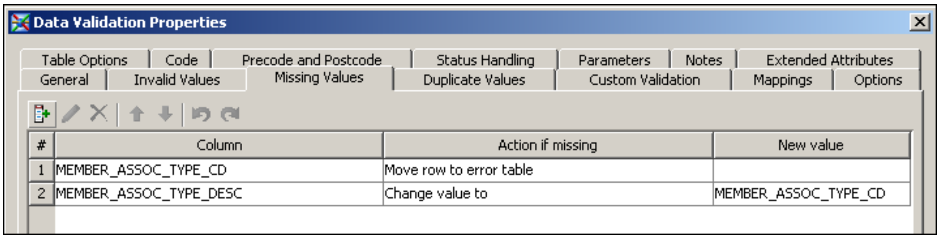


Column Name: MEMBER_ASSOC_TYPE_DESC

Action if missing: Change value to

New Value: MEMBER_ASSOC_TYPE_CD

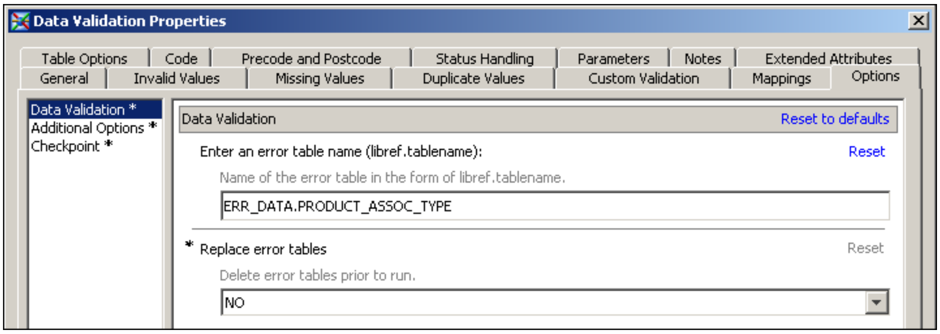
These checks for missing values appear as follows on the **Missing Values** tab of the Data Validation Properties window:



| # | Column | Action if missing | New value |
|---|---|---|---|
| 1 | MEMBER_ASSOC_TYPE_CD | Move row to error table | |
| 2 | MEMBER_ASSOC_TYPE_DESC | Change value to | MEMBER_ASSOC_TYPE_CD |

19. Select the **Options** Tab.

20. Type ERR_DATA.PRODUCT_ASSOC_TYPE as the value of the **Enter an error table name** option:



Enter an error table name (libref.tablename):

Name of the error table in the form of libref.tablename.

ERR_DATA.PRODUCT_ASSOC_TYPE

* Replace error tables

Delete error tables prior to run.

NO

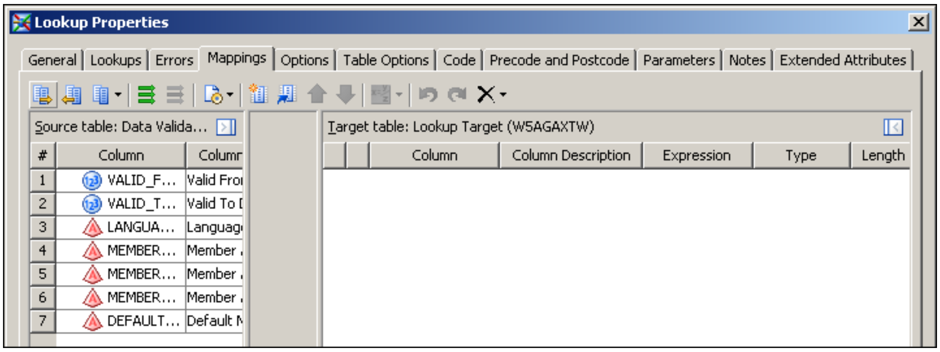
Click **OK** to save your changes.

21. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder, drag and drop the PRODUCT table onto the process diagram and connect it to the Lookup node.

22. In the process diagram, select the Lookup transformation. Right-click and select **Properties** from the pop-up menu.

23. Select the **Errors** tab. Select the **Create error table** check box and the **Create exception table** check box.

24. Select the **Mappings** tab:


Lookup Properties
General | Lookups | Errors | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes

Source table: Data Valida...

| # | Column | Column |
|---|---|---|
| 1 | VALID_F... | Valid Fro |
| 2 | VALID_T... | Valid To [ |
| 3 | LANGUA... | Languag |
| 4 | MEMBER... | Member . |
| 5 | MEMBER... | Member . |
| 6 | MEMBER... | Member . |
| 7 | DEFAULT... | Default N |

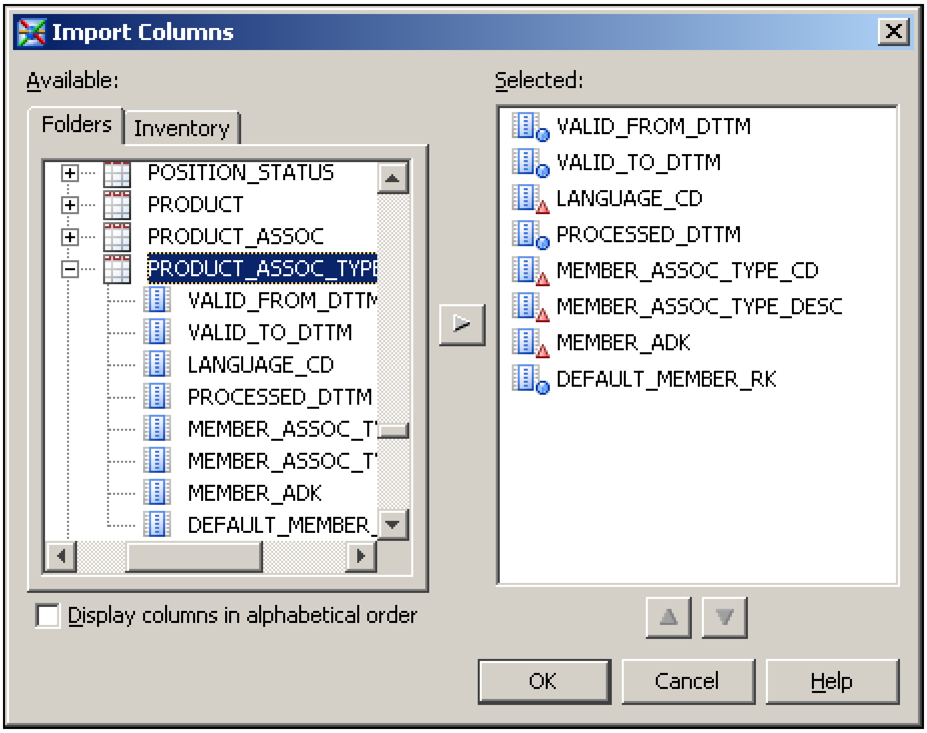
Target table: Lookup Target (W5AGAXTW)

| | | Column | Column Description | Expression | Type | Length |
|---|---|---|---|---|---|---|

In the **Target table** region, right-click to display the pop-up menu and select **Import Columns**.

The Import Columns window appears.

25. In the **Available** region of the Import Columns window, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder. Select the PRODUCT_ASSOC_TYPE table, and then click the right arrow button to move all its columns to the **Selected** region:



Available:

- POSITION_STATUS
- PRODUCT
- PRODUCT_ASSOC
- PRODUCT_ASSOC_TYPE
  - VALID_FROM_DTTM
  - VALID_TO_DTTM
  - LANGUAGE_CD
  - PROCESSED_DTTM
  - MEMBER_ASSOC_T
  - MEMBER_ASSOC_T
  - MEMBER_ADK
  - DEFAULT_MEMBER_

Selected:

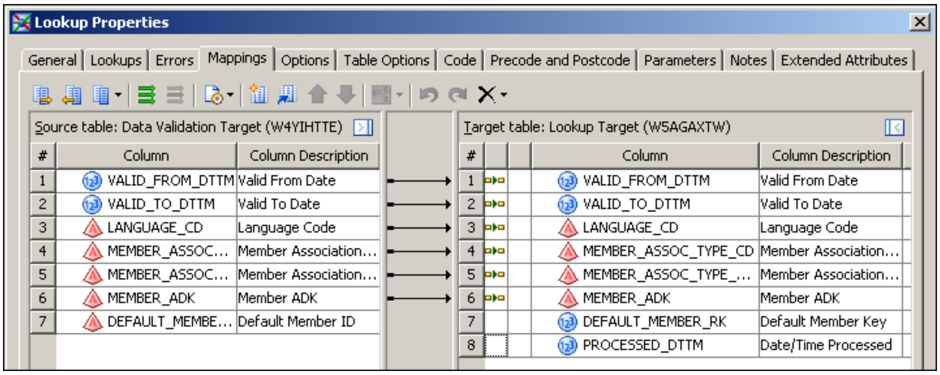
- VALID_FROM_DTTM
- VALID_TO_DTTM
- LANGUAGE_CD
- PROCESSED_DTTM
- MEMBER_ASSOC_TYPE_CD
- MEMBER_ASSOC_TYPE_DESC
- MEMBER_ADK
- DEFAULT_MEMBER_RK

☐ Display columns in alphabetical order

26. Click **OK** to save your changes and close the Import Columns window.

27. In the **Target table** region of the **Mappings** tab, right-click to display the pop-up menu and select **Map All**.

The **Mappings** tab now looks like this:



Source table: Data Validation Target (W4YIHTTE)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | LANGUAGE_CD | Language Code |
| 4 | MEMBER_ASSOC… | Member Association… |
| 5 | MEMBER_ASSOC… | Member Association… |
| 6 | MEMBER_ADK | Member ADK |
| 7 | DEFAULT_MEMBE… | Default Member ID |

Target table: Lookup Target (W5AGAXTW)

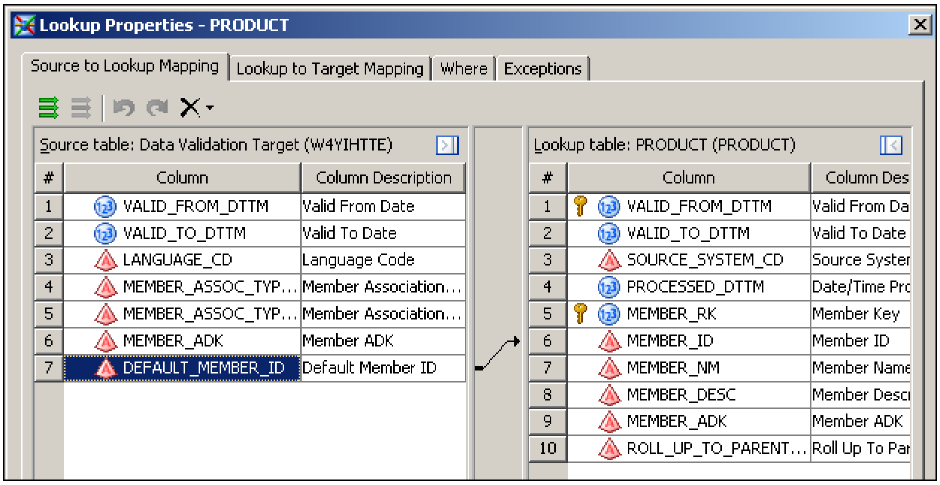
| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | LANGUAGE_CD | Language Code |
| 4 | MEMBER_ASSOC_TYPE_CD | Member Association… |
| 5 | MEMBER_ASSOC_TYPE_… | Member Association… |
| 6 | MEMBER_ADK | Member ADK |
| 7 | DEFAULT_MEMBER_RK | Default Member Key |
| 8 | PROCESSED_DTTM | Date/Time Processed |

28. Select the **Lookups** tab.

29. With the PRODUCT table selected, click **Lookup Properties**.

An inner Lookup Properties window appears.

30. On the **Source to Lookup Mapping** tab of the inner Lookup Properties window, drag the DEFAULT_MEMBER_ID column in the **Source table** region onto the MEMBER_ID column in the **Lookup table** region:



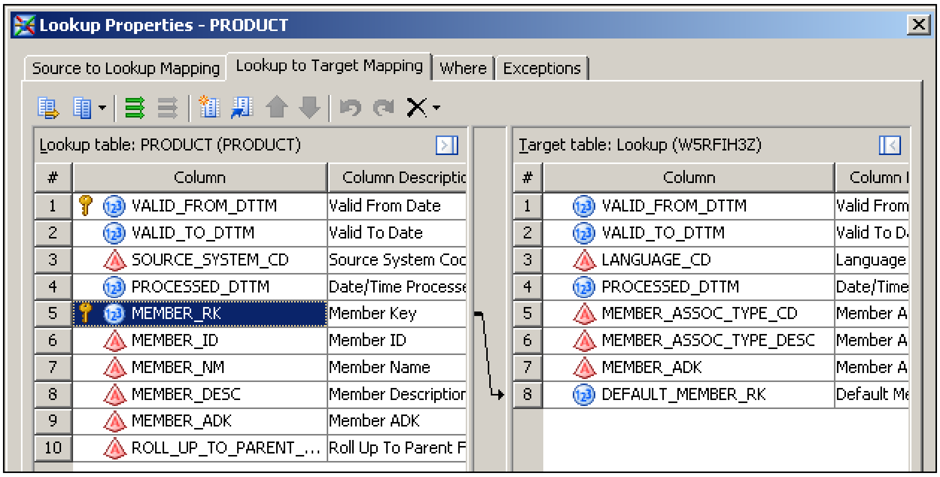
Source table: Data Validation Target (W4YIHTTE)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | LANGUAGE_CD | Language Code |
| 4 | MEMBER_ASSOC_TYP… | Member Association… |
| 5 | MEMBER_ASSOC_TYP… | Member Association… |
| 6 | MEMBER_ADK | Member ADK |
| 7 | DEFAULT_MEMBER_ID | Default Member ID |

Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Des |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Da |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | SOURCE_SYSTEM_CD | Source Syster |
| 4 | PROCESSED_DTTM | Date/Time Prc |
| 5 | MEMBER_RK | Member Key |
| 6 | MEMBER_ID | Member ID |
| 7 | MEMBER_NM | Member Name |
| 8 | MEMBER_DESC | Member Descı |
| 9 | MEMBER_ADK | Member ADK |
| 10 | ROLL_UP_TO_PARENT… | Roll Up To Par |

31. On the **Lookup to Target Mapping** tab of the inner Lookup Properties window, drag the MEMBER_RK column in the **Lookup table** region onto the DEFAULT_MEMBER_RK column in the **Target table** region:



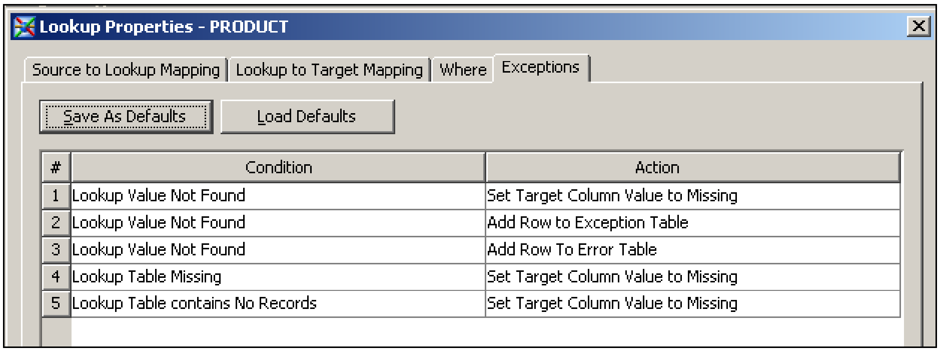
Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Descriptic |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | SOURCE_SYSTEM_CD | Source System Coc |
| 4 | PROCESSED_DTTM | Date/Time Processe |
| 5 | MEMBER_RK | Member Key |
| 6 | MEMBER_ID | Member ID |
| 7 | MEMBER_NM | Member Name |
| 8 | MEMBER_DESC | Member Descriptior |
| 9 | MEMBER_ADK | Member ADK |
| 10 | ROLL_UP_TO_PARENT_… | Roll Up To Parent F |

Target table: Lookup (W5RFIH3Z)

| # | Column | Column I |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From |
| 2 | VALID_TO_DTTM | Valid To D |
| 3 | LANGUAGE_CD | Language |
| 4 | PROCESSED_DTTM | Date/Time |
| 5 | MEMBER_ASSOC_TYPE_CD | Member A |
| 6 | MEMBER_ASSOC_TYPE_DESC | Member A |
| 7 | MEMBER_ADK | Member A |
| 8 | DEFAULT_MEMBER_RK | Default Me |

32. On the **Exceptions** tab of the Lookup Properties window, select the following values in the **Action** column:

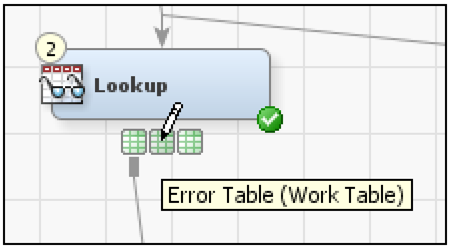


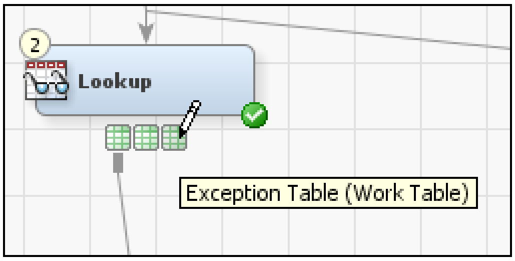
| # | Condition | Action |
|---|---|---|
| 1 | Lookup Value Not Found | Set Target Column Value to Missing |
| 2 | Lookup Value Not Found | Add Row to Exception Table |
| 3 | Lookup Value Not Found | Add Row To Error Table |
| 4 | Lookup Table Missing | Set Target Column Value to Missing |
| 5 | Lookup Table contains No Records | Set Target Column Value to Missing |

33. Click **OK** to close the inner Lookup Properties window.

34. Click **OK** to close the main Lookup Properties window.

35. As a result of the selections that you made on the **Errors** tab of the Lookup Properties window, the process diagram now includes an error table and an exception table:





To conform to the naming conventions of the predefined dimension types, rename the error and exception tables as explained in the following steps.

36. In the process diagram, select the Error Table icon. Right-click and select **Properties** from the pop-up menu.

    On the **General** tab, specify the following metadata name for the error table in the **Name** field: PRODUCT_ASSOC_TYPE_LKUP_ERR.

    On the **Physical Storage** tab, select `Redirect to a registered library` in the **Location** field. Click the ellipsis next to the **Library** field and select **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Error Data** ⇨ **Error Data**. Specify the following name for the physical error table in the **Physical name** field : PRODUCT_ASSOC_TYPE_LKUP_ERR.

    Click **OK**.

37. In the process diagram, select the Exception Table icon. Right-click and select **Properties** from the pop-up menu.

On the **General** tab, specify the following metadata name for the exception table in the **Name** field: PRODUCT_ASSOC_TYPE_LKUP_EXC.

On the **Physical Storage** tab, select `Redirect to a registered library` in the **Location** field. Click the ellipsis next to the **Library** field and select **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Error Data** ⇨ **Error Data**. Specify the following name for the physical error table in the **Physical name** field : PRODUCT_ASSOC_TYPE_LKUP_EXC.

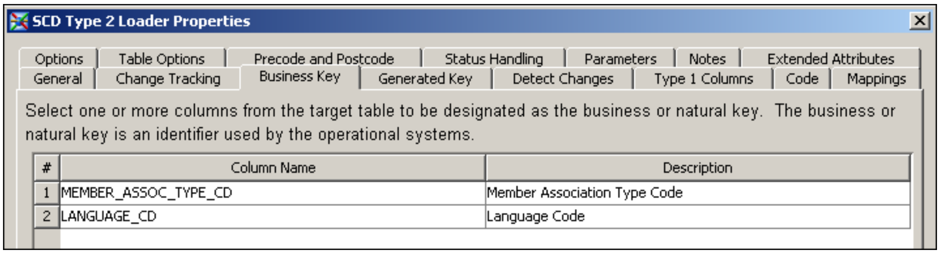
Click **OK**.

38. In the process diagram, select the SCD Type 2 Loader transformation. Right-click and select **Properties** from the pop-up menu.

39. Select the **Business Key** tab. Click **New** to add the following columns in this order:

    - MEMBER_ASSOC_TYPE_CD
    - LANGUAGE_CD

    The **Business Key** tab of the SCD Type 2 Loader Properties window now looks like this:



Select one or more columns from the target table to be designated as the business or natural key. The business or natural key is an identifier used by the operational systems.

| # | Column Name | Description |
|---|---|---|
| 1 | MEMBER_ASSOC_TYPE_CD | Member Association Type Code |
| 2 | LANGUAGE_CD | Language Code |

40. Select the **Options** tab. Provide values for some of the options as follows:

    - Under **SCD**, **Cross reference table name** is PRODUCT_ASSOC_TYPE_X.
    - **Format type for dates** depends on what you load into the STAGE_PRODUCT_ASSOC_TYPE table. For details, see "Setting a Valid Time Range for Data Records" on page 175.
    - Under **Additional Loader Options**, **Load time column** should contain PROCESSED_DTTM.

41. Select the **Mappings** tab.

42. Right-click to display the pop-up menu and select **Map All**. Select and delete the PROCESSED_DTTM-to-PROCESSED_DTTM mapping.

43. Click **OK** to close the SCD Type 2 Loader Properties window.

44. Select **File** ⇨ **Save** to save the contents of the job, and then close the job.

## Customize the Job That Loads the Detail Data Store Hierarchy Structure Table

The hierarchy structure table is the table whose name ends in ASSOC. Each row in this table describes a parent-child relationship between two members in a specific hierarchy. The members must be in the primary member table. The hierarchy must be identified by at least one record in the hierarchy identification table.

1. On the **Folders** tab, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder. Double-click the cind_dds_load_PRODUCT_assoc_table job to open it.

2. The Lookup transformation must be replaced by a Lookup transformation from the Transformations tab. Select the Lookup transformation in the process diagram.

3. Right-click to display the pop-up menu and select **Delete**.

4. On the **Transformations** tab, drag the Lookup transformation from the Data folder onto the process diagram.

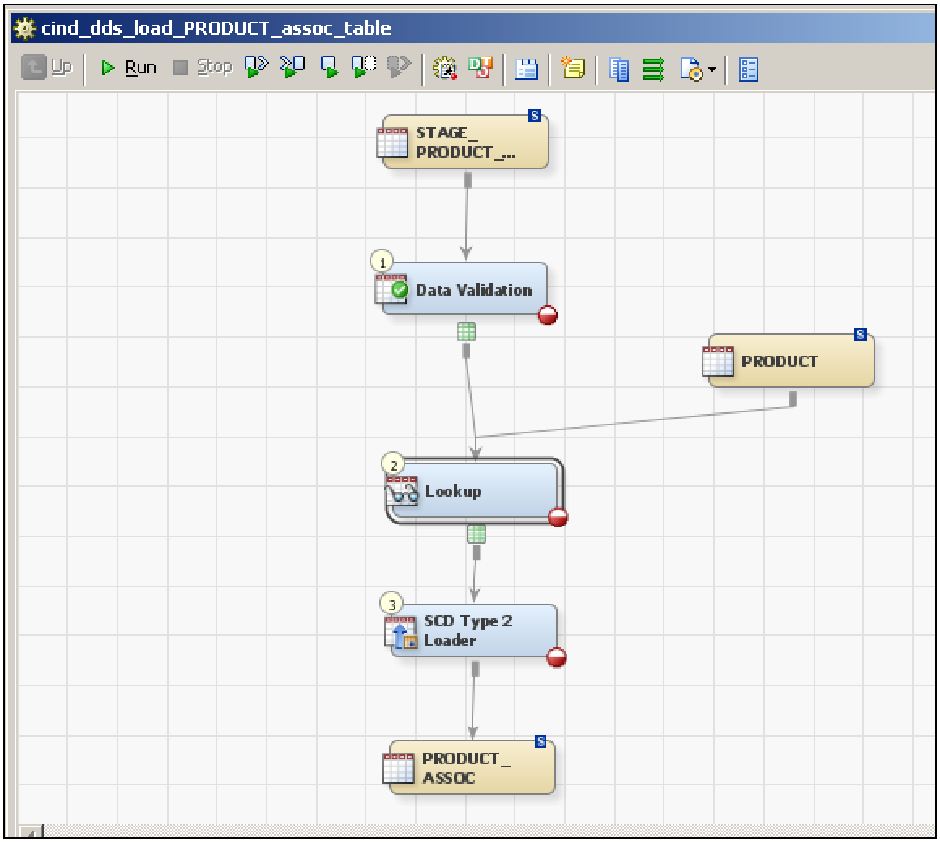
5. Connect the Lookup transformation to the Data Validation and SCD Type 2 Loader transformations.

6. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder, drag and drop the STAGE_PRODUCT_ASSOC table onto the process diagram and connect it to the Data Validation transformation.

7. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder, drag and drop the PRODUCT_ASSOC table onto the process diagram and connect it to the SCD Type 2 Loader transformation.

8. From the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder, drag and drop the PRODUCT table onto the process diagram and connect it to the Lookup transformation.

   With these three tables in place, the process diagram looks like this:

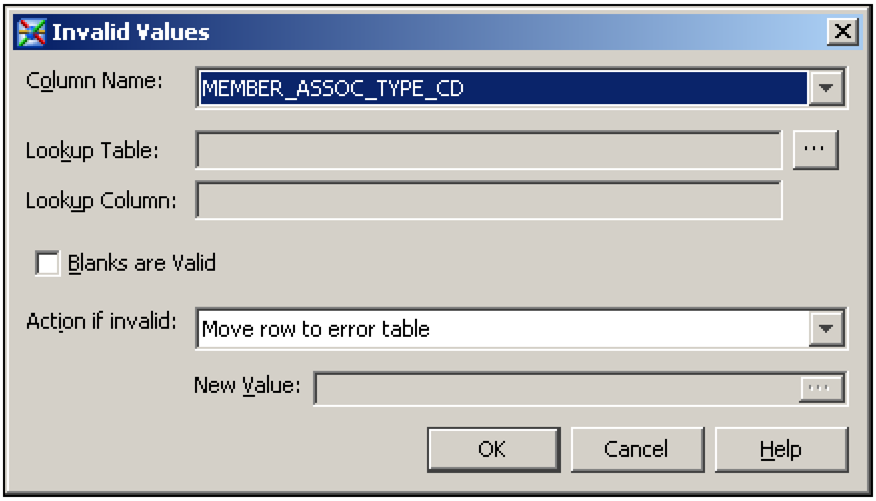


9. In the process diagram, select the Data Validation transformation. Right-click to display the pop-up menu and select **Propagate Columns** ⇨ **To Selected Transformation's Targets** ⇨ **From Sources**.

10. Right-click again to display the pop-up menu and select **Properties**. In the Data Validation Properties window, select the **Invalid Values** tab.

11. Click to display the Invalid Values window.

12. In the Invalid Values window, select MEMBER_ASSOC_TYPE_CD in the **Column Name** field:



Invalid Values

Column Name: MEMBER_ASSOC_TYPE_CD

Lookup Table:

Lookup Column:

☐ Blanks are Valid

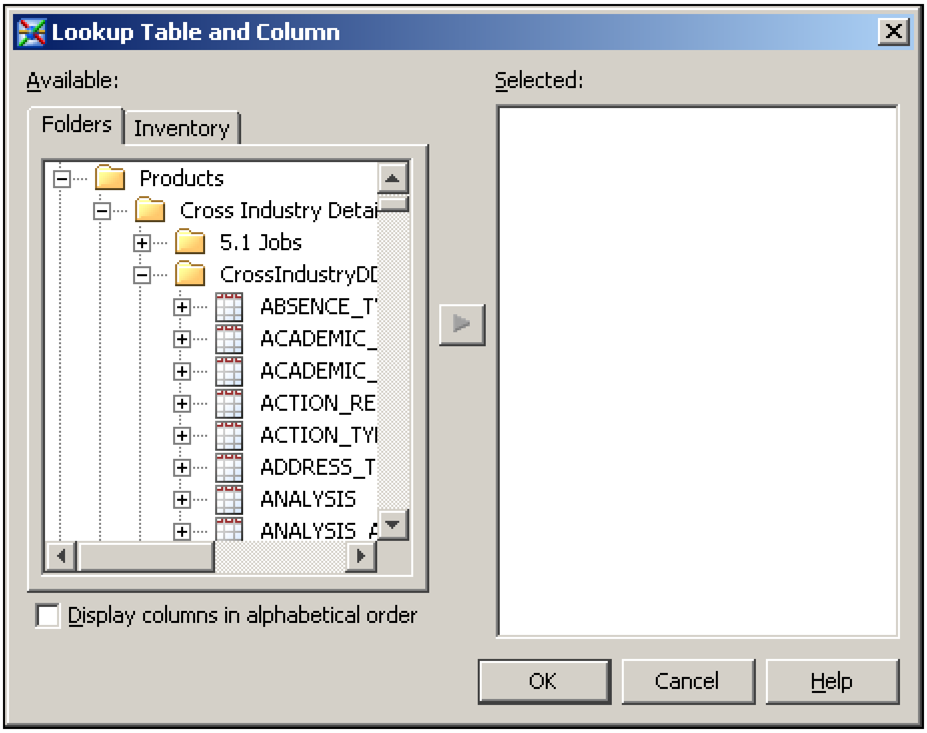
Action if invalid: Move row to error table

New Value:

13. Click the button that is next to the **Lookup Table** field.

    The Lookup Table and Column window appears.

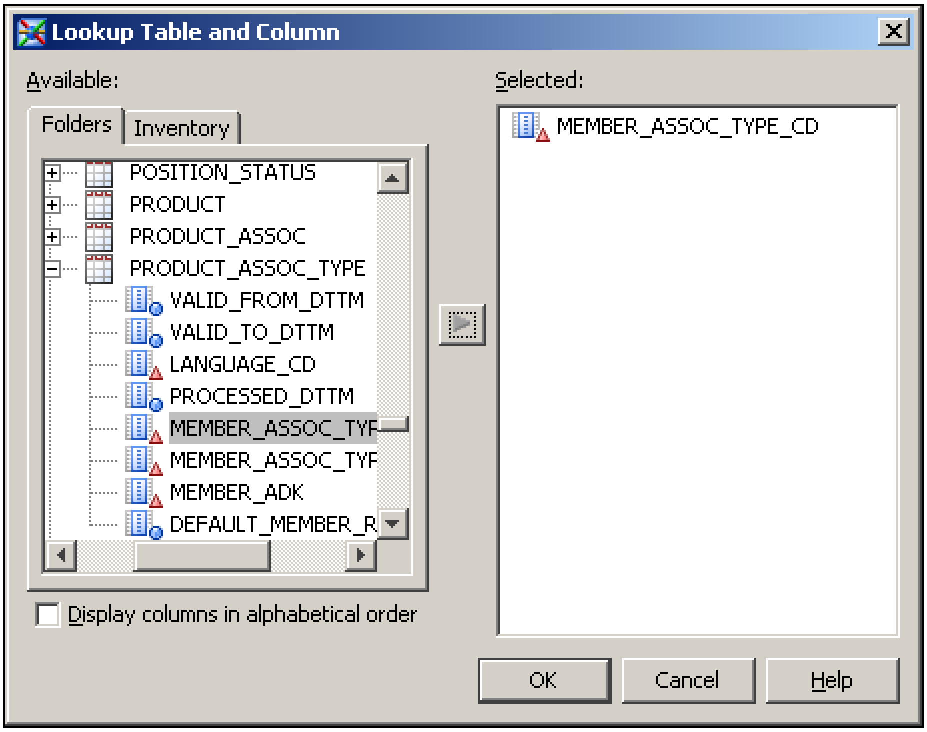
14. In the Lookup Table and Column window, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder:



Lookup Table and Column

☐ Display columns in alphabetical order

15. Scroll down to the PRODUCT_ASSOC_TYPE table. Click the + sign to expand the list of columns. Select the MEMBER_ASSOC_TYPE_CD column and move it to the **Selected** region:



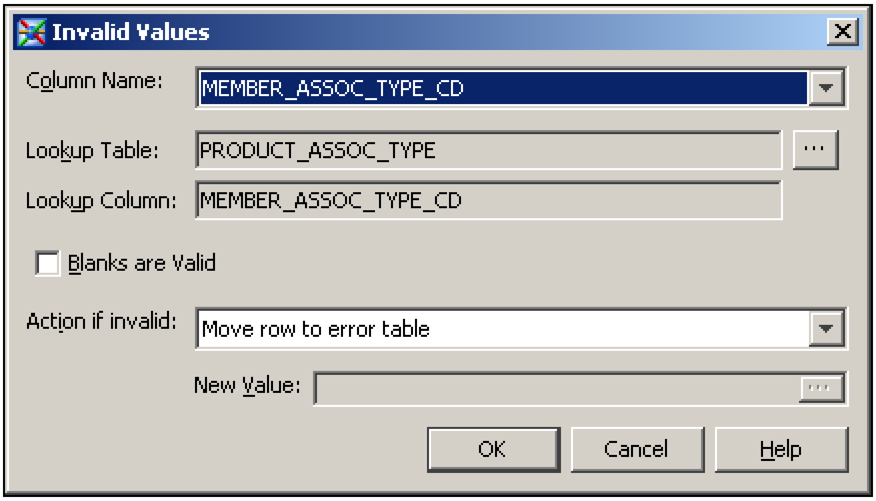
POSITION_STATUS
PRODUCT
PRODUCT_ASSOC
PRODUCT_ASSOC_TYPE
VALID_FROM_DTTM
VALID_TO_DTTM
LANGUAGE_CD
PROCESSED_DTTM
MEMBER_ASSOC_TYP
MEMBER_ASSOC_TYP
MEMBER_ADK
DEFAULT_MEMBER_R

Selected: MEMBER_ASSOC_TYPE_CD

□ Display columns in alphabetical order

16. Click **OK** to close the Lookup Table and Column window.

The Invalid Values window now looks like this:



Column Name: MEMBER_ASSOC_TYPE_CD

Lookup Table: PRODUCT_ASSOC_TYPE

Lookup Column: MEMBER_ASSOC_TYPE_CD

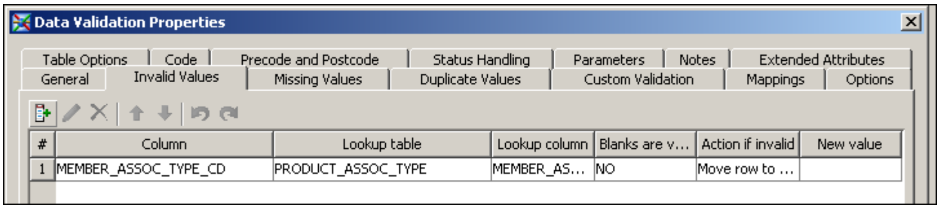
□ Blanks are Valid

Action if invalid: Move row to error table

New Value:

17. Click **OK** to close the Invalid Values window.

The **Invalid Values** tab of the Data Validation Properties window now looks like this:

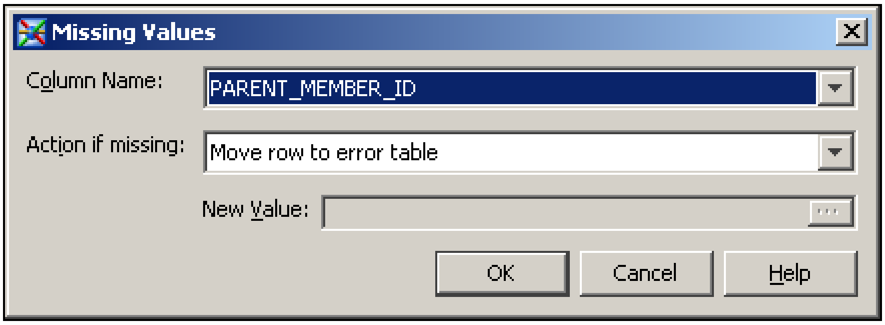


| # | Column | Lookup table | Lookup column | Blanks are v... | Action if invalid | New value |
|---|---|---|---|---|---|---|
| 1 | MEMBER_ASSOC_TYPE_CD | PRODUCT_ASSOC_TYPE | MEMBER_AS... | NO | Move row to ... | |

18. Select the **Missing Values** tab. Click to display the Missing Values window.

19. Use the Missing Values window to add these checks for missing values:

- If PARENT_MEMBER_ID is missing, then move the record to the error table:



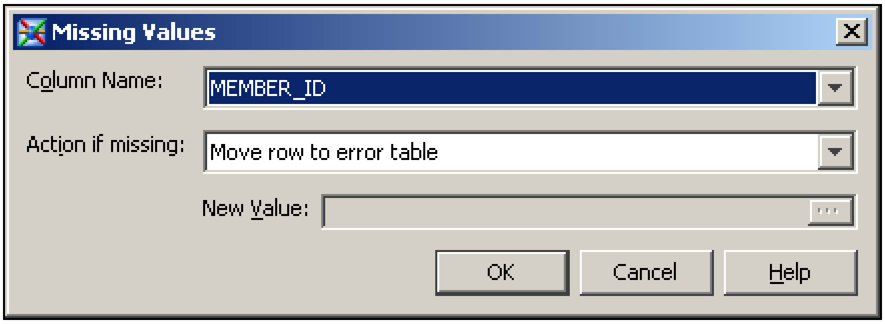
Missing Values

Column Name: PARENT_MEMBER_ID

Action if missing: Move row to error table

New Value:

- If MEMBER_ID is missing, then move the record to the error table:



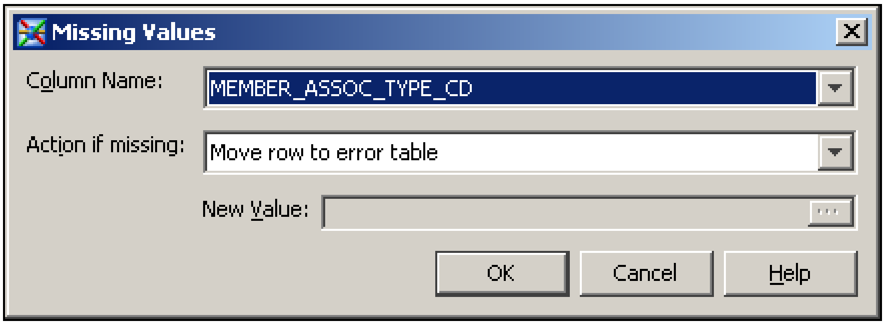
Missing Values

Column Name: MEMBER_ID

Action if missing: Move row to error table

New Value:

- If MEMBER_ASSOC_TYPE_CD is missing, then move the record to the error table:



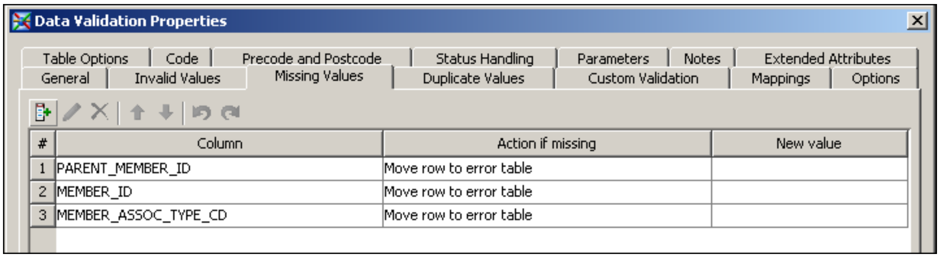
Missing Values

Column Name: MEMBER_ASSOC_TYPE_CD

Action if missing: Move row to error table

New Value:

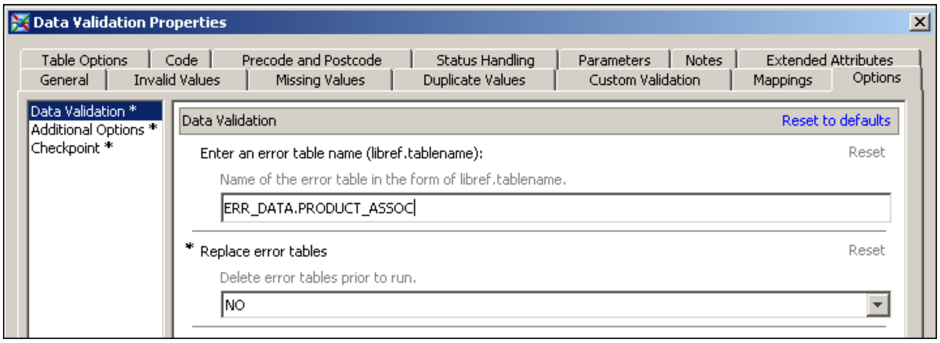
These checks for missing values appear as follows on the **Missing Values** tab of the Data Validation Properties window:



Data Validation Properties

| # | Column | Action if missing | New value |
|---|---|---|---|
| 1 | PARENT_MEMBER_ID | Move row to error table | |
| 2 | MEMBER_ID | Move row to error table | |
| 3 | MEMBER_ASSOC_TYPE_CD | Move row to error table | |

20. Select the **Options** tab. Type ERR_DATA.PRODUCT_ASSOC as the value of the Error Table option:



Enter an error table name (libref.tablename):

Name of the error table in the form of libref.tablename.

ERR_DATA.PRODUCT_ASSOC

* Replace error tables

Delete error tables prior to run.

NO

21. Click **OK** to save your changes.

22. This job must look up RK values for two columns of the STAGE_PRODUCT_ASSOC table (MEMBER_ID and PARENT_MEMBER_ID). Therefore, modifying its Lookup transformation involves several extra steps.

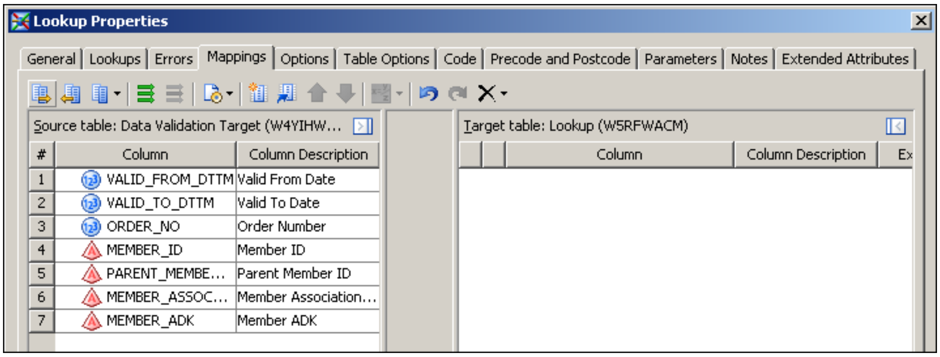
    In the process diagram, select the Lookup transformation. Right-click and select **Ports** ⇨ **Add Input Port** from the pop-up menu.

    Map the PRODUCT table to the new input port for the Lookup transformation. This enables you to define two lookups in the transformation.

23. In the process diagram, select the Lookup transformation. Right-click and select **Properties** from the pop-up menu.

24. Select the **Errors** tab. Select the **Create error table** check box and the **Create exception table** check box.

25. Select the **Mappings** tab:



Source table: Data Validation Target (W4YIHW...

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | MEMBER_ID | Member ID |
| 5 | PARENT_MEMBE... | Parent Member ID |
| 6 | MEMBER_ASSOC... | Member Association... |
| 7 | MEMBER_ADK | Member ADK |

Target table: Lookup (W5RFWACM)

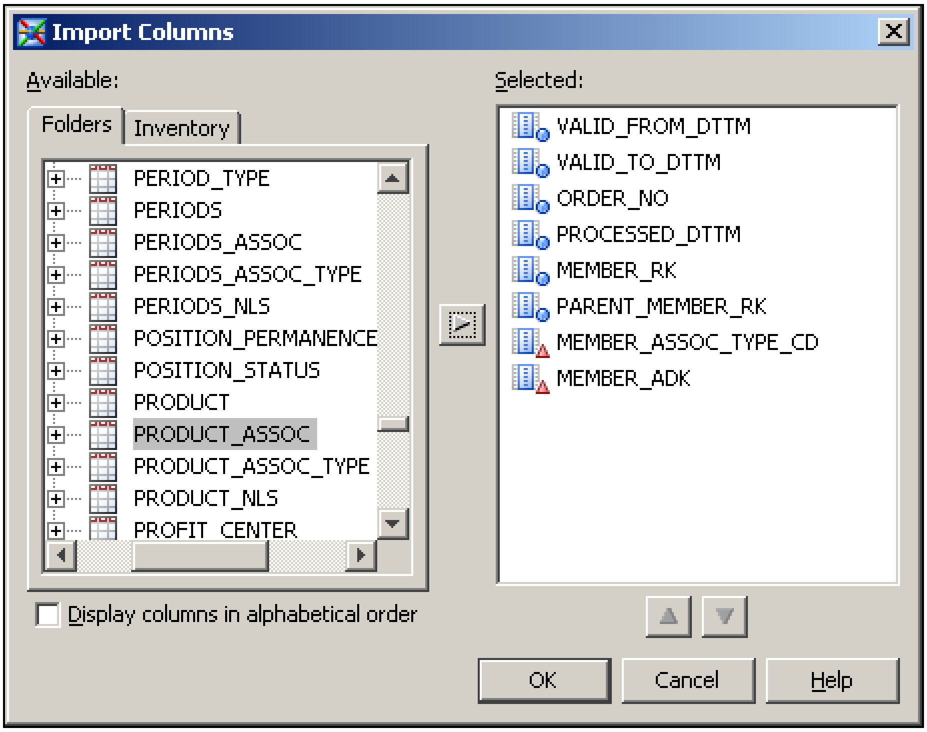
|  |  | Column | Column Description | Ex |
|---|---|---|---|---|

In the **Target table** region, right-click to display the pop-up menu and select **Import Columns**.

The Import Columns window appears.

26. In the **Available Columns** region of the Import Columns window, open the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder.

27. In that folder, select the PRODUCT_ASSOC table, and then click the right arrow button to move all its columns to the **Selected** region:



Available:

- PERIOD_TYPE
- PERIODS
- PERIODS_ASSOC
- PERIODS_ASSOC_TYPE
- PERIODS_NLS
- POSITION_PERMANENCE
- POSITION_STATUS
- PRODUCT
- PRODUCT_ASSOC
- PRODUCT_ASSOC_TYPE
- PRODUCT_NLS
- PROFIT_CENTER

Selected:

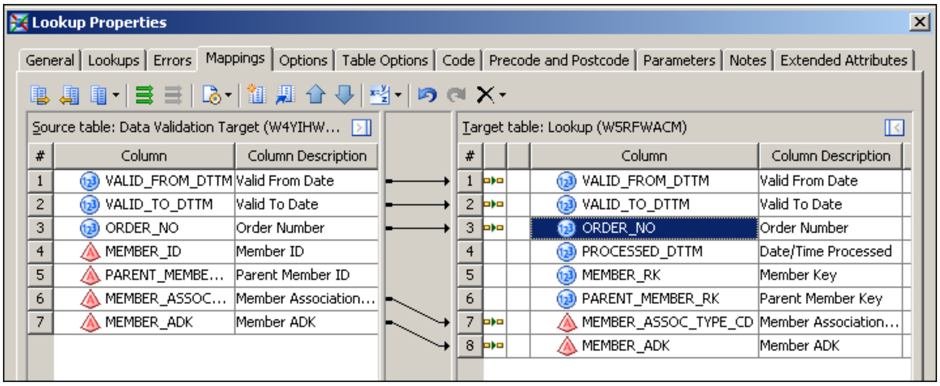
- VALID_FROM_DTTM
- VALID_TO_DTTM
- ORDER_NO
- PROCESSED_DTTM
- MEMBER_RK
- PARENT_MEMBER_RK
- MEMBER_ASSOC_TYPE_CD
- MEMBER_ADK

□ Display columns in alphabetical order

28. Click **OK** to save your changes and close the Import Columns window.

29. In the **Target table** region of the **Mapping** tab, right-click to display the pop-up menu and select **Map All**.

    The **Mapping** tab now looks like this:



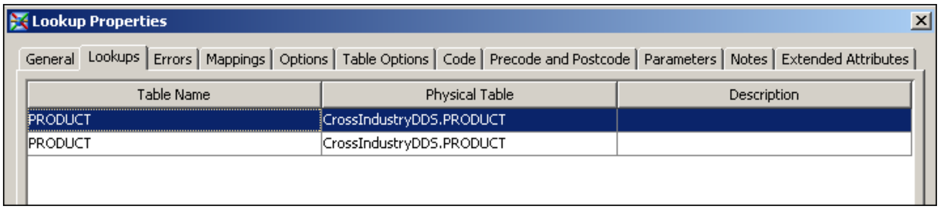
Source table: Data Validation Target (W4YIHW...

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | MEMBER_ID | Member ID |
| 5 | PARENT_MEMBE... | Parent Member ID |
| 6 | MEMBER_ASSOC... | Member Association... |
| 7 | MEMBER_ADK | Member ADK |

Target table: Lookup (W5RFWACM)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | PROCESSED_DTTM | Date/Time Processed |
| 5 | MEMBER_RK | Member Key |
| 6 | PARENT_MEMBER_RK | Parent Member Key |
| 7 | MEMBER_ASSOC_TYPE_CD | Member Association... |
| 8 | MEMBER_ADK | Member ADK |

30. Select the **Lookups** tab:

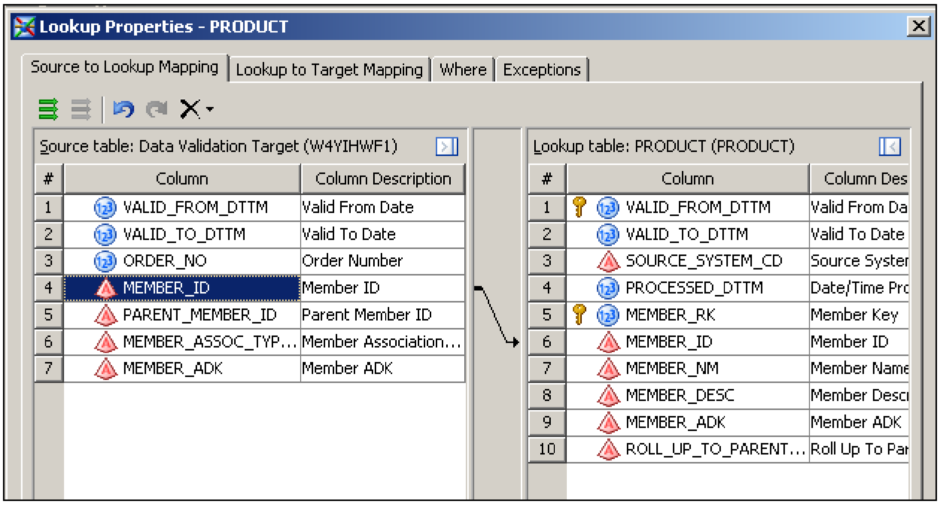


| Table Name | Physical Table | Description |
|---|---|---|
| PRODUCT | CrossIndustryDDS.PRODUCT | |
| PRODUCT | CrossIndustryDDS.PRODUCT | |

31. On the **Lookups** tab, with the first instance of the PRODUCT table selected as shown above, click **Lookup Properties**.

    An inner Lookup Properties window appears.

32. On the **Source to Lookup Mapping** tab of the inner Lookup Properties window, drag the MEMBER_ID column in the **Source table** region onto the MEMBER_ID column in the **Lookup table** region.



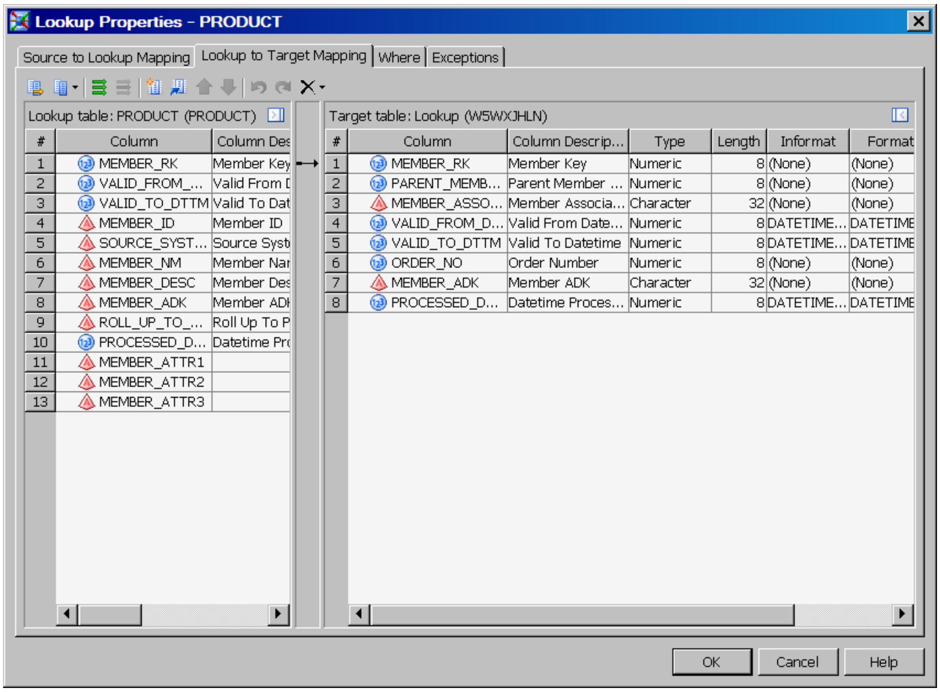
Source table: Data Validation Target (W4YIHWF1)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | MEMBER_ID | Member ID |
| 5 | PARENT_MEMBER_ID | Parent Member ID |
| 6 | MEMBER_ASSOC_TYP… | Member Association… |
| 7 | MEMBER_ADK | Member ADK |

Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Des |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Da |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | SOURCE_SYSTEM_CD | Source Syster |
| 4 | PROCESSED_DTTM | Date/Time Pro |
| 5 | MEMBER_RK | Member Key |
| 6 | MEMBER_ID | Member ID |
| 7 | MEMBER_NM | Member Name |
| 8 | MEMBER_DESC | Member Desci |
| 9 | MEMBER_ADK | Member ADK |
| 10 | ROLL_UP_TO_PARENT… | Roll Up To Par |

33. Drag the MEMBER_RK column in the **Target table** region onto the MEMBER_RK column in the **Lookup table** region:



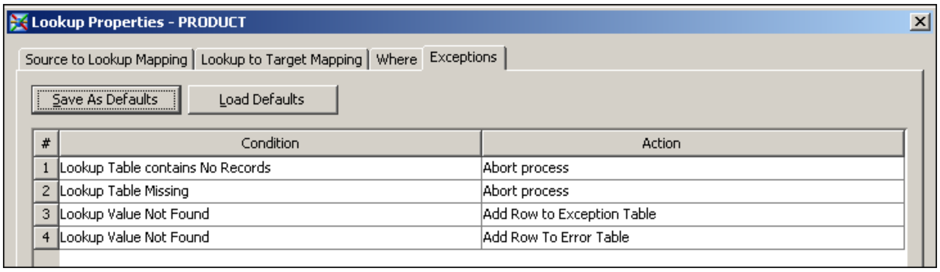
Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Des |
|---|---|---|
| 1 | MEMBER_RK | Member Key |
| 2 | VALID_FROM_… | Valid From D |
| 3 | VALID_TO_DTTM | Valid To Dat |
| 4 | MEMBER_ID | Member ID |
| 5 | SOURCE_SYST… | Source Syst |
| 6 | MEMBER_NM | Member Nar |
| 7 | MEMBER_DESC | Member Des |
| 8 | MEMBER_ADK | Member ADK |
| 9 | ROLL_UP_TO_… | Roll Up To P |
| 10 | PROCESSED_D… | Datetime Pro |
| 11 | MEMBER_ATTR1 | |
| 12 | MEMBER_ATTR2 | |
| 13 | MEMBER_ATTR3 | |

Target table: Lookup (W5WXJHLN)

| # | Column | Column Descrip… | Type | Length | Informat | Format |
|---|---|---|---|---|---|---|
| 1 | MEMBER_RK | Member Key | Numeric | 8 | (None) | (None) |
| 2 | PARENT_MEMB… | Parent Member … | Numeric | 8 | (None) | (None) |
| 3 | MEMBER_ASSO… | Member Associa… | Character | 32 | (None) | (None) |
| 4 | VALID_FROM_D… | Valid From Date… | Numeric | 8 | DATETIME… | DATETIME |
| 5 | VALID_TO_DTTM | Valid To Datetime | Numeric | 8 | DATETIME… | DATETIME |
| 6 | ORDER_NO | Order Number | Numeric | 8 | (None) | (None) |
| 7 | MEMBER_ADK | Member ADK | Character | 32 | (None) | (None) |
| 8 | PROCESSED_D… | Datetime Proces… | Numeric | 8 | DATETIME… | DATETIME |

34. On the **Exceptions** tab of the inner Lookup Properties window, select the following values in the **Action** column:



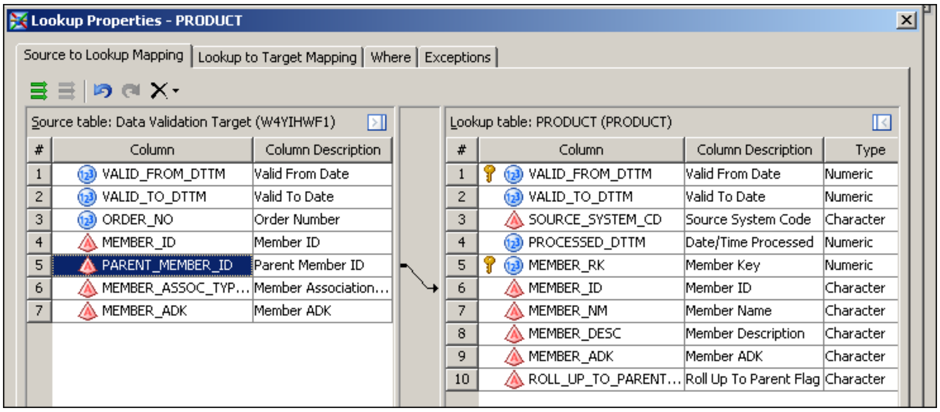
| # | Condition | Action |
|---|---|---|
| 1 | Lookup Table contains No Records | Abort process |
| 2 | Lookup Table Missing | Abort process |
| 3 | Lookup Value Not Found | Add Row to Exception Table |
| 4 | Lookup Value Not Found | Add Row To Error Table |

35. Click **OK** to close the inner Lookup Properties window.

36. On the **Lookups** tab, with the second instance of the PRODUCT table selected, click **Lookup Properties**.

    The inner Lookup Properties window appears again.

37. On the **Source to Lookup Mapping** tab of the inner Lookup Properties window, drag the PARENT_MEMBER_ID column in the **Source table** region onto the MEMBER_ID column in the **Lookup table** region:



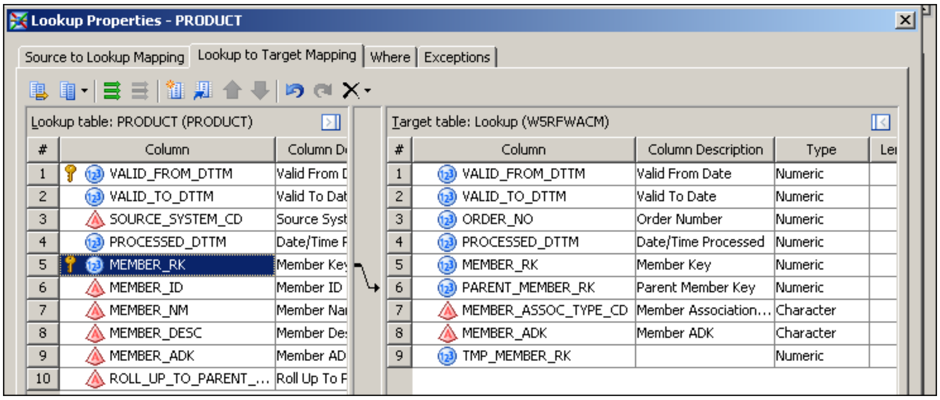
Source table: Data Validation Target (W4YIHWF1)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | MEMBER_ID | Member ID |
| 5 | PARENT_MEMBER_ID | Parent Member ID |
| 6 | MEMBER_ASSOC_TYP... | Member Association... |
| 7 | MEMBER_ADK | Member ADK |

Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Description | Type |
|---|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date | Numeric |
| 2 | VALID_TO_DTTM | Valid To Date | Numeric |
| 3 | SOURCE_SYSTEM_CD | Source System Code | Character |
| 4 | PROCESSED_DTTM | Date/Time Processed | Numeric |
| 5 | MEMBER_RK | Member Key | Numeric |
| 6 | MEMBER_ID | Member ID | Character |
| 7 | MEMBER_NM | Member Name | Character |
| 8 | MEMBER_DESC | Member Description | Character |
| 9 | MEMBER_ADK | Member ADK | Character |
| 10 | ROLL_UP_TO_PARENT... | Roll Up To Parent Flag | Character |

38. On the **Lookup to Target Mapping** tab of the inner Lookup Properties window, drag the MEMBER_RK column in the **Lookup table** region onto the PARENT_MEMBER_RK column in the **Target table** region:



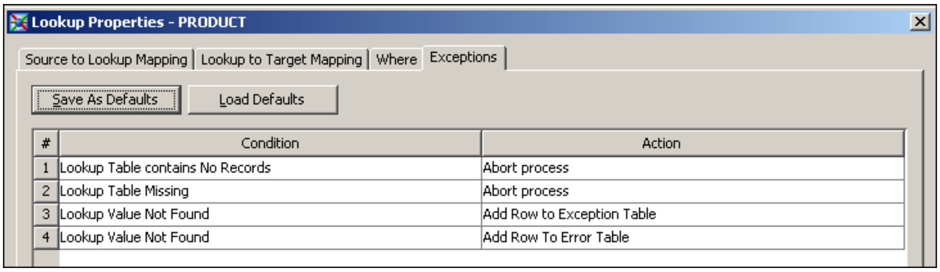
Lookup table: PRODUCT (PRODUCT)

| # | Column | Column D |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From D |
| 2 | VALID_TO_DTTM | Valid To Dat |
| 3 | SOURCE_SYSTEM_CD | Source Syst |
| 4 | PROCESSED_DTTM | Date/Time P |
| 5 | MEMBER_RK | Member Key |
| 6 | MEMBER_ID | Member ID |
| 7 | MEMBER_NM | Member Nai |
| 8 | MEMBER_DESC | Member De: |
| 9 | MEMBER_ADK | Member AD |
| 10 | ROLL_UP_TO_PARENT_... | Roll Up To F |

Target table: Lookup (W5RFWACM)

| # | Column | Column Description | Type | Le |
|---|---|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date | Numeric | |
| 2 | VALID_TO_DTTM | Valid To Date | Numeric | |
| 3 | ORDER_NO | Order Number | Numeric | |
| 4 | PROCESSED_DTTM | Date/Time Processed | Numeric | |
| 5 | MEMBER_RK | Member Key | Numeric | |
| 6 | PARENT_MEMBER_RK | Parent Member Key | Numeric | |
| 7 | MEMBER_ASSOC_TYPE_CD | Member Association... | Character | |
| 8 | MEMBER_ADK | Member ADK | Character | |
| 9 | TMP_MEMBER_RK | | Numeric | |

39. On the **Exceptions** tab of the inner Lookup Properties window, select the following values in the **Action** column:

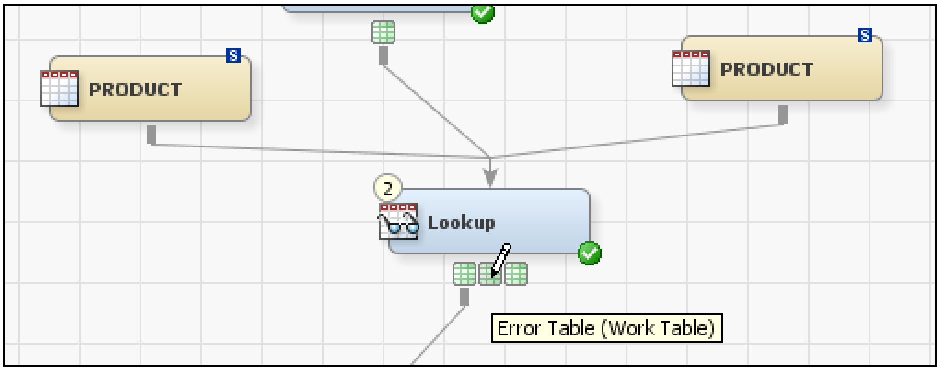


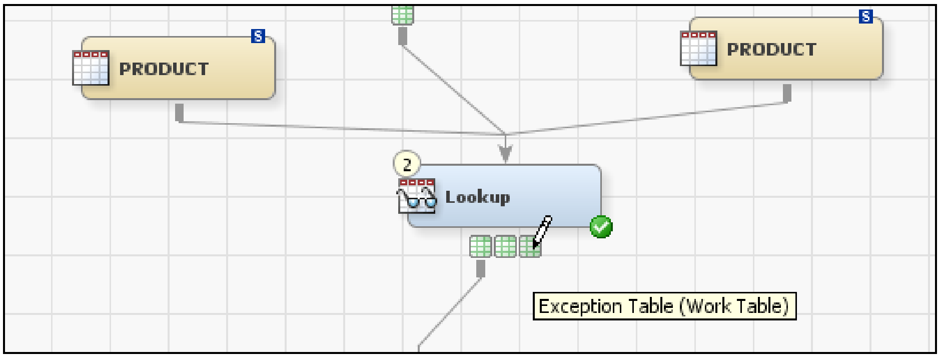
| # | Condition | Action |
|---|---|---|
| 1 | Lookup Table contains No Records | Abort process |
| 2 | Lookup Table Missing | Abort process |
| 3 | Lookup Value Not Found | Add Row to Exception Table |
| 4 | Lookup Value Not Found | Add Row To Error Table |

40. Click **OK** to close the inner Lookup Properties window.

41. Click **OK** to close the main Lookup Properties window.

42. As a result of the selections that you made on the **Errors** tab of the Lookup Properties window, the process diagram now includes an error table icon and an exception table icon:





To conform to the naming conventions of the predefined dimension types, rename the error and exception tables as explained in the following steps.

43. In the process diagram, select the Error Table icon. Right-click and select **Properties** from the pop-up menu.

    On the **General** tab, specify the following metadata name for the error table in the **Name** field: PRODUCT_ASSOC_LKUP_ERR.

    On the **Physical Storage** tab, select `Redirect to a registered library` in the **Location** field. Click the ellipsis next to the **Library** field and select **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Error Data** ⇨ **Error Data**. Specify the following name for the physical error table in the **Physical name** field: PRODUCT_ASSOC_LKUP_ERR.

    Click **OK**.

44. In the process diagram, select the Exception Table icon. Right-click and select **Properties** from the pop-up menu.

    On the **General** tab, specify the following metadata name for the exception table in the **Name** field: PRODUCT_ASSOC_LKUP_EXC.

    On the **Physical Storage** tab, select `Redirect to a registered library` in the **Location** field. Click the ellipsis next to the **Library** field and select **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Error Data** ⇨ **Error Data**. Specify the following name for the physical error table in the **Physical name** field: PRODUCT_ASSOC_LKUP_EXC.

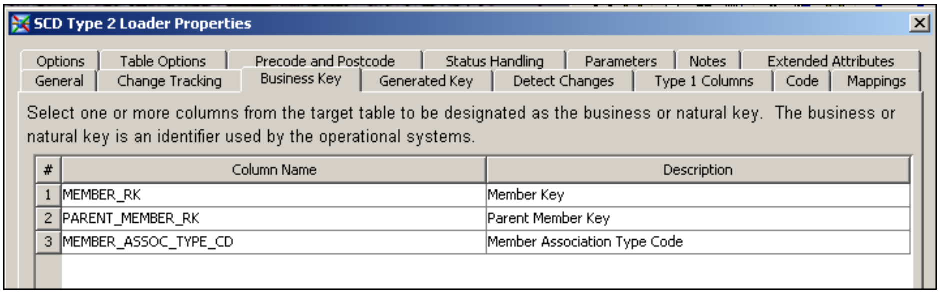
    Click **OK**.

45. In the process diagram, select the SCD Type 2 Loader transformation. Right-click and select **Properties** from the pop-up menu.

46. Select the **Business Key** tab. Click **New** to add the following columns in this order:

   - MEMBER_RK
   - PARENT_MEMBER_RK
   - MEMBER_ASSOC_TYPE_CD

   The **Business Key** tab of the SCD Type 2 Loader Properties window now looks like this:



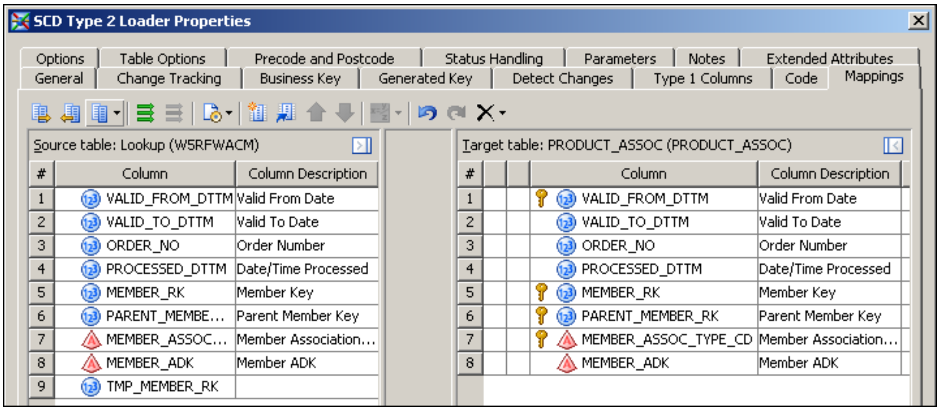
Select one or more columns from the target table to be designated as the business or natural key. The business or natural key is an identifier used by the operational systems.

| # | Column Name | Description |
|---|---|---|
| 1 | MEMBER_RK | Member Key |
| 2 | PARENT_MEMBER_RK | Parent Member Key |
| 3 | MEMBER_ASSOC_TYPE_CD | Member Association Type Code |

47. Select the **Options** tab. Provide values for some of the options as follows:

   - Under **SCD**, **Cross reference table name** is PRODUCT_ASSOC_X.
   - **Format type for dates** depends on what you load into the STAGE_PRODUCT_ASSOC table. For details, see "Setting a Valid Time Range for Data Records" on page 175.
   - Under **Additional Loader Options**, **Load time column** should contain PROCESSED_DTTM.

48. Select the **Mappings** tab:



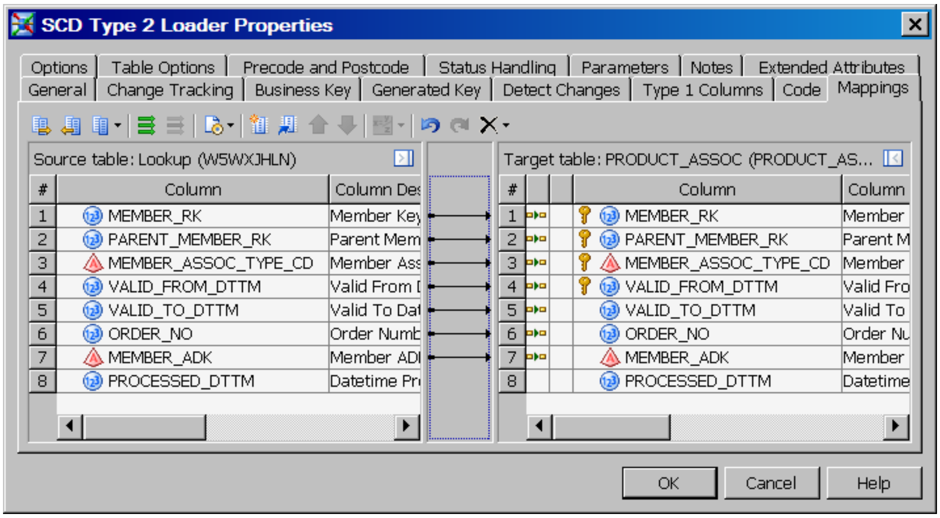
Source table: Lookup (W5RFWACM)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | PROCESSED_DTTM | Date/Time Processed |
| 5 | MEMBER_RK | Member Key |
| 6 | PARENT_MEMBE... | Parent Member Key |
| 7 | MEMBER_ASSOC... | Member Association... |
| 8 | MEMBER_ADK | Member ADK |
| 9 | TMP_MEMBER_RK | |

Target table: PRODUCT_ASSOC (PRODUCT_ASSOC)

| # | Column | Column Description |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date |
| 2 | VALID_TO_DTTM | Valid To Date |
| 3 | ORDER_NO | Order Number |
| 4 | PROCESSED_DTTM | Date/Time Processed |
| 5 | MEMBER_RK | Member Key |
| 6 | PARENT_MEMBER_RK | Parent Member Key |
| 7 | MEMBER_ASSOC_TYPE_CD | Member Association... |
| 8 | MEMBER_ADK | Member ADK |

49. Define the following mappings:



Source table: Lookup (W5WXJHLN)

| # | Column | Column Des |
|---|---|---|
| 1 | MEMBER_RK | Member Key |
| 2 | PARENT_MEMBER_RK | Parent Mem |
| 3 | MEMBER_ASSOC_TYPE_CD | Member Ass |
| 4 | VALID_FROM_DTTM | Valid From I |
| 5 | VALID_TO_DTTM | Valid To Dat |
| 6 | ORDER_NO | Order Numb |
| 7 | MEMBER_ADK | Member ADI |
| 8 | PROCESSED_DTTM | Datetime Pr |

Target table: PRODUCT_ASSOC (PRODUCT_AS...

| # | | Column | Column |
|---|---|---|---|
| 1 | 🔑 | MEMBER_RK | Member |
| 2 | 🔑 | PARENT_MEMBER_RK | Parent M |
| 3 | 🔑 | MEMBER_ASSOC_TYPE_CD | Member |
| 4 | 🔑 | VALID_FROM_DTTM | Valid Fro |
| 5 | | VALID_TO_DTTM | Valid To |
| 6 | | ORDER_NO | Order Nu |
| 7 | | MEMBER_ADK | Member |
| 8 | | PROCESSED_DTTM | Datetime |

You can achieve this result with the following steps:

a. Right-click a column name in either region and select **Map All** from the pop-up menu.

b. Select and delete the PROCESSED_DTTM-to-PROCESSED_DTTM mapping.

50. Click **OK** to save your changes and close the SCD Type 2 Loader Properties window.

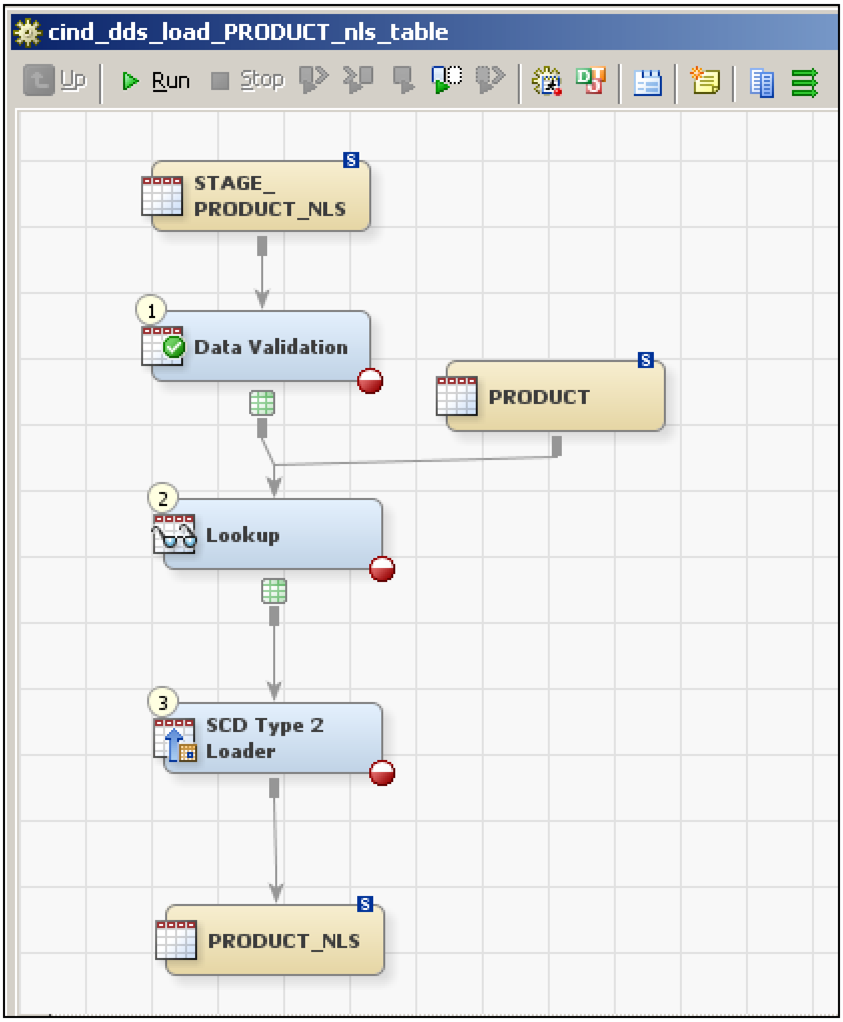
51. Select **File** ⇨ **Save** to save the contents of the job, and then close the job.

## Customize the Job That Loads the Detail Data Store Secondary Member Table

The secondary member table holds member names and descriptions in languages other than the default language. If your site uses names and descriptions in only one language, then you have no use for the secondary member table and you can ignore the job that loads it.

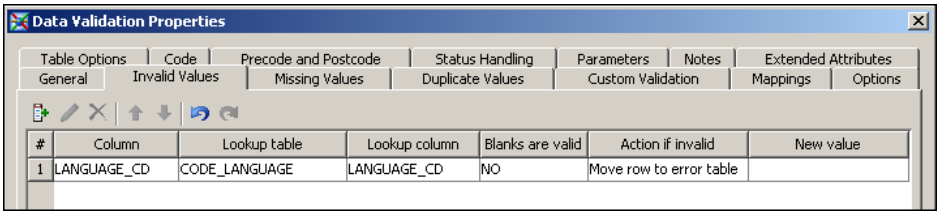
The steps that are required to customize this job are very similar to the steps that are required to customize the other jobs. Here is an outline of the required steps:

1. Display the process diagram. Replace the Lookup transformation and drag the input, output, and lookup tables into place:



2. In the process diagram, select the Data Validation transformation. Right-click to display the pop-up menu and select **Propagate Columns** ⇨ **To Selected Transformation's Targets** ⇨ **From Sources**.

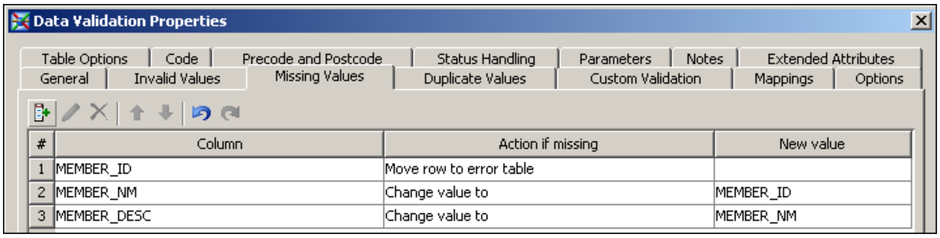
3. Right-click again and select **Properties** from the pop-up menu. On the **Invalid Values** tab of the Data Validation transformation, define a validation check for the LANGUAGE_CD column:



| # | Column | Lookup table | Lookup column | Blanks are valid | Action if invalid | New value |
|---|---|---|---|---|---|---|
| 1 | LANGUAGE_CD | CODE_LANGUAGE | LANGUAGE_CD | NO | Move row to error table | |

The required steps are identical to those for the LANGUAGE_CD validation check in the job that loads the hierarchy identification table. For details, see "Customize the Job That Loads the Detail Data Store Hierarchy Identification Table" on page 227.

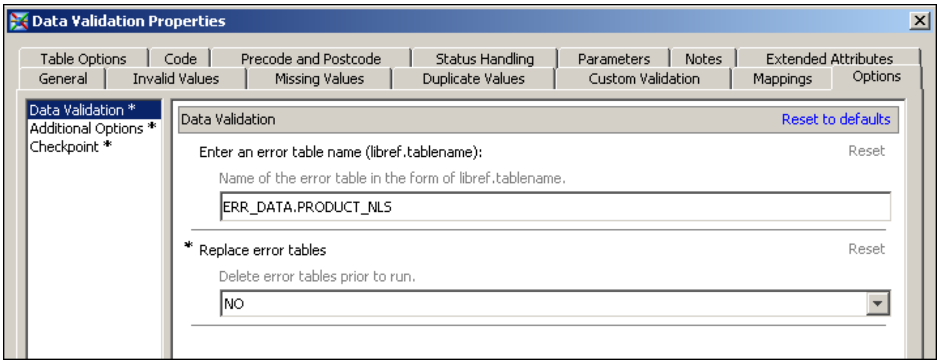
4. On the **Missing Values** tab of the Data Validation transformation, specify how to handle missing values in the member code, member name, and member description columns:



| # | Column | Action if missing | New value |
|---|---|---|---|
| 1 | MEMBER_ID | Move row to error table | |
| 2 | MEMBER_NM | Change value to | MEMBER_ID |
| 3 | MEMBER_DESC | Change value to | MEMBER_NM |

The required steps are identical to those for the same columns in the job that loads the primary member table. For details, see "Customize the Job That Loads the Detail Data Store Primary Member Table" on page 221 .

5. On the **Options** tab of the Data Validation transformation, type ERR_DATA.PRODUCT_NLS as the value of the **Enter an error table name** option:



Enter an error table name (libref.tablename):

Name of the error table in the form of libref.tablename.

ERR_DATA.PRODUCT_NLS

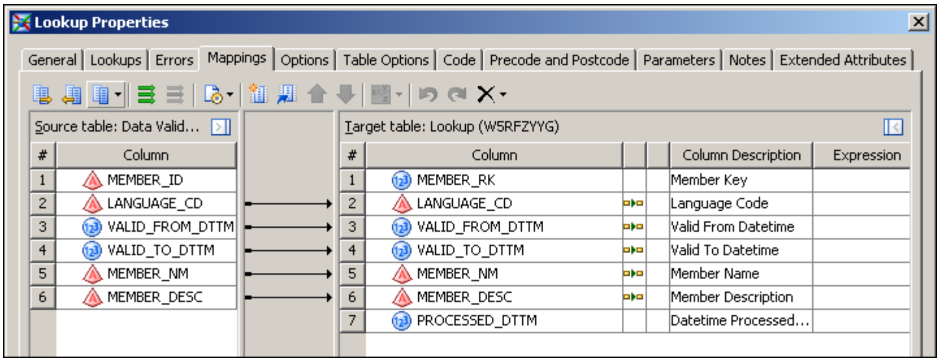
* Replace error tables

Delete error tables prior to run.

NO

6. On the **Errors** tab of the Lookup transformation, select the **Create error table** check box and the **Create exception table** check box.

7. On the **Mappings** tab of the Lookup transformation, define the following mappings:



Source table: Data Valid...

| # | Column |
|---|---|
| 1 | MEMBER_ID |
| 2 | LANGUAGE_CD |
| 3 | VALID_FROM_DTTM |
| 4 | VALID_TO_DTTM |
| 5 | MEMBER_NM |
| 6 | MEMBER_DESC |

Target table: Lookup (W5RFZYYG)

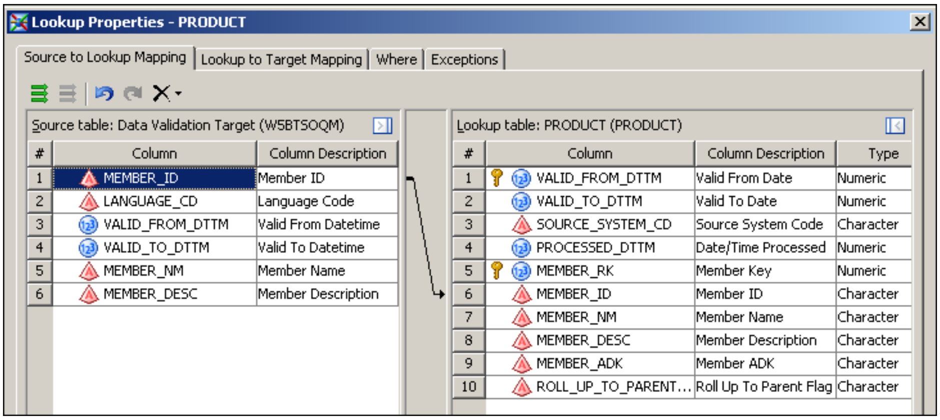
| # | Column | Column Description | Expression |
|---|---|---|---|
| 1 | MEMBER_RK | Member Key | |
| 2 | LANGUAGE_CD | Language Code | |
| 3 | VALID_FROM_DTTM | Valid From Datetime | |
| 4 | VALID_TO_DTTM | Valid To Datetime | |
| 5 | MEMBER_NM | Member Name | |
| 6 | MEMBER_DESC | Member Description | |
| 7 | PROCESSED_DTTM | Datetime Processed... | |

Except for one small difference, the required steps are identical to the steps for the Lookup transformation **Mappings** tab in the job that loads the hierarchy identification table. The difference in this task is that you must import columns from the PRODUCT_NLS table instead of the PRODUCT_ASSOC_TYPE table. For details, see "Customize the Job That Loads the Detail Data Store Hierarchy Identification Table" on page 227.

8. On the **Lookups** tab of the Lookup transformation, do everything that you did on the **Lookups** tab of the Lookup transformation for the job that loads the hierarchy identification table.

The only difference is that here you must look up an RK value for the MEMBER_ID column of the PRODUCT_NLS table instead of the DEFAULT_MEMBER_ID column of the PRODUCT_ASSOC_TYPE table. For details, see "Customize the Job That Loads the Detail Data Store Hierarchy Identification Table" on page 227.

The inner Lookup Properties window should look like this:



Lookup Properties - PRODUCT

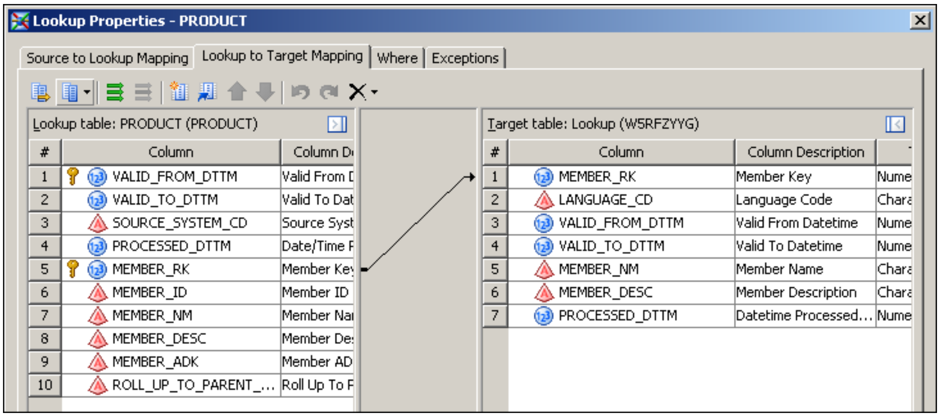
Source to Lookup Mapping | Lookup to Target Mapping | Where | Exceptions

Source table: Data Validation Target (W5BTSOQM)

| # | Column | Column Description |
|---|---|---|
| 1 | MEMBER_ID | Member ID |
| 2 | LANGUAGE_CD | Language Code |
| 3 | VALID_FROM_DTTM | Valid From Datetime |
| 4 | VALID_TO_DTTM | Valid To Datetime |
| 5 | MEMBER_NM | Member Name |
| 6 | MEMBER_DESC | Member Description |

Lookup table: PRODUCT (PRODUCT)

| # | Column | Column Description | Type |
|---|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From Date | Numeric |
| 2 | VALID_TO_DTTM | Valid To Date | Numeric |
| 3 | SOURCE_SYSTEM_CD | Source System Code | Character |
| 4 | PROCESSED_DTTM | Date/Time Processed | Numeric |
| 5 | MEMBER_RK | Member Key | Numeric |
| 6 | MEMBER_ID | Member ID | Character |
| 7 | MEMBER_NM | Member Name | Character |
| 8 | MEMBER_DESC | Member Description | Character |
| 9 | MEMBER_ADK | Member ADK | Character |
| 10 | ROLL_UP_TO_PARENT… | Roll Up To Parent Flag | Character |



Lookup Properties - PRODUCT

Source to Lookup Mapping | Lookup to Target Mapping | Where | Exceptions

Lookup table: PRODUCT (PRODUCT)

| # | Column | Column D |
|---|---|---|
| 1 | VALID_FROM_DTTM | Valid From D |
| 2 | VALID_TO_DTTM | Valid To Dat |
| 3 | SOURCE_SYSTEM_CD | Source Syst |
| 4 | PROCESSED_DTTM | Date/Time P |
| 5 | MEMBER_RK | Member Key |
| 6 | MEMBER_ID | Member ID |
| 7 | MEMBER_NM | Member Na |
| 8 | MEMBER_DESC | Member De |
| 9 | MEMBER_ADK | Member AD |
| 10 | ROLL_UP_TO_PARENT_… | Roll Up To P |

Target table: Lookup (W5RFZYYG)

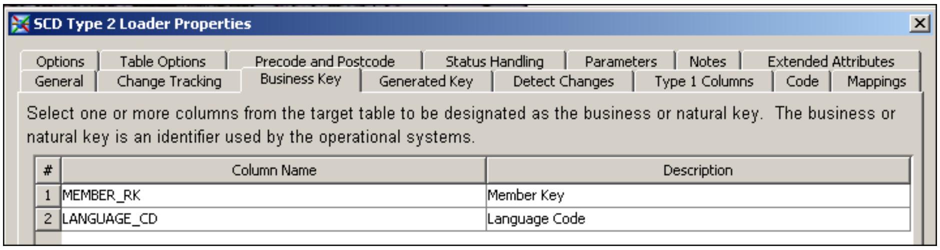
| # | Column | Column Description | T |
|---|---|---|---|
| 1 | MEMBER_RK | Member Key | Nume |
| 2 | LANGUAGE_CD | Language Code | Chara |
| 3 | VALID_FROM_DTTM | Valid From Datetime | Nume |
| 4 | VALID_TO_DTTM | Valid To Datetime | Nume |
| 5 | MEMBER_NM | Member Name | Chara |
| 6 | MEMBER_DESC | Member Description | Chara |
| 7 | PROCESSED_DTTM | Datetime Processed… | Nume |

9. Rename the error and exception tables using the string PRODUCT_NLS.

   The required steps are identical to the other jobs. For details, see either "Customize the Job That Loads the Detail Data Store Hierarchy Identification Table" on page 227 or "Customize the Job That Loads the Detail Data Store Hierarchy Structure Table" on page 236.

10. On the **Business Key** tab of the SCD Type 2 Loader transformation, click **New** to add the following columns in this order:

    - MEMBER_RK
    - LANGUAGE_CD



SCD Type 2 Loader Properties

Options | Table Options | Precode and Postcode | Status Handling | Parameters | Notes | Extended Attributes
General | Change Tracking | Business Key | Generated Key | Detect Changes | Type 1 Columns | Code | Mappings

Select one or more columns from the target table to be designated as the business or natural key. The business or natural key is an identifier used by the operational systems.

| # | Column Name | Description |
|---|---|---|
| 1 | MEMBER_RK | Member Key |
| 2 | LANGUAGE_CD | Language Code |

11. On the **Options** tab of the SCD Type 2 Loader transformation, provide values for some of the options as follows:

    - Under **SCD**, **Cross reference table name** is PRODUCT_NLS_X.

- **Format type for dates** depends on what you load into the STAGE_PRODUCT_ASSOC table. For details, see "Setting a Valid Time Range for Data Records" on page 175.
- Under **Additional Loader Options**, **Load time column** should contain PROCESSED_DTTM.

12. On the **Mappings** tab of the SCD Type 2 Loader transformation, right-click to display the pop-up menu and select **Map All**. Select and delete the PROCESSED_DTTM-to-PROCESSED_DTTM mapping.
13. Select **File** ⇨ **Save** to save the contents of the job, and then close the job.

## Loading New Dimension Types into the SDM

To load new dimension types into the SDM, run the solnsvc_2000_load_dimension_types job. On the **Folders** tab, this job is in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder.

Run the job and then review the log.

The job can run only if SAS Remote Services and the managed servers are running on the middle-tier server. See the System Administration part of this book for more information.

## Creating Dimensions in a New Dimension Type

The available methods for creating a dimension are the same for new dimension types and predefined dimension types. For details, see "Creating a Dimension" on page 187.

## Loading Members and Hierarchies into a Dimension That Belongs to a New Dimension Type

The procedure for loading members and hierarchies into a dimension is the same for new dimension types and predefined dimension types. For details, see "Loading Members and Hierarchies into a Dimension" on page 193.

It is important to load the detail data store tables in this order:

1. primary member table
2. hierarchy identification (ASSOC_TYPE) table
3. hierarchy structure (ASSOC) table
4. secondary member (NLS) table

As an aid to following this order, you can add sequence numbers to the names of the jobs that you created.

# *Chapter 20*
# Loading Measures

**Overview: Measures and Metrics** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253

**Moving Measure Data from Its Source to the Detail Data Store** . . . . . . . . . . . . . . 253

**Moving Measure Data from the Detail Data Store to the SDM** . . . . . . . . . . . . . . . 255

## Overview: Measures and Metrics

*Measure* and *metric*, as SAS solutions use these terms, have different but related meanings:

- A measure is a variable that is subject to numeric measurement. Examples are profit margin, number of employees, and number of complaints.
- A metric is a numeric value of a measure. Examples are 5% profit margin, 950 employees, and 400 complaints.

A measure typically has many metrics associated with it. The metrics also depend on other variables, such as the time period or the relevant organization. These other variables are represented by dimensions (a time dimension and an organization dimension, for example).

This chapter focuses on measures. "Loading Metrics" on page 257 focuses on metrics.

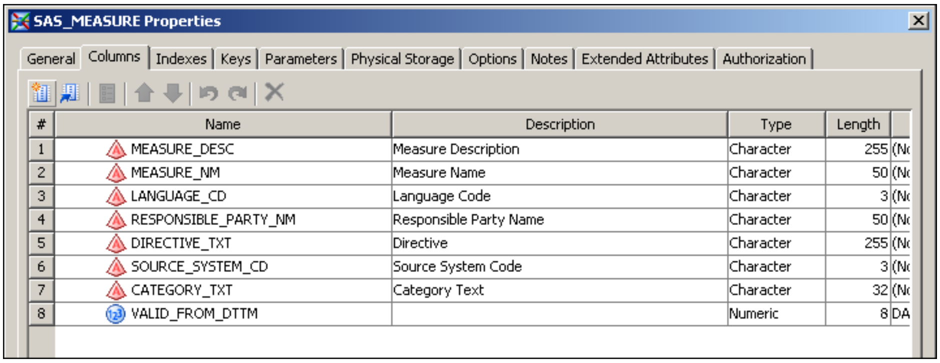
## Moving Measure Data from Its Source to the Detail Data Store

SAS supplies several hundred measures in the predefined SAS_MEASURE source table. On the **Folders** tab, this table is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **SAS Supplied** folder.

If the measures in the SAS_MEASURE table are the only measures that you need, then you should simply load the measures from this table into the detail data store.

If you need additional measures, then load the additional measures along with the predefined measures into the detail data store and from the detail data store to the SDM. To load additional measures into the detail data store:

1. Create a second source table of measures. Your source table of measures must have the same eight-column layout as the SAS_MEASURE table:



| # | Name | Description | Type | Length | |
|---|---|---|---|---|---|
| 1 | MEASURE_DESC | Measure Description | Character | 255 | (N |
| 2 | MEASURE_NM | Measure Name | Character | 50 | (N |
| 3 | LANGUAGE_CD | Language Code | Character | 3 | (N |
| 4 | RESPONSIBLE_PARTY_NM | Responsible Party Name | Character | 50 | (N |
| 5 | DIRECTIVE_TXT | Directive | Character | 255 | (N |
| 6 | SOURCE_SYSTEM_CD | Source System Code | Character | 3 | (N |
| 7 | CATEGORY_TXT | Category Text | Character | 32 | (N |
| 8 | VALID_FROM_DTTM | | Numeric | 8 | DA |

In building records for a second source table of measures, note the following:

- Language Code is not used but must contain a value. Use the value **`en`** in every record to be consistent with the SAS_MEASURE table.
- Responsible Party Name is not used. Leave this field empty.
- Directive is not used. Leave this field empty.
- Source System Code should be *ETL* in every record.
- Category Text is used to group measures in SAS Strategy Management. Reflect on how measures should be grouped for users before placing values in this column.
- Measure Name and Measure Description identify each measure for users of SAS Strategy Management.

2. Make your source table available in SAS Data Integration Studio. Right-click a metadata folder, select **Register Tables** and do the following:

   a. Select SAS as the table type.

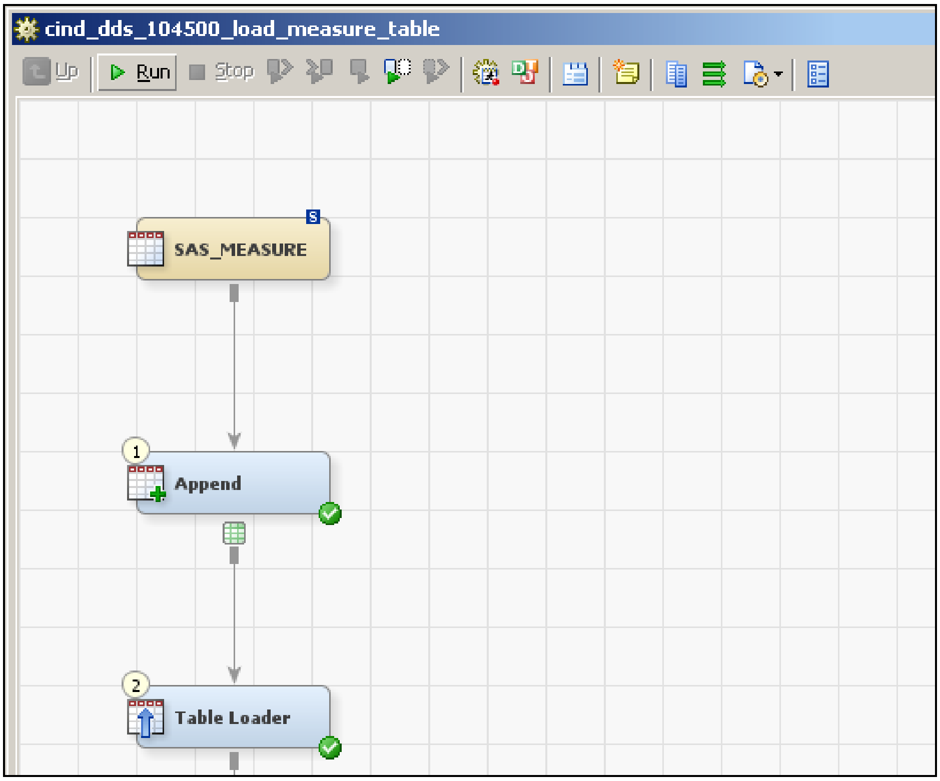
   b. If necessary, enter the server login information.

   c. Select the library where the data resides. (You must register the library first.)

   d. Select the table from the list of tables in the selected library.

   e. Select the SAS Data Integration Studio folder in which you want to place the table.

3. Drag your source table onto the cind_dds_104500_load_measure_table job and connect the source table to the append transformation:



On the **Folders** tab, this job is in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder.

4. Run the cind_dds_104500_load_measure_table job.

## Moving Measure Data from the Detail Data Store to the SDM

To load the contents of the MEASURE table into the SDM, run the solnsvc_3300_load_measure_table job. On the **Folders** tab, this job is in the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder.

# *Chapter 21*
# Loading Metrics

**Overview: Measures and Metrics** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257

**Data Pathways for Metrics** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257

**Preparing a Source Table of Metric Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 258

**Preparing Jobs to Load Metric Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 259

## Overview: Measures and Metrics

*Measure* and *metric*, as SAS solutions use these terms, have different but related meanings:

- A measure is a variable that is subject to numeric measurement. Examples are profit margin, number of employees, and number of complaints.
- A metric is a numeric value of a measure. Examples are 5% profit margin, 950 employees, and 400 complaints.

A measure typically has many metrics associated with it. The metrics also depend on other variables, such as the time period or the relevant organization. These other variables are represented by dimensions (a time dimension and an organization dimension, for example).

This chapter focuses on metrics. “Loading Measures” on page 253 focuses on measures.

## Data Pathways for Metrics

In the SDM, the software maintains a separate metric table for each combination of hierarchies that characterizes the metric data. For example, metrics that are associated with time hierarchy A will be in one table, metrics that are associated with time hierarchy B will be in another table, and metrics that are associated with both time hierarchy A and organization hierarchy Z will be in a third table. The software creates a new metric table each time data that is associated with a given combination of hierarchies arrives for the first time. If data arrives for a combination of hierarchies that has been used before, then that data is appended to an existing metric table.

Each metric table in the SDM has a name that enables users of SAS Strategy Management or the KPI Viewer to select it for use.

Metric data can enter the SDM metric tables in three different ways:

- A SAS Human Capital Management job can compute metrics from data in the HCM Data Mart. An optional job can then load the computed HCM metrics into SDM metric tables. For details, see “Loading HCM Metrics into a Metric Table” on page 291.
- A SAS Solutions Services job can load data into SDM metric tables from source tables that you build. This data pathway is the subject of the rest of this chapter.
- Users of the SAS Financial Management Add-in for Microsoft Excel can export data to SDM metric tables.

## Preparing a Source Table of Metric Data

Prepare a separate source table for each distinct metric table in the SDM. Name each source table in a way that indicates which metric table it is the source for.

Each source table of metric data must have the following columns:

- A column named MEASURE_NM with format $100. For each row of the table, this column must contain the name of an existing measure.
- A column named SOURCE_SYSTEM_CD with format $3. For each row of the table, this column must contain the code ETL.
- One or more columns that represent dimension types. A column that represents the time dimension type is required. Columns for all other dimension types are optional. Each dimension type column must have a format of $32. and must have a name that ends with the three characters _ID. For each row of the table, each dimension type column must contain a member code of a member that belongs to the hierarchy within that dimension type that is specified in the job that loads the data into the SDM.
- One or more columns with a NUMERIC format to hold the numeric values of metrics. For example, a source table could have a single numeric column named VALUE or two numeric columns named VALUE1 and VALUE2, or ACTUAL and BUDGET. Having numeric columns named ACTUAL and BUDGET is an alternative to having an ANALYSIS_ID dimension type column, which can contain ACTUAL and BUDGET member codes.
- A column named DIRECTIVE_TXT with format $255. For each row of the table, either leave this column empty or use it to specify the directive that you want to associate with the metric in any scorecard that displays the metric. A user who views a scorecard can click a directive to produce an appropriate action.

  You can specify any predefined directive or any custom directive that has been defined at your site. The following predefined directives are most likely to be useful in this context:

  Launch a URL

  Displays a specified Web page.

  Place the following in the DIRECTIVE_TXT column: **`URL_Redirect&url=`*****`target_url`***

  For example: **`URL_Redirect&url=http://www.sas.com/`**

  Open a Document

  Opens a specified document.

Place the following in the DIRECTIVE_TXT column:
```
OpenDocument&Document=GUID_of_document
```

For example:
```
OpenDocument&Document=fedb550b-0a0b-0b93-01a7-401ccccec5c
5
```

You create a source table of metric data on the server, and then import it to a folder in SAS Data Integration Studio. You might want to create a folder that is dedicated to source tables of metric data.

To import a source table:

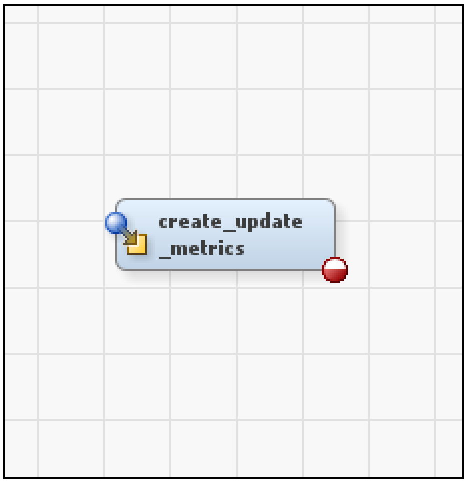
1. Right-click a metadata folder and select **Register Tables**.
2. Select SAS, and then click **Next**.
3. If necessary, enter the server login information.
4. Select the SAS library where the source table resides. If necessary, you can define the SAS library first.
5. Select the desired source table from the list of tables in the library.

## Preparing Jobs to Load Metric Data

Prepare a separate job for each distinct metric table in the SDM.

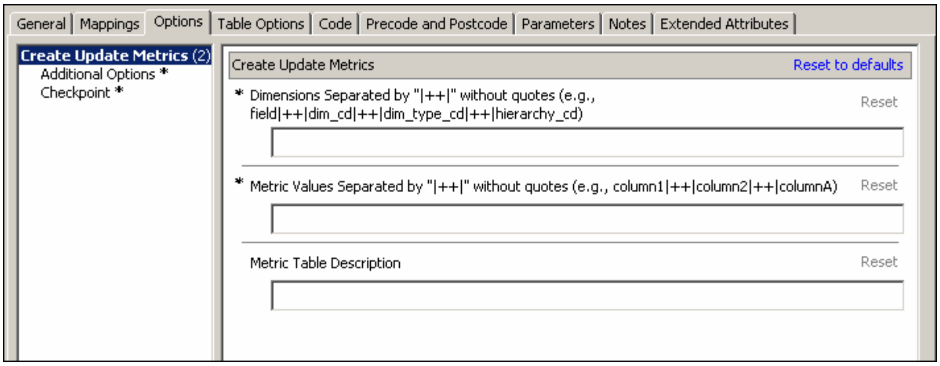
To prepare a job:

1. In the **Products** ⇨ **SAS Solutions Services** ⇨ **Jobs** folder on the **Folders** tab, create a copy of the solnsvc_3400_load_metric_table job, using one of the methods described in "Copy Jobs" on page 170.
2. Name the job in a way that indicates which metric table it loads.
3. Open the process diagram for the job by double-clicking the job title.
4. Drag the appropriate source table onto the diagram and connect it to the create_update_metrics transformation:



5. Right-click the create_update_metrics icon and select **Properties** from the pop-up menu.

6. In the Properties window, select the **Options** tab:



Create Update Metrics

* Dimensions Separated by "|++|" without quotes (e.g., field|++|dim_cd|++|dim_type_cd|++|hierarchy_cd)

* Metric Values Separated by "|++|" without quotes (e.g., column1|++|column2|++|columnA)

Metric Table Description

7. Supply appropriate option values as explained below.

8. Click **OK** to close the Properties window.

9. Select **File** ⇨ **Save**.

For each source table column that represents a dimension type, the Dimensions option must contain a four-item colon-delimited list: *field:dim_cd:dim_type_cd:hierarchy_cd*

The four items in this list are the following:

- *field* is the name of the column in the source table.
- *dim_cd* is the code of the dimension to which all the members in this column belong.

  A dimension code that is used in a metric table must not be a MySQL reserved word. See "MySQL Reserved Words" on page 735.
- *dim_type_cd* is the code of the dimension type to which all the members in this column belong.
- *hierarchy_cd* is the code of the hierarchy to which all the members in this column belong.

For a source table that has two or more columns that represent dimension types, separate the colon-delimited lists with the vertical bar symbol: |.

For a source table that uses only the Time dimension type, for example, the Dimensions option value might look like this:

```
PERIOD_ID:TIME_DIM1:TIME:TIME_HIER1
```

For a source table that uses both Time and Analysis, the Dimensions option value might look like this:

```
PERIOD_ID:TIME_DIM1:TIME:TIME_HIER1|

ANALYSIS_ID:ANALYSIS_DIM1:ANALYSIS:ANALYSIS_HIER1
```

For a source table that uses Time and Analysis and Internal Organization, the Dimensions option value might look like this:

```
PERIOD_ID:TIME_DIM1:TIME:TIME_HIER1|
     ANALYSIS_ID:ANALYSIS_DIM1:ANALYSIS:ANALYSIS_HIER1|
          ORG_ID:ORG_DIM1:INTORG:ORG_HIER1
```

To obtain the necessary codes for dimension types, dimensions, and hierarchies, use the SAS Solutions Dimension Editor.

The Metric Values option value must be a space-delimited list of the names of the columns that contain numeric values.

For example, for a source table that has a single column of numeric values named VALUE, the Metric Values option value must be:

```
VALUE
```

For a source table that has two columns of numeric values named ACTUAL and BUDGET, the Metric Values option value must be:

```
ACTUAL
BUDGET
```

The Metric Table Description option value is the name that is given to the metric table in the SDM, which users of SAS Strategy Management and the KPI Viewer can use to select it. This name is used only if the job creates a metric table. If a metric table with the specified combination of hierarchies already exists, then that table keeps the name that it already has, no matter what you specify here.

Be careful not to use the same value for the Metric Table Description option with different values for the Dimensions option. If you accidentally do this, then users of SAS Strategy Management and the KPI Viewer will see multiple metric tables that have the same name.

If a metric table with the specified combination of hierarchies already exists, then the job updates the existing metric table according to the following rules:

- Any record that contains a new combination of measure name and dimension members is added to the existing table.
- Any record that contains a combination of measure name and dimension members that is already represented by a record in the existing table replaces the record that it matches. The result is to update the numeric value or values in the existing record.
- Any record in the existing table for which no matching record is loaded remains in the table unchanged.

# *Chapter 22*
# Creating a Stored Process from a SAS Data Integration Studio Job

**Creating a Stored Process from a SAS Data Integration Studio Job** . . . . . . . . . . 263

## Creating a Stored Process from a SAS Data Integration Studio Job

You can make any SAS Data Integration Studio job available as a stored process. This gives a larger set of users the ability to run the code, which might be appropriate in some cases.

Before you create a stored process from a SAS Data Integration Studio job, make sure that you have made all appropriate modifications to the job. This includes specifying appropriate values for any job options.

To create a stored process from a SAS Data Integration Studio job, first do the following in SAS Data Integration Studio:

1. Select the job, and then select **Open**.

   The process diagram appears in the Process Designer.

2. In the Process Designer, select the **Code** tab.

   The code for the job is displayed.

3. Select **File** ⇨ **Save to File** ⇨ **Local**.

   The Save File window appears.

4. Use the Save File window to specify the target location and name for the file that will contain the code, and then click **Save**.

   Save the file in a location such as the following: ***`SAS-config-dir`***`\lev1\SASApp\SASEnvironment\[Solutions Services, HumanCapitalManagement]\UserDefined`. (Create the `UserDefined` directory if it does not already exist.)

Next, use an appropriate editor to do the following:

1. Open the saved file.

2. At the beginning of the file, add the following statement:

   ```
   %rptinit;
   ```

3. At the end of the file, add the following statements:

```
%include
sasautos(etlstatus.sas);
%stpend;
```

The %INCLUDE statement will create a job status report, which lets you see the status of your job.

4. Save these changes and close the editor.

Register the stored process in SAS Management Console. For instructions, see “Working with Stored Processes” in the *SAS Solutions Services: Customization Guide.*

# *Chapter 23*
# Loading Data for SAS Human Capital Management

**Overview of Data Needs for SAS Human Capital Management . . . . . . . . . . . . . . 266**

**The Locale for SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . . . . 266**

**Competency Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 266**

**Moving SAS Human Capital Management Data from Its Source to the Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 267**
Overview of Moving SAS Human Capital Management Data from Its Source to the Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 267
Loading Organization Dimension Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . 268
Loading Time Dimension Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 268
Loading Staging Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 269

**Loading HCM Data from the Staging Tables to the Detail Data Store . . . . . . . . . 271**

**Moving Data from the Detail Data Store to the SDM . . . . . . . . . . . . . . . . . . . . . . 273**

**Auxiliary Files and Information Sources . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274**
Overview of Auxiliary Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274
Editing the Properties Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274
Editing the SAS Macro Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274
SAS_DEFAULT_PROPERTIES MySQL Table . . . . . . . . . . . . . . . . . . . . . . . . . . 275
Editing the HCM Formats . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 275
Editing the PREBUILD.SAS Macro File . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 278
Editing HCM Measures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 279

**Moving Data from the Detail Data Store to the HCM Data Mart . . . . . . . . . . . . . 279**
Overview of Moving Data from the Detail Data Store to the HCM Data Mart . . . 279
Loading the HCM Data Mart by Running One Job at a Time . . . . . . . . . . . . . . . . 281
Loading the HCM Data Mart by Running the Umbrella Job . . . . . . . . . . . . . . . . . 282
Loading Certain HCM Data Mart Tables Directly . . . . . . . . . . . . . . . . . . . . . . . . . 285
Summary of Data Flow into HCM Data Mart Tables . . . . . . . . . . . . . . . . . . . . . . 289
Loading SAS Human Capital Management Users . . . . . . . . . . . . . . . . . . . . . . . . 291

**Loading HCM Metrics into a Metric Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 291**

**Creating the HCM Information Maps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 293**

## Overview of Data Needs for SAS Human Capital Management

This chapter guides you through the entire process of loading data for SAS Human Capital Management. However, it is not self-contained. It contains many references to other chapters. Some of these other chapters focus on specific aspects of SAS Human Capital Management, while others cover topics that apply across all solutions.

Before you can load any data for SAS Human Capital Management, you must set up your SAS Data Integration Studio environment,as described in "Setting Up the SAS Data Integration Studio Environment" on page 169.

To obtain a general understanding of how the SAS Data Integration Studio environment enables you to supply data to solutions, read "Using SAS Data Integration Studio to Supply Data to Solutions" on page 171.

## The Locale for SAS Human Capital Management

The language in which data is displayed to HCM users depends on the following settings:

- The HCM Data Mart is governed by the locale that is set by the SAS Solutions Services installation. The locale cannot be changed after installation. If your locale is incorrect, contact Technical Support.

  The locale code in the SAS_DEFAULT_PROPERTIES table is used to select the labels and titles for a particular language. For information about modifying these labels and titles, see the Administration Application part of this book.

- In all the staging tables that hold data for SAS Human Capital Management, the Language Code column must contain the values from the languages in the CODE_LANGUAGE table. For a detailed discussion of loading language codes, see "Loading Language Codes and Data Locale Codes" on page 181.

## Competency Tables

There are detail data store tables and corresponding staging tables for competency data. However, SAS Solutions Services does not include HCM Data Mart tables for this data or jobs for loading this data into the HCM Data Mart.

If you want to load competency data using SAS Solutions Services, discuss your needs with your SAS consultant. This chapter's discussion of steps for loading HCM data does not apply to competency data.

The following detail data store competency tables are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** metadata folder on the **Folders** tab:

- COMPETENCY
- COMPETENCY_ASSOC

- COMPETENCY_ASSOC_TYPE
- COMPETENCY_CATALOG
- COMPETENCY_CATEGORY
- COMPETENCY_CATEGORY_CLASS
- COMPETENCY_CLASS
- COMPETENCY_EVIDENCE
- COMPETENCY_TYPE
- COMPETENCY_WEIGHT_BOUNDS
- COMPETENCY_X_TAXONOMY_SOURCE
- EMPLOYEE_X_COMPETENCY
- EMPLOYEE_X_EVIDENCE
- EMPLOYEE_X_JOB_X_COMPETENCY
- EVIDENCE_TYPE
- JOB_X_COMPETENCY
- SPECIAL_REF_COMP
- TAXONOMY
- TAXONOMY_CATALOG
- TAXONOMY_COMP_CLASS
- TAXONOMY_SOURCE
- TAXONOMY_SPECIAL_REF_COMP
- WEIGHT_TYPE

The corresponding staging competency tables are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** metadata folder on the **Folders** tab.

For information about the columns of all competency tables, see the Data Model part of this book.

## Moving SAS Human Capital Management Data from Its Source to the Staging Tables

### *Overview of Moving SAS Human Capital Management Data from Its Source to the Staging Tables*

You can load the staging tables that hold data for SAS Human Capital Management in any order. This section lists the staging tables that are used by SAS Human Capital Management alphabetically, ignoring the STAGE_ prefix.

*Note:* The competency staging tables are not listed here. See "Competency Tables" on page 266.

For information about the columns of the HCM staging tables, see the Data Model part of this book. For a general discussion of loading staging tables, see "Moving Data from Its Source to the Staging Tables" on page 173.

The staging tables that receive predefined data from the SAS_ tables are loaded by the same jobs that load the predefined data into the detail data store tables. You do not load these staging tables as a separate prior step. See "Loading HCM Data from the Staging Tables to the Detail Data Store" on page 271.

## Loading Organization Dimension Staging Tables

You must load the following staging tables for the organization dimension type in the following order:

1. INTERNAL_ORG
2. INTERNAL_ORG_ASSOC
3. INTERNAL_ORG_ASSOC_TYPE
4. INTERNAL_ORG_NLS, if multiple languages are loaded into the detail data store

For a detailed discussion of these tables, see the following sections:

- "Tables for Each Dimension Type " on page 194
- "Requirements for All or Most Dimension Types " on page 196
- "Special Requirements for the Organization Dimension Type" on page 198

The following macro variables in the prebuild.sas file must have values that correctly designate the organization dimension and the organization hierarchies that SAS Human Capital Management uses:

- INTORG_DIMENSION_CD
- HIERS
- NUMBER_OF_HIERS

*Note:* INTORG_DIMENSION_CD is used for display purposes only, and is accurate only if the INTERNAL_ORG dimension is loaded to the SDM, and the value of this macro variable is updated with the corresponding SDM Dimension Code.

For details about these HCM macro variables, see "Macro Variables in the PREBUILD.SAS Macro File" on page 304.

## Loading Time Dimension Staging Tables

If SAS Strategy Management is installed and you would like to view HCM metrics in SAS Strategic Management, load the following staging tables for the time dimension type in the following order:

1. TIME_PERIOD
2. TIME_PERIOD_ASSOC_TYPE
3. TIME_PERIOD_ASSOC
4. TIME_PERIOD_NLS, if multiple languages are loaded into the detail data store

For a detailed discussion of these tables, see the following sections:

- "Tables for Each Dimension Type " on page 194
- "Requirements for All or Most Dimension Types " on page 196
- "Special Requirements for the Time Dimension Type" on page 200

SAS Human Capital Management uses time periods when it generates metrics for the SDM. The SAS Solutions Services installation places a set of time periods sufficient for this purpose in the SDM. However, what the installation procedure provides is limited in two respects:

- The latest installed time period is December 2010. Beyond that, you must supply time periods. One way to do that is to load them into the SDM through the staging tables and detail data store tables for the time dimension type.
- In order to generate metrics in the SDM, SAS Human Capital Management requires that the time periods be in both the SDM and the detail data store. The installation procedure does not place any time periods in the detail data store. One way to get time periods into the detail data store is to export the installed time periods from the SDM to the staging tables as explained in "Exporting and Promoting Members and Hierarchies" on page 209, and then load them from the staging tables to the detail data store. Alternatively, you can load your own set of time periods into the SDM, through the staging tables and detail data store tables for the time dimension type.

## *Loading Staging Tables*

Load the following staging tables:

1. ABSENCE_TYPE
2. ACADEMIC_CREDIT
3. ACADEMIC_HONORS
4. ACTION_REASON
5. ACTION_TYPE
6. APPLICATION_STATUS
7. ATTENDANCE_STATUS
8. CODE_LANGUAGE

   For details, see "Loading Language Codes and Data Locale Codes" on page 181.
9. COMPENSATION
10. COMPENSATION_TYPE
11. COUNTY
12. COURSE_LEVEL
13. DATES_OF_ATTENDANCE
14. DEGREE_CONCENTRATION
15. DEGREE_OPTION
16. DEGREE_PROGRAM
17. DEGREE_TYPE
18. EDUCATION_HISTORY
19. EDUCATION_LEVEL
20. EDUCATION_VALUE
21. EDUCATION_VALUE_SYSTEM
22. EDUCATION_VALUE_TYPE

23. EEO_CLASS
24. EMPLOYEE
25. EMPLOYEE_ABSENCE
26. EMPLOYEE_ACTION
27. EMPLOYEE_X_INTERNAL_ORG
28. EMPLOYEE_X_JOB
29. EMPLOYMENT_APPLICATION
30. EMPLOYEE_STATUS
31. EMPLOYEE_TYPE
32. EMPLOYEE_UNION
33. ENROLLMENT_STATUS
34. ETHNICITY
35. EXEMPT_STATUS
36. EXTERNAL_ORG
37. EXTERNAL_ORG_ADDRESS
38. FICE
39. FLSA_STATUS
40. GENDER
41. GRADUATING_DEGREE
42. HONORS_PROGRAM
43. JOB
44. JOB_GROUP
45. JOB_POSITION
46. MARITAL_STATUS
47. MILITARY_EXPERIENCE
48. MILITARY_EXPERIENCE_TYPE
49. OTHER_HONORS
50. PAY_LEVEL
51. PAY_LEVEL_STRUCTURE
52. POSITION_PERMANENCE
53. POSITION_STATUS
54. RECRUITMENT_SOURCE
55. REJECTION_REASON
56. SCHOOL_DEPT
57. SCHOOL_DEPT_TYPE
58. SCHOOL_NAME_TYPE
59. SCHOOL_OR_INSTITUTION

60. SCHOOL_TYPE
61. STATE_REGION
62. TIME_FREQUENCY
63. TIME_UNIT_OF_MEASURE

## Loading HCM Data from the Staging Tables to the Detail Data Store

For each detail data store table, there is a job whose sole purpose is to load that table. The name of the job that loads a given detail data store table contains the name of the target table. For example, to load data into the COMPENSATION table, you run the cind_dds_107200_load_compensation_table job.

*Note:* There are no jobs for the competency tables. See "Competency Tables" on page 266.

Because some detail data store tables have data dependencies on other detail data store tables, there are constraints on the order in which you can load the detail data store tables. The order in which you should run the jobs is indicated by the six-digit sequence numbers that are in the job names. If you follow the sequence numbers, you respect all the inter-table dependencies.

For a general discussion of loading detail data store tables, see "Using SAS Data Integration Studio to Supply Data to Solutions" on page 171.

All detail data store jobs are in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **Jobs** folder on the **Folders** tab. Not all jobs in the **Jobs** folder are relevant to SAS Human Capital Management. If SAS Human Capital Management is the only solution that you are concerned with, then you should run the jobs that load the following detail data store tables:

1. COUNTRY
2. CODE_LANGUAGE
3. PERIOD_TYPE
4. SOURCE_SYSTEM
5. TIME_FREQUENCY
6. ABSENCE_TYPE
7. ACTION_REASON
8. ACTION_TYPE
9. APPLICATION_STATUS
10. COMPENSATION_TYPE
11. COUNTY
12. CURRENCY
13. EDUCATION_LEVEL
14. EEO_CLASS
15. EMPLOYEE_STATUS

16. EMPLOYEE_TYPE
17. EMPLOYEE_UNION
18. ETHNICITY
19. EXEMPT_STATUS
20. FLSA_STATUS
21. GENDER
22. JOB_GROUP
23. JOB
24. MARITAL_STATUS
25. MILITARY_EXPERIENCE_TYPE
26. MEASURE

    For a discussion of loading measures, see "Loading Measures" on page 253.

27. PAY_LEVEL_STRUCTURE
28. PAY_LEVEL
29. POSITION_PERMANENCE
30. POSITION_STATUS
31. RECRUITMENT_SOURCE
32. REJECTION_REASON
33. STATE_REGION
34. TIME_UNIT_OF_MEASURE
35. TIME_PERIOD

    The TIME_PERIOD table and the two tables that follow are subject to special considerations. For details, see "Loading Time Dimension Staging Tables" on page 268.

36. TIME_PERIOD_ASSOC_TYPE
37. TIME_PERIOD_ASSOC
38. EMPLOYEE
39. INTERNAL_ORG
40. INTERNAL_ORG_ASSOC_TYPE
41. INTERNAL_ORG_ASSOC
42. JOB_POSITION
43. EXTERNAL_ORG
44. EXTERNAL_ORG_ADDRESS

    The EXTERNAL_ORG and EXTERNAL_ORG_ADDRESS tables are needed if the education tables are being loaded.

45. COMPENSATION
46. EMPLOYEE_X_INTERNAL_ORG
47. EMPLOYEE_X_JOB

48. EMPLOYEE_ABSENCE
49. EMPLOYEE_ACTION
50. EMPLOYMENT_APPLICATION
51. MILITARY_EXPERIENCE
52. ACADEMIC_CREDIT

    Tables numbered 53 or above are Competency tables. These tables are not required for core HCM functionality.

53. ACADEMIC_HONORS
54. ATTENDANCE_STATUS
55. COURSE_LEVEL
56. DEGREE_CONCENTRATION
57. DEGREE_OPTION
58. DEGREE_PROGRAM
59. DEGREE_TYPE
60. EDUCATION_VALUE_SYSTEM
61. EDUCATION_VALUE_TYPE
62. ENROLLMENT_STATUS
63. FICE
64. GRADUATING_DEGREE
65. HONORS_PROGRAM
66. OTHER_HONORS
67. SCHOOL_DEBT_TYPE
68. SCHOOL_TYPE
69. SCHOOL_NAME_TYPE
70. SCHOOL_OR_INSTITUTION
71. SCHOOL_DEBT
72. EDUCATION_HISTORY
73. EDUCATION_VALUE
74. DATES_OF_ATTENDANCE

## Moving Data from the Detail Data Store to the SDM

Before you load any data into the HCM Data Mart, load the following data into the SDM:

- currencies for your currency dimension.
- organizations and organization hierarchies for your organization dimension.

- time periods and time hierarchies for your time dimension, if appropriate. For a discussion of the issues concerning time, see "Loading Time Dimension Staging Tables" on page 268.
- measures. For details, see "Loading Measures" on page 253.

For a detailed discussion of loading the members and hierarchies of any dimension into the SDM, see "Moving Member and Hierarchy Data from the Detail Data Store to the SDM" on page 202.

If any of these jobs run with errors, then correct those errors and load the SDM data successfully before you load any data into the HCM Data Mart.

## Auxiliary Files and Information Sources

### *Overview of Auxiliary Files*

In addition to getting their main input from detail data store tables, some of the jobs that load data into the HCM Data Mart get key pieces of information from certain auxiliary files and information sources. These include the properties files, the prebuild.sas macro, and the following information sources:

- SAS_DEFAULT_PROPERTIES MySQL table
- HCM Formats
- HCM Measures

Before you run any jobs that load data into the HCM Data Mart, make sure that all these auxiliary files and sources contain the correct information.

### *Editing the Properties Files*

There is a pair of properties files for each supported locale. One member of each pair contains column labels. The other member contains report and table titles. You might want to edit some of the text in the pair of files that corresponds to your site's designated locale (see "The Locale for SAS Human Capital Management" on page 266).

To edit properties files:

1. Find the files.

   The properties files reside at the following location: **`!SASROOT\hrds\sasmisc`** (Windows) or **`!SASROOT\misc\hrds`** (UNIX).

2. Make a backup copy of any properties file that you plan to edit, giving the copy a different name.
3. Edit the file with any text editor.

### *Editing the SAS Macro Files*

To edit any one of the SAS macro files for SAS Human Capital Management:

1. Find the file.

The SAS macro files reside at the following location: **`!SASROOT\hrds\sasmacro`** (Windows) or **`!SASROOT\sasautos`** (UNIX).

2. Copy the file to the designated override location. SAS Human Capital Management will use that copy instead of the file at the original location.

   The HCM designated override location for the SAS macro files is typically: ***`SAS-config-dir`*****`\Lev1\SASApp\SASEnvironment\HumanCapitalManagement\SASMacro`**.

3. Edit the copy of the file at the override location with any text editor.

## SAS_DEFAULT_PROPERTIES MySQL Table

The SAS_DEFAULT_PROPERTIES table in the HCM Data Mart contains properties such as the locale code and graph settings. For more information, see the Administration Application part of this book.

## Editing the HCM Formats

### Overview of Editing Formats

To edit the formats, take the following steps:

1. Log on to SAS Human Capital Management as an HCM Administrator.
2. Select the **Administration** link in the **Manage** category.
3. On the **Data** tab, select the **Formats** folder.
4. Follow the instructions in the online Help to edit formats or create new formats.

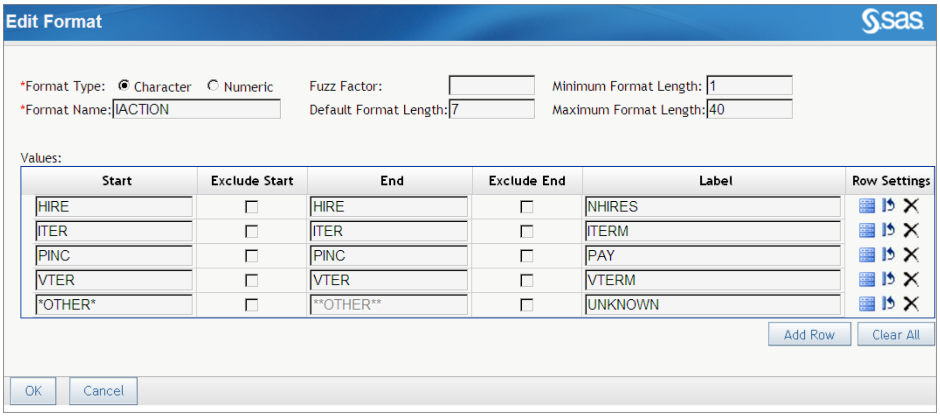
### Internal Formats

Make sure that all the internal formats are appropriate for your site. The list of internal formats follows:

- IETHNIC
- ICHURN
- IEEOCL
- IACTION
- IEMPSTA
- IEMPTYP
- IEXEMPT
- IGENDER
- IMNSTAT
- IONPYRL
- IPAYPER
- IREGTMP
- ISTECLS
- ITERM

Each format entry has a **Label** as well as a range that is represented by **Start** and **End** values. If the range consists of a single value, the **Start** and **End** values are the same.

The **Label** value corresponds to a keyword in the SAS Human Capital Management software. Here is an example of the IACTION format:



*Format Type: ● Character ○ Numeric

*Format Name: IACTION

Fuzz Factor:

Default Format Length: 7

Minimum Format Length: 1

Maximum Format Length: 40

Values:

| Start | Exclude Start | End | Exclude End | Label | Row Settings |
|---|---|---|---|---|---|
| HIRE | □ | HIRE | □ | NHIRES | |
| ITER | □ | ITER | □ | ITERM | |
| PINC | □ | PINC | □ | PAY | |
| VTER | □ | VTER | □ | VTERM | |
| *OTHER* | □ | **OTHER** | □ | UNKNOWN | |

The ITERM label represents involuntary termination. By default it has a value of `ITER`. If your site's data uses the value `XYZ` to signify involuntary terminations, then you must change the ITERM range so that both the **Start** and **End** values contain `XYZ`.

*Note:* Remember to change the **Start** and **End** values for internal formats. Do not change the **Label** values; these are the keywords that are understood by the SAS Human Capital Management software.

"Internal Formats" on page 329 lists the internal formats and their keywords.

### Display Formats

In addition to adjusting the internal formats, modify display formats as appropriate for your data.

The following formats are used in the HCM Build or are used for display. They can be modified based on the site's requirements:

- AGERNG
- LOS
- RNGEFMT
- EVALRES
- YESNO

Most of the display formats are maintained via detail data store ETL jobs or SAS Human Capital Management ETL jobs. Do not modify those formats, which are listed below.

*Note:* Some of the listed formats might not exist in SAS Human Capital Management, depending on the ETL jobs that have been run.

- ACADEMICCREDIT
- ACADEMICHONORS
- ACTION
- ACTRSN
- APPST
- ATTENDANCESTATUS

- AYN
- COMPCATALOG
- COMPCATEGORY
- COMPCLASS
- COMPETENCYTYPE
- COMPHIER
- COMPSPREF
- COMPTYP
- COUNTRY
- COURSELEVEL
- DEGREECONCENTRATION
- DEGREEOPTION
- DEGREEPROGRAM
- DEGREETYPE
- EDUVALUESYSTEM
- EDUVALUETYPE
- EEOCL
- EMPSTAT
- EMPTYPE
- ENROLLSTATUS
- ETHNIC
- EVIDENCETYPE
- EXEMPT
- FICE
- FLSA
- GENDER
- GRADUATINGDEGREE
- GRECTYP
- GRPnnFM format (for example, GRP11FM)
- HONORSPROGRAM
- INTORG
- IORGS
- JOBGRP
- LANG
- LVCODE
- MARITAL
- MONEY

- OTHERHONORS
- PAYPER
- POSTY
- PSTAT
- RECSRC
- REGTEMP
- REJRSN
- SCHOOLDEPTTYPE
- SCHOOLNAMETYPE
- SCHOOLTYPE
- STATE
- TAXCOMPCLASS
- TAXCOMPSPREF
- TAXONOMY
- TAXONOMYCATALOG
- UNION
- VETERAN
- VTGROUP
- WEIGHTTYPE

Any other formats that are in SAS Human Capital Management, but are not listed here, should not be modified.

### *Editing the PREBUILD.SAS Macro File*

To edit this file, follow the procedure described in “Editing the SAS Macro Files” on page 274.

In the prebuild.sas macro file, make sure that all the macro variables are assigned values that are appropriate for your site. You can edit the assignment statements that you find in this file. This is a better approach:

1. Create a section near the end of the file that contains copies of the code segments that you want to change.
2. Edit the assignment statements that you have copied into this new section.

The value that is assigned to a macro variable in the new section overrides the value that is assigned to it earlier in the file. By keeping all the edited assignment statements in one place, you make it easier to review them as necessary and to transfer them to a new version of the prebuild.sas file after you install a new version of the software.

For descriptions of all the macro variables in the prebuild.sas macro file, see “Macro Variables in the PREBUILD.SAS Macro File” on page 304.

### Editing HCM Measures

Predefined measure formulas can be modified, or new site-defined measures can be added to SAS Human Capital Management. To modify or add a measure, take the following steps:

1. Log on to SAS Human Capital Management as an HCM Administrator.
2. Click **Administration** .
3. Select the HR Measures folder.
4. Read the online Help for instructions on modifying an existing measure or adding a new measure.

Predefined factors, site-defined factors, and other measures can be used to define a new measure or modify an existing measure. Site factors are defined in the FACTORSITE table.

For a discussion of the FACTORSITE table, see "The FACTORSITE Table" on page 285.

## Moving Data from the Detail Data Store to the HCM Data Mart

### Overview of Moving Data from the Detail Data Store to the HCM Data Mart

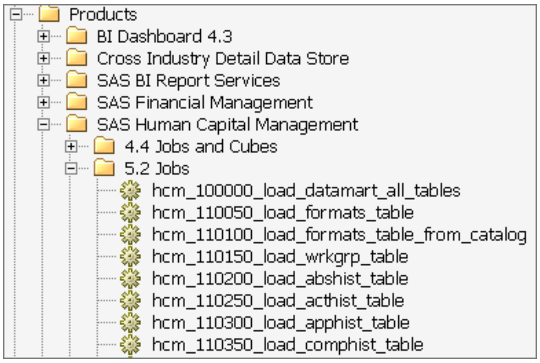
Before you load any data into the HCM Data Mart, you might need to do the following:

- Load certain data into the SDM, as described in "Moving Data from the Detail Data Store to the SDM" on page 273.
- Edit certain auxiliary files and information sources, as described in "Auxiliary Files and Information Sources" on page 274.

The HCM Data Mart contains four types of tables:

- *Detail tables* contain all the data that is in the HCM-specific detail data store tables, but it is organized in a different way. Data from the HCM-specific detail data store tables flows into the detail tables first.
- *Master tables* combine the data from the detail tables in various ways and also include additional columns that contain calculated values. These tables are optimized for reporting and querying.
- *Summary tables* contain summarized data from the master tables, with less historical detail.
- *Cubes* contain data from the master and summary tables that is reorganized into a multidimensional structure. HCM cubes can be viewed by means of information maps, using SAS Web Report Studio.

Typically, all of the SAS Data Integration Studio jobs that have either output or input in the HCM Data Mart are located in **Products** ⇨ **SAS Human Capital Management** ⇨ **Jobs** metadata folder on the **Folders** tab.



```
Products
  BI Dashboard 4.3
  Cross Industry Detail Data Store
  SAS BI Report Services
  SAS Financial Management
  SAS Human Capital Management
    4.4 Jobs and Cubes
    5.2 Jobs
      hcm_100000_load_datamart_all_tables
      hcm_110050_load_formats_table
      hcm_110100_load_formats_table_from_catalog
      hcm_110150_load_wrkgrp_table
      hcm_110200_load_abshist_table
      hcm_110250_load_acthist_table
      hcm_110300_load_apphist_table
      hcm_110350_load_comphist_table
```

The **Jobs** folder contains the following types of jobs:

- Detail table jobs consist of the jobs that load data from the HCM-specific detail data store tables into the detail tables. These jobs are labeled with numbers that start with 110 and 118.
- Master and summary table jobs consist of the jobs that build master tables and summary tables, using the data in the detail tables. These jobs are labeled with numbers that start with 120, 125, and 126.
- Cube jobs consist of the jobs that build the cubes. These jobs are labeled with numbers that start with 200 and 210.
- Metrics jobs consist of the jobs that compute HCM metrics and load them into HCM and to the SDM. These jobs are labeled with numbers that start with 128.
- Retention analytics jobs consist of the jobs that build the retention analysis tables. These jobs are labeled with numbers that start with 140.
- Other jobs consist of several special-purpose jobs.

There are three ways to run these jobs:

- Run the jobs one at a time in the order that is indicated by their numbers.

  See "Loading the HCM Data Mart by Running One Job at a Time" on page 281.
- Run the hcm_100000_load_datamart_all_tables job. This umbrella job invokes all the necessary jobs, except hcm_300000_create_information_maps, in the proper order. Not all available HCM jobs are executed by this job. After the umbrella job runs successfully, run the hcm_300000_create_information_maps job.

  See "Loading the HCM Data Mart by Running the Umbrella Job" on page 282 and "Creating the HCM Information Maps" on page 293.
- Run individual jobs as needed.

  After you fully load the HCM Data Mart, you can go back and rerun individual jobs. Rerunning a job can produce a different result if you first change some aspect of the job's environment, such as the value of a job option, the value of a relevant macro variable, or the data in a detail data store table that holds the job's input.

Running the umbrella job is convenient. However, it is advisable to run the jobs one at a time initially, until you are confident that each job runs without errors.

## *Loading the HCM Data Mart by Running One Job at a Time*

### *Run the Detail, Master, and Summary Table Jobs*

To load the HCM Data Mart by running one job at a time, run the jobs in the order that is specified by these topics. Run the jobs specified in the following topics in the **Detail Table Jobs** folder in the order that is indicated by their numbers. Run jobs 110*xxx* through 126*xxx*. A job should be run only if there is data available to load the target table for that job.

### *Create HCM Measures and Metrics*

Run one or more of the jobs whose numbers begin with 128. For a detailed discussion of HCM metrics, see "Loading HCM Metrics into a Metric Table" on page 291.

### *Run the hcm_129990_load_sas_hierarchy_mapping_table Job*

Run the hcm_129990_load_sas_hierarchy_mapping_table job.

### *Run the Cube Jobs*

Cube jobs process the data in the master and summary tables and use it to create the cubes. Run the jobs with numbers that begin with 200 or 210. The hcm_200000_create_all_cubes job can be run in the place of the individual cube jobs after the individual cube jobs have been successfully run. The individual cube jobs can be run in any order.

The first time you run the cube jobs, edit the %PREBUILD macro to set this option:

```
%let cube_delete_type=DELETE
```

After you run the cube jobs for the first time, edit the macro again and set the delete type to the default value:

```
%let cube_delete_type=DELETE_PHYSICAL
```

For more information about the %PREBUILD macro, see "Editing the PREBUILD.SAS Macro File" on page 278.

### *Run the Retention Analysis Jobs*

Retention Analysis Jobs are the jobs whose numbers start with 140. These are optional jobs for the Retention Analysis, which is a separately available analysis for SAS Human Capital Management.

### *Create HCM Information Maps*

After you load all the HCM data, you must create the HCM information maps. For details, see "Creating the HCM Information Maps" on page 293 .

### *Refresh the Cache*

Run the job hcm_900000_refresh_cache. This job refreshes the HCM Cache, which is associated with HCM Web Application. The job should be run any time that one or more other jobs have been run.

## Loading the HCM Data Mart by Running the Umbrella Job

### Overview of Loading the HCM Data Mart by Running the Umbrella Job

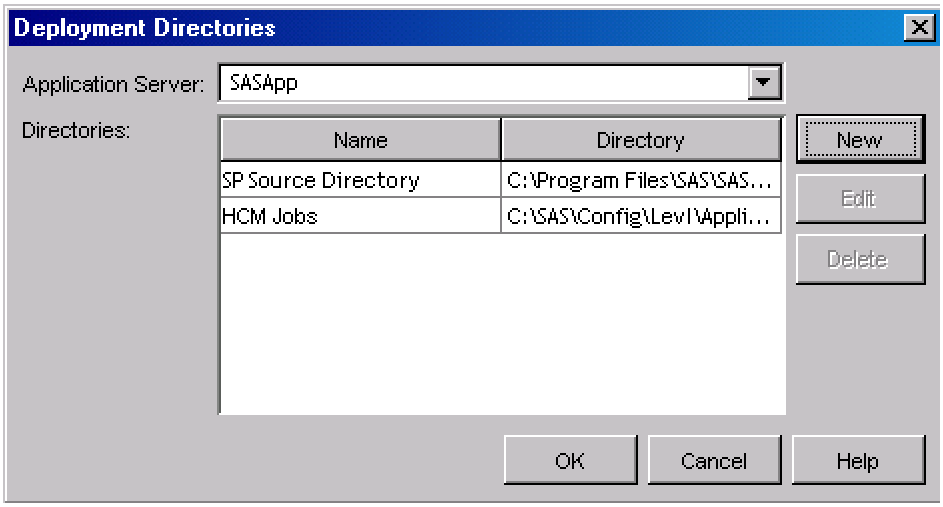
If you run the hcm_100000_load_datamart_all_tables umbrella job, then there is an additional preparatory step: you must deploy for scheduling all the jobs that are under the umbrella.

### Set Up a Deployment Directory

Before you can deploy the jobs for scheduling, a deployment directory must be set up. To ensure that you have a suitable deployment directory, do the following:

1. Launch SAS Management Console.
2. Under **Environment Management**, right-click **Schedule Manager** and select **Deployment Directories** from the pop-up menu.

   The Deployment Directories window appears.
3. In the **Application Server** field of the Deployment Directories window, select **SASApp** or **SASMain**. Only one of these application servers will be listed.



Application Server: SASApp

Directories:

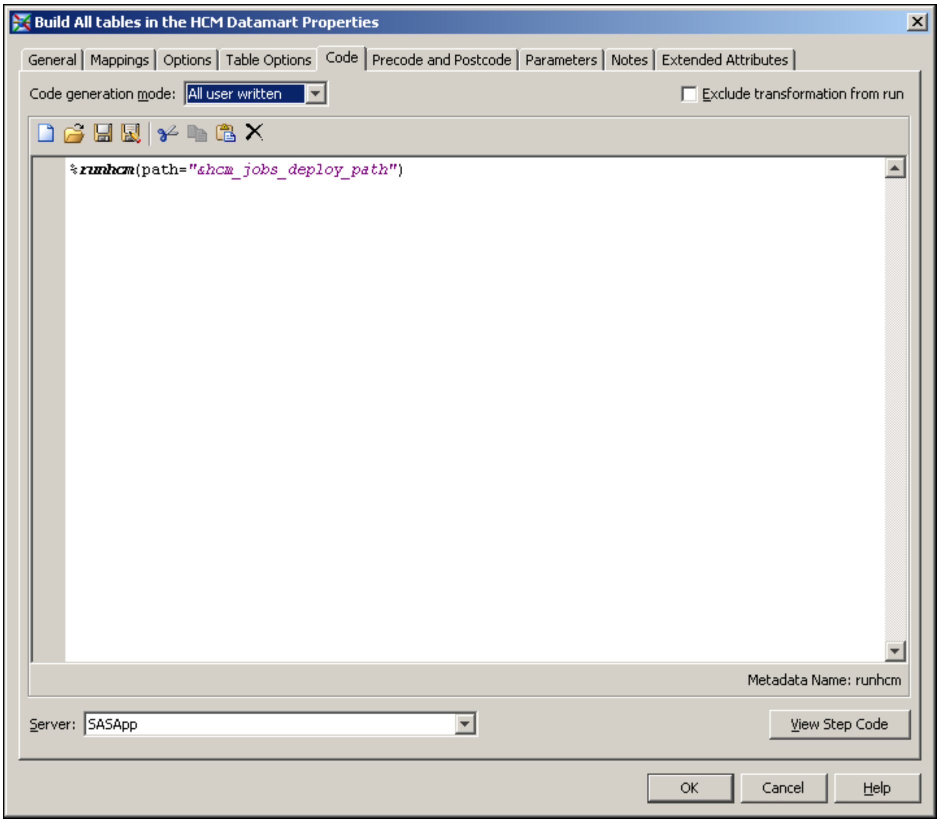
| Name | Directory |
|---|---|
| SP Source Directory | C:\Program Files\SAS\SAS... |
| HCM Jobs | C:\SAS\Config\Lev1\Appli... |

4. Check for the existence of a directory named HCM Jobs. If the HCM Jobs directory does not exist, click **New**. In the New Directory window, name the directory HCM Jobs. For the directory path, click **Browse** to select the following path: ***SAS-config-dir*\Lev1\Applications\SASHumanCapitalManagement5.2\SASCode\Jobs\Deploy**. This is the default directory path for deploying HCM jobs. If a different path is used, change the path that is specified in the Umbrella Job to this path.
5. Give the SAS Server Users group full access to this directory.

To view or change the path that is specified in the hcm_100000_load_datamart_all_tables job:

1. On the **Folders** tab, double-click the hcm_100000_load_datamart_all_tables umbrella job in the **Products** ⇨ **SAS Human Capital Management** ⇨ **Jobs** folder.
2. In the job window, double-click the transformation. The transformation properties window appears.

3. In the transformation properties window, select the **Code** tab to display the path.



Build All tables in the HCM Datamart Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
Code generation mode: All user written
□ Exclude transformation from run
Metadata Name: runhcm
Server: SASApp
View Step Code
OK
Cancel
Help

```
%runhcm(path="&hcm_jobs_deploy_path")
```

The path is specified by default as

```
&hcm_jobs_deploy_path
```

This path references ***`SAS-config-dir`*`\Lev1\Applications \SASHumanCapitalManagement5.2\SASCode\Jobs\Deploy`**. If the HCM Jobs deployment directory path is different from this path, then replace this path with the correct HCM Jobs path.

```
&hcm_jobs_deploy_path
```

When you deploy a job for scheduling, a corresponding .sas file that contains information about the job is created. Any time you modify a job, you must deploy the job again, so that the corresponding .sas file correctly represents the modified job.

### *Deploy the Jobs for Scheduling*

Before you deploy any job for scheduling, check that you have set all the job options correctly.

To deploy the jobs for scheduling:

1. In SAS Data Integration Studio, on the **Folders** tab, select every HCM job in the **Products** ⇨ **SAS Human Capital Management** ⇨ **Jobs** folder. Click the first job in the folder, and then click the last job in the folder while holding down the SHIFT key. (There are several jobs here that you do not need to deploy for scheduling, but it does no harm to deploy them all.)

2. Press the right mouse button and select **Scheduling** ⇨ **Deploy** from the pop-up menu. The **Deploy a job for scheduling** window appears.

3. Select SASApp – SAS Data Step Batch Server or **SASMain – SAS Data Step Batch Server** in the **Batch Server** field and **HCM Jobs** in the **Deployment Directory** field, and then click **OK**.

*Note:* If a Data Step Batch Server is not available for SASApp or SASMain, then a Data Step Batch Server Component must be set up for that application server before you can deploy jobs to the HCM Jobs directory.

### How the Umbrella Job Works

The hcm_100000_load_datamart_all_tables umbrella job executes the predefined runhcm.sas file. This file contains a list of .sas files that are created by deploying the jobs to the HCM Jobs directory.

```
%macro runhcm(path="&hcm_jobs_deploy_path");
options source source2;
/*  Set the path to the HCM Load Code.  */
filename HCMCODE &path
%include HCMCODE(hcm_110050_load_formats_table.sas,
                 hcm_110150_load_wrkgrp_table.sas,
                 hcm_110200_load_abshist_table.sas,
                 hcm_110250_load_acthist_table.sas,
                 hcm_110300_load_apphist_table.sas,
                 hcm_110350_load_comphist_table.sas,
                 hcm_110400_load_empgen_table.sas,
                 hcm_110450_load_useremployee_table.sas,
                 hcm_110500_load_grade_table.sas,
                 hcm_110550_load_jobs_table.sas,
                 hcm_110600_load_openpos_table.sas,
                 hcm_110650_load_pos_table.sas,
                 hcm_110700_load_cgrade_table.sas,
                 hcm_110750_load_cjobs_table.sas,
                 hcm_110800_load_cwrkgrp_table.sas,
                 hcm_120050_load_abshmast_table.sas,
                 hcm_120100_load_acthmast_table.sas,
                 hcm_120150_load_apphmast_table.sas,
                 hcm_120200_load_churn_table.sas,
                 hcm_120250_load_empmast_table.sas,
                 hcm_120300_load_oposmast_table.sas,
                 hcm_120350_load_termmast_table.sas,
                 hcm_125050_load_headsum_summary_table.sas,
                 hcm_125100_load_opossum_summary_table.sas,
                 hcm_125150_load_salhist_summary_table.sas,
                 hcm_125200_load_salhsum_summary_table.sas,
                 hcm_125250_load_tip_summary_table.sas,
                 hcm_128050_load_sas_measures_table.sas,
                 hcm_129990_load_sas_hierarchy_mapping_table.sas,
                 hcm_210050_create_abshcube_cube.sas,
                 hcm_210100_create_acthcube_cube.sas,
                 hcm_210150_create_apphcube_cube.sas,
                 hcm_210200_create_empcube_cube.sas,
                 hcm_210250_create_hdsmcube_cube.sas,
                 hcm_210300_create_oposcube_cube.sas,
                 hcm_210350_create_salhcube_cube.sas,
                 hcm_210400_create_termcube_cube.sas,
                 hcm_210450_create_tipcube_cube.sas,
                 hcm_210500_create_opsmcube_cube.sas,
                 hcm_900000_refresh_cache.sas
                );
%mend;
```

To run the hcm_100000_load_datamart_all_tables job is to execute all the jobs that are represented in this list, in the listed order. You can edit the runhcm.sas macro and modify this list to change the jobs that are executed when running the hcm_100000_load_datamart_all_tables job.

After you load all HCM data, you must create the HCM information maps. For details, see "Creating the HCM Information Maps" on page 293.

## Loading Certain HCM Data Mart Tables Directly

### Overview of Loading Certain HCM Data Mart Tables Directly

The following tables in the HCM Data Mart must be loaded directly, by means of jobs that you write:

- FACTORSITE
- HCBNCHMRK

There is no route to these tables through the detail data store, and there are no predefined jobs for loading them.

### The FACTORSITE Table

The FACTORSITE table is designed to contain values of certain factors that occur in the formulas that are used to calculate HCM metrics.

Typically, the following factors are extracted and populated from the HCM Data Mart when the HCM metrics are calculated or recalculated:

- Comp
- CompCont
- CompExec
- CompMgr
- CompReg
- CompStaff
- CompVar
- CompWFOnP
- FTE
- FTECont
- FTEContWFOffP
- FTEContWFOnP
- FTEEx
- FTENonex
- FTEReg
- FTERegWFOffP
- FTERegWFOnP

- FTEWFOffP
- FTEWFOnP
- HC
- HCCont
- HCContEx
- HCContNonex
- HCContWFOffP
- HCContWFOnP
- HCReg
- HCRegEx
- HCRegNonex
- HCRegWFOffP
- HCRegWFOnP
- HCWFOffP
- HCWFOnP
- Hires
- HiresEx
- HiresExec
- HiresMgr
- HiresNonex
- HiresStaff
- Seps
- SepsEx
- SepsInvol
- SepsInvolEx
- SepsInvolNonex
- SepsNonex
- SepsVol
- SepsVol0to1
- SepsVol10plus
- SepsVol1to3
- SepsVol3to5
- SepsVol5to10
- SepsVolEx
- SepsVolEx0to1
- SepsVolEx10plus
- SepsVolEx1to3

- SepsVolEx3to5
- SepsVolEx5to10
- SepsVolNonex
- SepsVolNonex0to1
- SepsVolNonex10plus
- SepsVolNonex1to3
- SepsVolNonex3to5
- SepsVolNonex5to10
- Tenure
- TenureEx
- TenureNonex
- Xfrs
- XfrsEx
- XfrsNonex

Depending on the jobs that are run and the data that is available, not all these factors might be populated. Because these factors are extracted and populated from the HCM Data Mart, they should not be added to the FACTORSITE table. However, many factors that are used by the predefined metric formulas are not populated from the HCM Data Mart. In addition, a site can create new metrics that reference these factors. To populate these factors, the factors must be loaded directly into the FACTORSITE table. In addition, new factors can be loaded, as required, that are not referenced by existing metrics, and then new measures can be created to incorporate those factors in their metric formulas.

You can use SAS Human Capital Management without loading any data into this table. If the FACTORSITE table is empty, then any formula that contains a factor whose value is not populated is ignored. In that case, the metric that depends on the factor is not computed. This is the only consequence of leaving the FACTORSITE table empty.

The FACTORSITE table has the following columns:

1. VALID_DT contains the date. The date format can vary based on the MySQL installed environment.
2. MEASURE_NM, with format varchar(100), contains the names of parameters, or factors, that are used in formulas for computing metrics. The names in this table must follow SAS naming conventions, and they must match exactly the names of the factors that are used in metric formulas, predefined or site-defined, that are populated from the HCM Data Mart. Include any factor names that occur in formulas.
3. MEASURE_VALUE_NO, with format decimal(15,2), contains the numeric values of factors to two decimal places.
4. MEASURE_DESC, with format varchar(100), contains descriptions of the parameters. The information in this column is not used by the software.

In each row of the FACTORSITE table, the VALID_DT value designates the first day of the time period to which the MEASURE_VALUE_NO value applies. Pairs of successive VALID_DT values from pairs of records that have the same MEASURE_NM value are used to correctly associate the MEASURE_VALUE_NO values with other time-dependent data values.

For any site-defined factors that are added to the FACTORSITE table, a factor with the same name must exist in the SAS_MEASURE_FORMULAS table, so that Metric Formulas can reference those factors. If SAS_MEASURE_FORMULAS does not contain a factor that has been added to FACTORSITE, then the factor must be added. Run the following SQL code for each factor that needs to be added to SAS_MEASURE_FORMULAS:

```
Insert into hcm.SAS_MEASURE_FORMULAS (VARIABLE, SAS_LABEL, DESCRIPTION,
SAS_FORMAT) Values ('FactorName1','Factor1 Label','Factor1 Description','');
```

Replace *FactorName1* with the factor name. Replace *Factor1 Label* with the factor label. Replace *Factor1 Description* with the factor description.

### The HCBNCHMRK Table

The HCBNCHMRK table contains Saratoga benchmark data that is available from the Saratoga Institute.

You can use SAS Human Capital Management without loading any data into this table. If the HCBNCHMRK table is empty, then the metric tables that you load from SAS Human Capital Management will not contain any Saratoga benchmark values. The metric table columns whose purpose is to hold Saratoga benchmark values will be empty. This is the only consequence of leaving the HCBNCHMRK table empty.

The column names of the HCBNCHMRK table are identical to the names of corresponding columns in the table of benchmark data that the Saratoga Institute delivers. Write a job that loads each of the following columns from the identically named column in the Saratoga file:

1. ELEMENT_BOOK_NM, with format varchar(255), contains measure names, in the terminology of SAS Solutions Services. Every name that is used by the Saratoga institute is provided in the SAS_MEASURE table.
2. ELEMENT_ID, with format varchar(11), contains Saratoga IDs.
3. FORMAT_DS, with format decimal(10,2), contains Saratoga formats.
4. MEAN_VAL, with format decimal(10,2), contains mean values.
5. MEDIAN_VAL, with format decimal(10,2), contains median values.
6. P10_VAL, with format decimal(10,2), contains tenth percentile values.
7. P25_VAL, with format decimal(10,2), contains twenty-fifth percentile values.
8. P75_VAL, with format decimal(10,2), contains seventy-fifth percentile values.
9. P90_VAL, with format decimal(10,2), contains ninetieth percentile values.
10. PERIOD_DS, with format varchar(100), contains period descriptions.
11. PERIOD_YEAR_NUM, with format decimal(10,0), contains 4-digit year values.
12. SECTION_DS, with format varchar(100), contains measure categories as defined by the Saratoga Institute.
13. STAT_DS, with format varchar(100), contains industry types as defined by the Saratoga Institute. These industry types are the possible values of the STGACUT macro variable. See "Macro Variables in the PREBUILD.SAS Macro File" on page 304.
14. STAT_TYPE_DS, with format varchar(30), contains demographic cuts as defined by the Saratoga Institute. These demographic cuts are the possible values of the STGACUTA macro variable. See "Macro Variables in the PREBUILD.SAS Macro File" on page 304.

## *Summary of Data Flow into HCM Data Mart Tables*

The following table shows the input-output relationships between HCM tables. All the tables in the Output column are MySQL tables in the HCM Data Mart. In the Source or Sources column, CIND_DDS indicates SAS tables in the CrossIndustryDDS library and HCMDATA indicates MySQL tables in the HCM Data Mart.

| Output | Source or Sources |
|---|---|
| FORMATS | CIND_DDS(MARITAL_STATUS, GENDER, EMPLOYEE_STATUS, EMPLOYEE_TYPE, ETHNICITY, LANGUAGE, COUNTRY, ABSENCE_TYPE, COMPENSATION_TYPE, STATE_REGION, ACTION_TYPE, ACTION_REASON, RECRUITMENT_SOURCE, REJECTION_REASON, JOB_GROUP, EEO_CLASS, FLSA_STATUS, EXEMPT_STATUS, CURRENCY, APPLICATION_STATUS, TIME_FREQUENCY, MILITARY_EXPERIENCE, EMPLOYEE_UNION, POSITION_STATUS, POSITION_PERMANENCE, PAY_LEVEL_STRUCTURE, ACADEMIC_CREDIT, ACADEMIC_HONORS, ATTENDANCE_STATUS, COURSE_LEVEL, DEGREE_CONCENTRATION, DEGREE_OPTION, DEGREE_PROGRAM, DEGREE_TYPE, EDUCATION_VALUE_SYSTEM, EDUCATION_VALUE_TYPE, ENROLLMENT_STATUS, FICE, GRADUATING_DEGREE, HONORS_PROGRAM, OTHER_HONORS, SCHOOL_DEPT_TYPE, SCHOOL_TYPE, SCHOOL_NAME_TYPE, SCHOOL_OR_INSTITUTION, SCHOOL_DEPT, COMPETENCY_CATALOG, COMPETENCY_CLASS, competency_type, COMPETENCY_CATEGORY, EVIDENCE_TYPE, SPECIAL_REF_COMP, TAXONOMY, TAXONOMY_CATALOG, TAXONOMY_COMP_CLASS, TAXONOMY_SPECIAL_REF_COMP, WEIGHT_TYPE, COMPETENCY_ASSOC_TYPE) |
| WRKGRP | CIND_DDS(INTERNAL_ORG_ASSOC, INTERNAL_ORG, EMPLOYEE, INTERNAL_ORG_NLS) |
| ABSHIST | CIND_DDS(EMPLOYEE_ABSENCE, EMPLOYEE_X) |

| Output | Source or Sources |
|---|---|
| ACTHIST | CIND_DDS(EMPLOYEE_ACTION, EMPLOYEE, EMPLOYEE_X_JOB, JOB_POSITION, EMPLOYEE_X_INTERNAL_ORG, INTERNAL_ORG_X, EMPLOYEE_X) |
| APPHIST | CIND_DDS(EMPLOYMENT_APPLICATION, EMPLOYEE_X) |
| COMPHIST | CIND_DDS(COMPENSATION, EMPLOYEE_X) |
| EMPGEN | CIND_DDS(EMPLOYEE, MILITARY_EXPERIENCE) |
| GRADE | CIND_DDS(PAY_LEVEL) |
| JOBS | CIND_DDS(JOB, JOB_GROUP) |
| OPENPOS | CIND_DDS(JOB_POSITION) |
| POS | CIND_DDS(JOB_POSITION, EMPLOYEE_X_JOB, EMPLOYEE_X, INTERNAL_ORG_X) |
| CGRADE | HCMDATA(GRADE) |
| CJOBS | HCMDATA(JOBS) |
| CWRKGRP | HCMDATA(WRKGRP) |
| ABSHMAST | HCMDATA(ABSHIST, ACTHIST, POS, JOBS, WRKGRP, EMPGEN) |
| APPHMAST | HCMDATA(APPHIST, POS, JOBS, WRKGRP) |
| ACTHMAST | HCMDATA(ACTHIST, POS, JOBS, WRKGRP, GRADE, EMPGEN) |
| EMPMAST | HCMDATA(ACTHIST, POS, JOBS, WRKGRP, GRADE, EMPGEN)<br>HCMDATA(OPENPOS - optional) |
| OPOSMAST | HCMDATA(OPENPOS, POS, JOBS, WRKGRP) |
| TERMMAST | HCMDATA(ACTHMAST) |
| SALHIST | HCMDATA(ACTHMAST) |
| SALHSUM | HCMDATA(COMPHIST, ACTHIST, POS, JOBS, WRKGRP, GRADE, EMPGEN) |
| OPOSSUM | HCMDATA(OPENPOS, POS, JOBS, WRKGRP) |
| HEADSUM | HCMDATA(ACTHIST, POS, JOBS, WRKGRP, GRADE, ACTHMAST, EMPGEN) |

| Output | Source or Sources |
|---|---|
| TIP | HCMDATA(ACTHMAST) |
| ABSHCUBE | HCMDATA(ABSHMAST) |
| ACTHCUBE | HCMDATA(ACTHMAST) |
| APPHCUBE | HCMDATA(APPHMAST) |
| EMPCUBE | HCMDATA(EMPMAST) |
| HDSMCUBE | HCMDATA(HEADSUM) |
| OPOSCUBE | HCMDATA(OPOSMAST) |
| SALHCUBE | HCMDATA(SALHSUM) |
| TERMCUBE | HCMDATA(TERMAST) |
| TIPCUBE | HCMDATA(TIP) |
| OPSMCUBE | HCMDATA(OPOSSUM) |

### *Loading SAS Human Capital Management Users*

The hcm_110450_load_useremployee_table job must be run in order to load SAS Human Capital Management users into the SAS_USER_EMPLOYEE table. Users cannot log on to SAS Human Capital Management until this job is run.

For this job, the User Names and their associated Employee IDs are extracted from the HCM EMPGEN table (columns USER_NM and EMPLOYEE_ID). The User Name extracted from EMPGEN is the same as the User Name populated in the User Manager plug-in of SAS Management Console. This is the User Name for the user that is specified on the General tab of the User Properties in SAS Management Console, not the Account User ID Login.

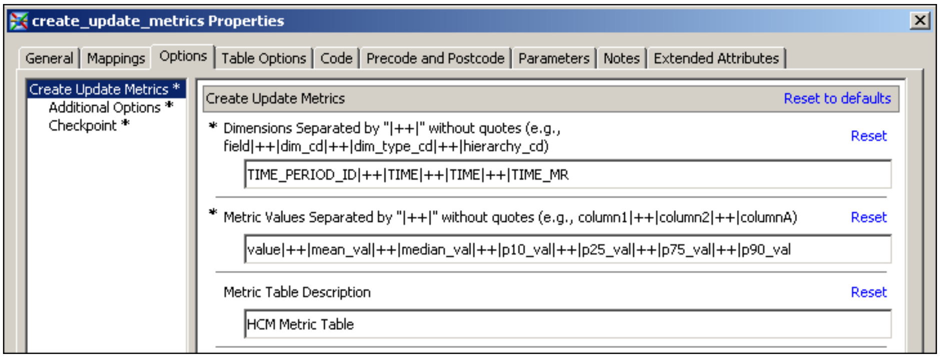
Only the users that are in the metadata group "HCM Solution Users" are extracted and loaded into SAS_USER_EMPLOYEE. You can specify a different metadata group by modifying the &hcmgroupname prebuild option.

## Loading HCM Metrics into a Metric Table

There are two HCM jobs that load HCM metrics into SAS Human Capital Management, and two optional jobs that load HCM metrics into the SASSDM:

- The hcm_128050_load_sas_measures_table job computes and loads HCM metrics, which vary by time. These metrics apply to the entire organization.
- The hcm_128100_load_sas_measures_table_with_org job computes and loads HCM metrics, which vary by time and department. These metrics apply to individual departments within the entire organization.

- The optional hcm_128900_load_sdm_metric_table job loads metrics, which vary with time, to the SASSDM. The create_update_metrics transformation of this job has the following options and default option values:



Create Update Metrics

* Dimensions Separated by "|++|" without quotes (e.g., field|++|dim_cd|++|dim_type_cd|++|hierarchy_cd)

TIME_PERIOD_ID|++|TIME|++|TIME|++|TIME_MR

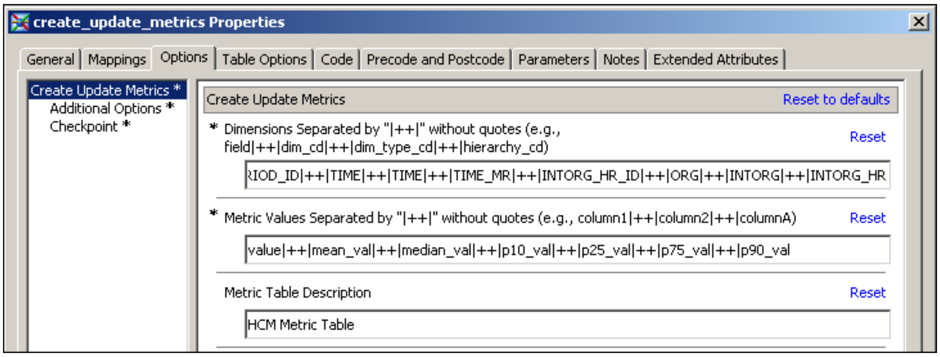
* Metric Values Separated by "|++|" without quotes (e.g., column1|++|column2|++|columnA)

value|++|mean_val|++|median_val|++|p10_val|++|p25_val|++|p75_val|++|p90_val

Metric Table Description

HCM Metric Table

- The optional hcm_128901_load_sdm_metric_table_with_org job loads metrics, which vary with time and organization, to the SASSDM. The create_update_metrics transformation of this job has the following options and default option values:



Create Update Metrics

* Dimensions Separated by "|++|" without quotes (e.g., field|++|dim_cd|++|dim_type_cd|++|hierarchy_cd)

RIOD_ID|++|TIME|++|TIME|++|TIME_MR|++|INTORG_HR_ID|++|ORG|++|INTORG|++|INTORG_HR

* Metric Values Separated by "|++|" without quotes (e.g., column1|++|column2|++|columnA)

value|++|mean_val|++|median_val|++|p10_val|++|p25_val|++|p75_val|++|p90_val

Metric Table Description

HCM Metric Table

For the two jobs that load metrics into SAS Human Capital Management, the hcm_128050_load_sas_measures_table job is included in the umbrella job by default. If required, the other job, hcm_128100_load_sas_measures_table_with_org, can be included in the umbrella job by editing the runhcm.sas macro.

The two jobs that load metrics to the SASSDM are optional. They should be run if SAS Strategy Management (StM) is installed, and the site wants to access the HCM metrics from StM. The jobs must be run after the HCM metrics jobs are run.

The two SASSDM metrics jobs have the same options as the solnsvc_3400_load_metric_table job. For a detailed discussion of these job options, see "Preparing Jobs to Load Metric Data" on page 259. For these two HCM jobs, the following additional points apply:

- For the Dimensions option, you must provide time values in the first job and time and organization values in the second job. You cannot omit these dimension types or include other dimension types.

  A dimension code that is used in a metric table must not be a MySQL reserved word. See "MySQL Reserved Words" on page 735.

- For the Metric Values option, do not make any changes. The computed metric values are loaded into the VALUE column, and the Saratoga benchmark values are loaded into the other specified columns (provided that you have loaded Saratoga benchmark values into the HCBNCHMRK table, as explained in "The HCBNCHMRK Table" on page 288).

- For the Metric Table Description option, do not make any changes.

There is an exception to this rule. If you run a job with one set of values for the Dimensions option and then change the Dimensions option values and run the job again, you should change the Metric Table Description too. This is because each set of values for the Dimensions option creates a different metric table, and each metric table should have a unique name so that users of scorecards can know which metric table to select.

The two metric jobs that load to the SDM are not part of the umbrella job by default. You can place one of the other jobs in the umbrella job by editing the runhcm.sas macro file. See "Loading the HCM Data Mart by Running the Umbrella Job" on page 282. However, you cannot have both of these jobs in the umbrella job at the same time.

In general, you should make a one-time choice to use one of these two jobs and ignore the other.

The four metrics jobs are affected by all of the following:

- The values of the following macro variables, which are set in the prebuild.sas macro file:

  - FCTRVARS
  - HEDUNIT
  - MEAS_LEV
  - PERIOD_TYPE_CD
  - SALUNIT
  - STGACUT
  - STGACUTA
  - STGAMAX

  The HEDUNIT, PERIOD_TYPE_CD, and SALUNIT macro variables must all specify the same unit of time. If you change the values of these three macro variables, then you must rerun the hcm_125050_load_headsum_summary_table and hcm_125200_load_salhsum_summary_table jobs before you run the metrics jobs.

  For a discussion of the prebuild.sas macro file, see "Editing the PREBUILD.SAS Macro File" on page 278. For details about all macro variables, see "Macro Variables in the PREBUILD.SAS Macro File" on page 304.

- The metric formulas defined in SAS Human Capital Management. For a list of all HCM measures and their metric formulas, log on to SAS Human Capital Management as an administrator.

  *Note:* A list of all measures is displayed, whether or not they are populated with data. These measures can be modified, and new measures can be added.

- The FACTORSITE table, which contains the numeric values of certain parameters that are used in metrics formulas. See "The FACTORSITE Table" on page 285.

## Creating the HCM Information Maps

The HCM information maps enable users to view the data in the SAS Human Capital Management master tables and cubes using SAS Web Report Studio. There is an information map for each master table and each cube.

To create the HCM information maps, run the hcm_300000_create_information_maps job.

# Chapter 24
# Modifying the Data Model for SAS Human Capital Management

**Overview of Modifying the HCM Data Model** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **295**

**Adding a Column** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **295**
Overview of Adding a Column . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 295
Adding Columns to Physical Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 296
Updating Table Metadata . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 297
Modifying Jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 298

**Changing the Character Length of a Column** . . . . . . . . . . . . . . . . . . . . . . . . . . . **298**

**Adding a Table** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **299**

**Adding a Cube** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **300**

## Overview of Modifying the HCM Data Model

You can modify the HCM data model in three main ways:

- adding columns to existing tables
- increasing the character length of existing columns
- adding whole new tables

## Adding a Column

### *Overview of Adding a Column*

Adding a column to the data model involves three primary activities:

1. adding columns to the physical tables that will hold the additional data
2. updating table metadata for those tables
3. modifying the relevant jobs to handle the additional data

You can accomplish these tasks in more than one way. Moreover, the details of what you must do can vary from case to case. Here is an outline of the process of adding a column.

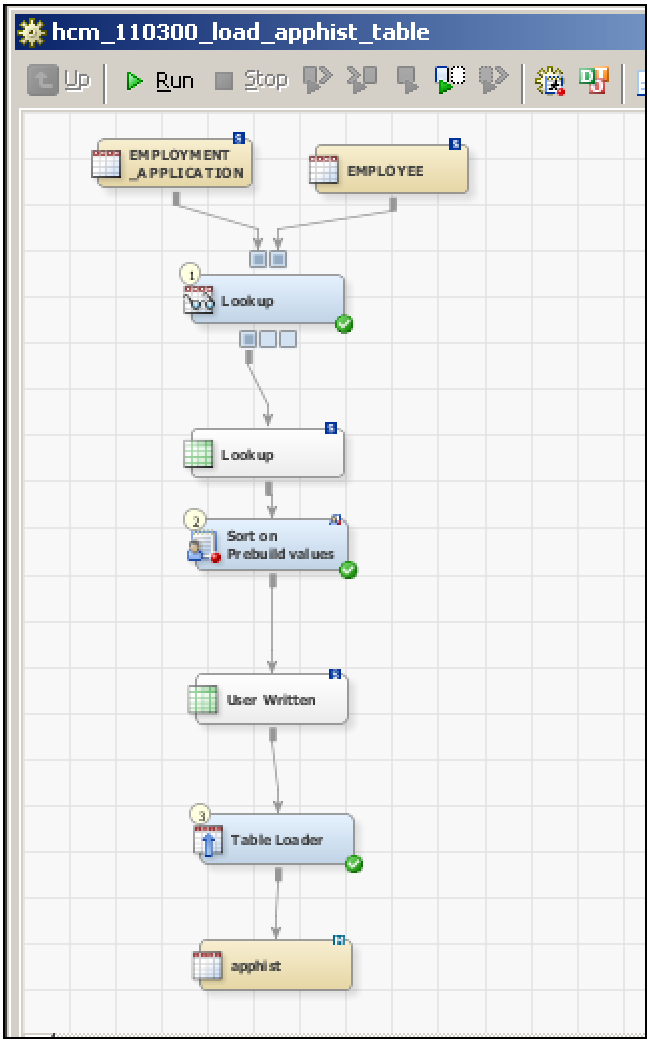
## Adding Columns to Physical Tables

In every case, you must add an appropriate column to the physical staging table and to the corresponding physical detail data store table. You can add these columns using SAS code that you write. Before you modify any physical table, make a backup copy.

Staging tables and detail data store tables are in the following locations: ***SAS-config-dir*\Lev1\SASApp\Data\SolutionsServices\stagedds** and ***SAS-config-dir*:\SAS\Config\Lev1\SASApp\Data\SolutionsServices\DDSData**.

For some detail data store tables, the DDSData directory also contains a table whose name is the name of the detail data store table followed by the characters _X. For example, the EMPLOYEE table is accompanied by the EMPLOYEE_X table. If you are adding a primary key column to such a detail data store table, then you must also add the column to the accompanying _X table.

In addition, you might need to add an appropriate column to the HCM Data Mart detail table (or tables) that the detail data store table feeds. This step is required for a given detail table if and only if the job that loads the table does not call user-written code. If the job that loads a given detail table calls user-written code, then the job automatically recreates that detail table to reflect all the columns in all the detail data store tables that feed it. For example, if you add columns that map into the ACTHIST detail table, then you do not have to add a column to ACTHIST because the job that loads ACTHIST calls user-written code:



To add a column, you can use MySQL Workbench, a script, or an interactive MySQL session

## *Updating Table Metadata*

In every case, you must update the metadata for the staging table and the corresponding detail data store table. The metadata for the detail data store table includes the metadata for the original table in the detail data store.

To update the metadata for the staging table, do the following:

1. On the **Folders** tab, select the table in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **StageDDS** folder.
2. Select **Update Metadata** from the pop-up menu.

To update the metadata for the original table in the detail data store, do the following:

1. On the **Folders** tab, select the table in the **Products** ⇨ **Cross Industry Detail Data Store** ⇨ **CrossIndustryDDS** folder.
2. Select **Update Metadata** from the pop-up menu.

In addition, you might have to update the metadata for the HCM Data Mart detail table (or tables) that the detail data store table feeds. This step is required for a given detail table if and only if the job that loads the table does not call user-written code. If the job that loads a given detail table calls user-written code, then the job automatically updates the metadata for that detail table. See the examples of jobs that do and do not call user-written code in "Adding Columns to Physical Tables" on page 296.

To update the metadata for an HCM Data Mart detail table, do the following:

1. On the **Folders** tab, select the table in the **Products** ⇨ **SAS Human Capital Management** ⇨ **Data Sources** ⇨ **HCMData** folder.
2. Select **Update Metadata** from the pop-up menu.

### *Modifying Jobs*

Whenever you add a column to a staging table and the corresponding detail data store table, there are several jobs that require attention:

- In the job that loads the staging table, you must add code to extract the data for the new column from the appropriate source system and load that data into the new column of the staging table.
- In the job that loads the detail data store table from the staging table, you must update the mappings in all transformations to handle the new column.
- In any job that loads an HCM Data Mart detail table that the detail data store table feeds, you might have to update the mappings in all transformations to handle the new column. This step is required for a given job if and only if the job does not call user-written code. If the job that loads a given detail table calls user-written code, then this job does not need to be modified. The user-written code has the flexibility to handle additional columns of data. See the examples of jobs that do and do not call user-written code in "Adding Columns to Physical Tables" on page 296.

## Changing the Character Length of a Column

The task of changing the character length of a column closely parallels the task of adding a column, which is outlined in "Adding a Column" on page 295. The differences are as follows:

- Instead of adding a column to each relevant physical table, you change the character length of an existing column in each relevant physical table.
- If you are changing the character length of a primary key column and the detail data store table has an accompanying _X table, then you must also change the character length of the corresponding column in the _X table.
- In the job that loads the detail data store table, you do not need to write code to extract new data from your source system.
- Instead of updating the mappings in transformations to include a new column, you must use the **Mapping** tab of each transformation to correctly specify the new column length.

# Adding a Table

You can add a table to the HCM Data Mart via the Administration application of SAS Human Capital Management. You can also use the Administration application to register a table that already exists in the HCM Data Mart. Follow these steps:

1. Log on to SAS Human Capital Management as an administrator. On the **Data** tab in the Administration application, select **Add Table**.

   Follow the online Help instructions for importing a table and registering a new table or an existing table. Be sure to select the option to register the table in metadata. Do not select the option to build the information map.

2. Modify the column attributes for the table:

   a. On the **Data** tab, select **Tables** ⇨ ***table-name*** to display the Column Attributes page for the table.

   b. Modify or define all column labels and all necessary formats. Follow the online Help instructions for this page for information about all column attributes.

3. Create the information map for the new table:

   a. On the **Data** tab, click **Tables**.

   b. Click the action menu ▾ to the left of the table name and select **Build Info Map** in the pop-up menu. You receive a message when the information map has been built.

4. After the changes are complete, click **Refresh Cache** in the menu bar.

If you want to load the newly added HCM Data Mart table through the detail data store, perform these tasks to create the channel for the data to flow through:

- Create a physical staging table and register its metadata in SAS Data Integration Studio.
- Create a physical detail data store table and register its metadata in SAS Data Integration Studio.
- Create a job to load the staging table.
- Create a job to load the detail data store table.
- Create a job to load the HCM Data Mart detail table from the detail data store table. Place a call to %PREBUILD in the precode for the job, and place a call to %UPDATEDD in the postcode for the job. Use other HCM jobs for an example of these precode and postcode changes.

*Note:* Some new tables in the HCM Data Mart might be populated only by data from existing detail data store tables. In that case, there is no need to create a new staging table or detail data store table.

## Adding a Cube

If you add a table to the HCM Data Mart as described in "Adding a Table" on page 299, then you might want to use that new table as a source table for a new cube. To add a new cube that receives data from a new table, do the following:

1. Make sure that you have defined all formats and column labels for the source table.

   This is done in the SAS Human Capital Management Administration application.

2. Log in to SAS Human Capital Management as an administrator.

3. On the **Data** tab in the Administration application, click **New Cube** in the menu bar.

   The New Cube wizard page is displayed. Follow the steps in the wizard, and click **Finish** to build the cube.

   The cube name can be up to 32 characters in length, and can contain only letters, numbers, and underscores. If you want to run an ETL job to refresh the cube, then the cube name must be uppercase. On the Dimensions page, the ORGDIM, MGRDIM, and GEODIM dimensions display whether those dimensions are available in the source table or not. Selecting them does not cause an error for a source table that doesn't have them. When you create a new dimension, multiple columns that are selected for the dimension provide a drillable hierarchy for that dimension.

4. Build an information map for the newly created cube.

5. On the **Data** tab in the Administration application, click the Cubes folder.

6. Click the menu icon for the new cube and select **Build Information Maps** from the pop-up menu.

   You will receive a message when the information map has been built.

If you want to create an ETL job so that the cube can be refreshed on a regular basis, then you must do the following:

1. Open SAS Data Integration Studio.

2. On the **Folders** tab, in the **Products** ⇨ **SAS Human Capital Management** ⇨ **Jobs** folder, create a job that loads data into the new cube from its source table.

   You can use the cube jobs that are already in this folder as a model by copying and renaming an existing cube job. For example, copy and rename the job hcm_210050_create_abshcube_cube. The cube jobs consist entirely of a User Written Code transformation, which is available on the **Transformations** tab in the Data group.

3. Open up the new job and double-click on the transformation in the job.

4. On the **Code** tab in the Properties window for the job, delete the existing code and add the following code

   In this example, the cube is named MYCUBE, and the source table is named MYTABLE:

   ```
   %cubegen(cubename=MYCUBE, baseds=&hcmlib..MYTABLE)
   ```

5. Click **OK** and close the job to save your changes.

6. Right-click the job and select **Properties**.

On the **Precode and Postcode** tab, make sure that the Precode box contains the following code:

```
%hcmlibol
%prebuild
```

7. Click **OK** to save any changes.

   The above steps to add a cube to SAS Human Capital Management need to be followed before you can run a job in Data Integration Studio to refresh the cube. Run this new job to refresh the cube. You can rerun this job whenever you need to load fresh data into the new cube. If you add a line for the new job to the umbrella job, then the new job runs whenever the umbrella job runs. For a detailed discussion of the umbrella job, see "Loading the HCM Data Mart by Running the Umbrella Job" on page 282.

If you run the hcm_300000_create_information_maps job, the information map for the new cube is rebuilt, along with all the other information maps in SAS Human Capital Management.

# *Chapter 25*
# Macro Variables in the PREBUILD.SAS Macro File

**Overview of SAS Human Capital Management Macro Variables** . . . . . . . . . . . . **304**

**Families of Table-Acronym Macro Variables** . . . . . . . . . . . . . . . . . . . . . . . . . . . **304**
ANAL Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 304
CUBED Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 305
CUBEV Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 306
CUTOFF Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 307
M Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 307
SORT Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 308
START Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 309
STAT Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 309
_UH Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 310
UNIT Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 311
V Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 312
VARS Family . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 312

**Individual Macro Variables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **313**
AGEFRACT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 313
CHRNDROP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 313
CHRNINDS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 313
CHRNVARS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 314
CUBE_DELETE_TYPE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 314
CUBE_DRILL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 314
CUBE_META_PATH . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 315
CUBEPATH . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 315
DEBUG . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 315
DIRECT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 315
EDU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 316
EMPPOP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 316
FCTRVARS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
FMT_UPDATE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
HCMGROUPNAME . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
HCMLIB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 317
HIERORDS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 318
HIERS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 318
INCOPEN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 318
INTORG_DIMENSION_CD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 319
MAXLEVL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 319
MEAS_LEV . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 319
MPRINT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 320
NOTES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 320
NUMBER_OF_HIERS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 321

ONPAYRL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 321
ORGMEMBER_EXCLUDES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 321
PERIOD_TYPE_CD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 322
POSDIR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 322
POSVAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 322
POSVDIR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 323
SALCVAR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 323
SOURCE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 324
SRVFRACT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 324
STGACUT and STGACUTA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 324
STGAMAX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 326

# Overview of SAS Human Capital Management Macro Variables

This chapter contains detailed information about the macro variables that you can use in the prebuild.sas macro file to configure the SAS Human Capital Management build process. These macro variables are presented in two groups:

- Families of macro variables whose names begin with table acronyms are in "Families of Table-Acronym Macro Variables" on page 304. They are alphabetized by the suffix that follows the table acronym.
- Individual macro variables are presented alphabetically in "Individual Macro Variables" on page 313.

# Families of Table-Acronym Macro Variables

## *ANAL Family*

### *Description*

Use the <TABLEACR>ANAL macro variables to specify additional analysis columns to include in a designated cube. An analysis column is a numeric column that provides input to statistical computations. For each cube, there is one analysis column that is included by default. The <TABLEACR>ANAL macro variables enable you to include others. The analysis columns that you specify must be present in the corresponding master table.

Compare the CUBEV family, which concerns non-analysis columns.

### *Syntax*

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSANAL: refers to the columns in the ABSHMAST table to be included in the ABSHCUBE table
- ACTANAL: refers to the columns in the ACTHMAST table to be included in the ACTHCUBE table

- APPANAL: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- EMPANAL: refers to the columns in the EMPMAST table to be included in the EMPCUBE table
- HDSMANAL: refers to the columns in the HEADSUM table to be included in the HDSMCUBE table
- OPOSANAL: refers to the columns in the OPOSMAST table to be included in the OPOSCUBE table
- OPMANAL: refers to the columns in the OPOSSUM table to be included in the OPSMCUBE table
- SALHANAL: refers to the columns in the SALHSUM table to be included in the SALHCUBE table
- TERMANAL: refers to the columns in the TERMMAST table to be included in the TERMCUBE table
- TIPANAL: refers to the columns in the TIP table to be included in the TIPCUBE table

**`column-1 <... column-n>`** are one or more analysis columns to include in the cube.

### Example

Include the SALARY column with the other analysis columns in the ACTHCUBE table:

```
%LET actanal=salary;
```

## CUBED Family

### Description

Use the <TABLEACR>CUBED macro variables to specify cube dimensions whose values should be displayed in descending order.

Do not specify the same column with a CUBED macro variable and the corresponding CUBEV macro variable.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSCUBED: refers to the columns in the ABSHMAST table to be included in the ABSHCUBE table
- ACTCUBED: refers to the columns in the ACTHMAST table to be included in the ACTHCUBE table
- APPCUBED: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- EMPCUBED: refers to the columns in the EMPMAST table to be included in the EMPCUBE table
- HDSMCUBED: refers to the columns in the HEADSUM table to be included in the HDSMCUBE table

- OPOSCUBED: refers to the columns in the OPOSMAST table to be included in the OPOSCUBE table
- OPMCUBED: refers to the columns in the OPOSSUM table to be included in the OPSMCUBE table
- SALHCUBED: to the columns in the SALHSUM table to be included in the SALHCUBE table
- TERMCUBED: refers to the columns in the TERMMAST table to be included in the TERMCUBE table
- TIPCUBED: refers to the columns in the TIP table to be included in the TIPCUBE table

**`column-1 <... column-n>`** are one or more columns to base the sort on.

### Example

In the ACTHCUBE table, sort the SALARY dimension in descending order:

```
%LET actcubed=salary;
```

## CUBEV Family

### Description

Use the <TABLEACR>CUBEV macro variables to specify non-analysis columns to include in a designated cube and display with the default ascending sort order. The columns that you specify are included in the designated cube, but they are not included in any hierarchy. These columns must be present in the corresponding master table.

Do not specify the same column with a CUBEV macro variable and the corresponding CUBED macro variable.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSCUBEV: refers to the columns in the ABSHMAST table to be included in the ABSHCUBE table
- ACTCUBEV: refers to the columns in the ACTHMAST table to be included in the ACTHCUBE table
- APPCUBEV: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- EMPCUBEV: refers to the columns in the EMPMAST table to be included in the EMPCUBE table
- HDSMCUBEV: refers to the columns in the HEADSUM table to be included in the HDSMCUBE table
- OPOSCUBEV: refers to the columns in the OPOSMAST table to be included in the OPOSCUBE table
- OPMCUBEV: refers to the columns in the OPOSSUM table to be included in the OPSMCUBE table
- SALHCUBEV: refers to the columns in the SALHSUM table to be included in the SALHCUBE table

- TERMCUBEV: refers to the columns in the TERMMAST table to be included in the TERMCUBE table

**`column-1 <... column-n>`** are one or more table columns that are to be included in the cube.

### *Example*

Include the AGE column from the ACTHMAST table in the ACTHCUBE column list:

```
%LET actcubev=age;
```

## *CUTOFF Family*

### *Description*

Use the <TABLEACR>CUTOFF macro variables to restrict the data in certain tables to records whose effective dates are later than the cutoff date that you specify.

### *Syntax*

```
%LET macro-variable=date;
```

**`macro-variable`** is one of the following:

- HDCUTOFF: refers to the HEADSUM table
- OPCUTOFF: refers to the OPOSSUM table
- PDCUTOFF: used for the Retention Analysis data extraction
- SLCUTOFF: refers to the SALHIST table

**`date`** is a past date in the format DDMMMYYYY, where DD is the number of the day of the month, MMM is the first three letters of the month, and YYYY is the year. This date cannot be February 29 of a leap year.

### *Example*

Restrict the HEADSUM table to records whose effective dates are later than January 1, 1995:

```
%LET hdcutoff=01JAN1995;
```

## *M Family*

### *Description*

Use the <TABLEACR>M macro variables to exclude specified detail table columns from the master tables that they would otherwise be included in. For each master table, there is a base detail table. The master table is formed by merging its base detail table with one or more other detail tables. You cannot exclude columns that come from the base detail table, but you can exclude columns that come from any other detail table that participates in the merge.

### *Syntax*

```
%LET macro-variable=DROP=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ACTHISTM: refers to columns in the ACTHIST detail table

- EMPEDUCM: refers to columns in the EMPEDUC detail table
- EMPGENM: refers to columns in the EMPGEN detail table
- GRADEM: refers to columns in the GRADES detail table
- JOBSM: refers to columns in the JOBS detail table
- POSM: refers to columns in the POS detail table
- WRKGRPM: refers to columns in the WRKGRP detail table

**`column-1 <... column-n>`** are one or more columns to exclude from the master tables.

### Example

Prevent the HOURLY_SALARY and MONTHLY_SALARY columns from the ACTHIST table from being included in any master tables that use ACTHIST as input (except the ACTHMAST table, where ACTHIST is the base table):

```
%LET acthistm=drop=hourly_salary monthly_salary;
```

## SORT Family

### Description

Use the <TABLEACR>SORT macro variables to specify columns that are used to determine the order of the records in the table.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSHSORT: refers to the ABSHIST table
- ABSMSORT: refers to the ABSHMAST table
- ACTHSORT: refers to the ACTHIST table
- ACTMSORT: refers to the ACTHMAST table
- APPHSORT: refers to the APPHIST table
- APPMSORT: refers to the APPHMAST table
- CMPSORT: refers to the COMPHIST table
- GRADSORT: refers to the GRADE table
- JOBSORT: refers to the JOBS table
- OPOSSORT: refers to the OPOSHIST table
- OPSMSORT: refers to the OPOSMAST table
- POSSORT: refers to the POS table

**`column-1 <... column-n>`** are one or more columns to sort on.

### Example

Specify that the ABSHIST table should be sorted by the EMPLOYEE_ID and ABSENCE_START_DT columns:

```
%LET abshsort=employee_id absence_start_dt;
```

The keyword DESCENDING can be placed in front of any column to specify a descending sort order for that column.

## START Family

### Description

Use the <TABLEACR>START macro variables to specify the position in time from which to begin summarizing certain tables.

### Syntax

```
%LET macro-variable=time-position;
```

**`macro-variable`** is one of the following:

- HEDSTART: refers to the HEADSUM table
- OPSSTART: refers to the OPOSSUM table
- SALSTART: refers to the SALHIST table

**`time-position`** is one of the following:

- BEGIN
- MIDDLE
- END

### Example

Summarize the HEADSUM table from the beginning of the time unit that is specified with the HEDUNIT macro variable:

```
%LET hedstart=begin;
```

## STAT Family

### Description

Use the <TABLEACR>STAT macro variables to specify additional statistical columns to include in a cube. A statistical column holds the output of statistical computations. For each cube, there is one statistical column that is included by default. The <TABLEACR>STAT macro variables enable you to include others.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSSTAT: refers to the columns in the ABSHMAST table to be included in the ABSHCUBE table
- ACTSTAT: refers to the columns in the ACTHMAST table to be included in the ACTHCUBE table
- APPSTAT: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- EMPSTAT: refers to the columns in the EMPMAST table to be included in the EMPCUBE table

- HDSMSTAT: refers to the columns in the HEADSUM table to be included in the HDSMCUBE table
- OPOSSTAT: refers to the columns in the OPOSMAST table to be included in the OPOSCUBE table
- OPMSTAT: refers to the columns in the OPOSSUM table to be included in the OPSMCUBE table
- SALHSTAT: refers to the columns in the SALHSUM table to be included in the SALHCUBE table
- TERMSTAT: refers to the columns in the TERMMAST table to be included in the TERMCUBE table
- TIPSTAT: refers to the columns in the TIP table to be included in the TIPCUBE table

**`column-1 <... column-n>`** are one or more statistical columns to include in the cube.

### Example

Include the N and NMISS statistical columns in the ACTHCUBE table:

```
%LET actstat=n nmiss;
```

## _UH Family

### Description

Use the <TABLEACR>_UH macro variables to specify user-defined hierarchy columns that you can include in a cube. You can number hierarchies sequentially (for example, ACT_UH, ACT_UH1, ACT_UH2).

### Syntax

```
%LET macro-variable=column-1<... column-n>/name="name-of-hierarchy";
```

**`macro-variable`** is one of the following:

- ABS_UH: refers to the columns in the ABSHMAST table to be included in the ABSHCUBE table
- ACT_UH: refers to the columns in the ACTHMAST table to be included in the ACTHCUBE table
- APP_UH: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- EMP_UH: refers to the columns in the APPHMAST table to be included in the APPHCUBE table
- HED_UH: refers to the columns in the HEADSUM table to be included in the HDSMCUBE table
- OPS_UH: refers to the columns in the OPOSMAST table to be included in the OPOSCUBE table
- OPM_UH: refers to the columns in the OPOSSUM table to be included in the OPSMCUBE table
- SALH_UH: refers to the columns in the SALHSUM table to be included in the SALHCUBE table

- TERM_UH: refers to the columns in the TERMMAST table to be included in the TERMCUBE table
- TIP_UH: refers to the columns in the TIP table to be included in the TIPCUBE table

**`column-1 <... column-n>`** are one or more visual hierarchy columns to include in the cube.

**`name-of-hierarchy`** is the user-defined name for the hierarchy.

### Example

Use the columns ACTCODE1, ACTCODE2, and ACTCODE3 to create a visual hierarchy named Action Codes in the ACTHCUBE cube:

```
%LET act_uh1=actcode1 actcode2 actcode3 / name="Action Codes";
```

## UNIT Family

### Description

Use the <TABLEACR>UNIT macro variables to specify the unit of time that is summarized in certain tables.

### Syntax

```
%LET macro-variable=time-unit;
```

**`macro-variable`** is one of the following:

- HEDUNIT: refers to the HEADSUM table
- OPSUNIT: refers to the OPOSSUM table
- PDUNIT: used for the Retention Analysis data extraction
- SALUNIT: refers to the SALHIST table

**`time-unit`** is one of the following:

- DAY
- WEEK
- MONTH
- QTR
- YEAR

HEDUNIT and SALUNIT must be given the same value. In addition, the time period length that you specify with HEDUNIT and SALUNIT must be the same as the time period length that you specify with PERIOD_TYPE_CD, although the character strings that you use are different. See “PERIOD_TYPE_CD” on page 322.

### Example

Summarize the HEADSUM table by month:

```
%LET hedunit=month;
```

## V Family

### Description

Use the <TABLEACR>V macro variables to exclude specified columns from specified master and summary tables.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- ABSHMSTV: refers to the ABSHMAST table
- ACTHMSTV: refers to the ACTHMAST table
- APPHMSTV: refers to the APPHMAST table
- EMPMSTV: refers to the EMPMAST table
- HEADSUMV: refers to the HEADSUM table
- OPOSMSTV: refers to the OPOSMAST table
- OPSSUMV: refers to the OPOSSUM table
- PDSUMV: used for the Retention Analysis data extraction
- SALHISTV: refers to the SALHIST table
- SALHSUMV: refers to the SALHSUM table
- TERMNATV: refers to the TERMINAT table

**`column-1 <... column-n>`** are one or more columns to exclude from a master or summary table.

### Example

Exclude the CITY and STATE columns from the ACTHMAST table:

```
%LET acthmstv=city state;
```

## VARS Family

### Description

Use the <TABLEACR>VARS macro variables to specify columns that you want to add to a specified summary table.

### Syntax

```
%LET macro-variable=column-1 < ... column-n>;
```

**`macro-variable`** is one of the following:

- HEDVARS: refers to the HEADSUM table
- OPSVARS: refers to the OPOSSUM table
- SALVARS: refers to the SALHIST table
- TERMVARS: refers to the TERMINAT table

- TIPVARS: refers to the TIP table

**`column-1 <... column-n>`** are one or more columns to add to the summary table.

### *Example*

Add the AGE and GENDER columns to the SALHSUM summary table:

```
%LET salvars=age gender;
```

# Individual Macro Variables

## *AGEFRACT*

### *Description*

Use the AGEFRACT macro variable to specify whether to include decimal places in the AGE column.

### *Syntax*

```
%LET AGEFRACT = option;
```

**`option`** is either Y (to include decimal places in the AGE column) or blank (for no decimal places).

### *Example*

Include decimal places in the AGE column:

```
%LET AGEFRACT = Y;
```

## *CHRNDROP*

### *Description*

Use the CHRNDROP macro variable to specify columns to be dropped when the CHURN table is built from the base table that is designated by the CHRNINDS macro variable.

### *Syntax*

```
%LET CHRNDROP = column-1<...column-n>;
```

### *Example*

```
%LET CHRNDROP = JOB_CD;
```

## *CHRNINDS*

### *Description*

Use the CHRNINDS macro variable to designate the base table for the CHURN table.

### *Syntax*

```
%LET CHRNINDS = table_name;
```

**table_name** is an existing SAS Human Capital Management table.

### Example

```
%LET CHRNINDS = ACTHMAST;
```

## CHRNVARS

### Description

Use the CHRNVARS macro variable to specify the columns that are used to indicate a churn.

### Syntax

```
%LET CHRNVARS = column-1 <...column-n>;
```

### Example

```
%LET CHRNVARS = INTORG_HR;
```

## CUBE_DELETE_TYPE

### Description

Use the CUBE_DELETE_TYPE macro variable to specify the delete type for the cubes.

### Syntax

```
%LET CUBE_DELETE_TYPE = DELETE_PHYSICAL|DELETE;
```

DELETE_PHYSICAL prevents the metadata and permissions from being erased when the cube job is run.

DELETE deletes the cube and its metadata; it should be used when the cube structure changes.

### Example

```
%LET CUBE_DELETE_TYPE = DELETE_PHYSICAL;
```

## CUBE_DRILL

### Description

Use the CUBE_DRILL macro variable to indicate whether cubes should be built with the drill-to-detail feature. To give the drill-to-detail feature to some cubes but not others, build the cubes individually, with CUBE_DRILL set appropriately for each.

*Note:* SAS Human Capital Management row-level security is not applied for the drill to detail feature, meaning users can see all source detail rows for a cube.

### Syntax

```
%LET CUBE_DRILL = N|Y;
```

The default value is N.

### Example

Enable drill-to-detail:

```
%LET CUBE_DRILL = Y;
```

## CUBE_META_PATH

### Description

Use the CUBE_META_PATH macro variable to specify the metadata path for the OLAP cubes.

### Example

```
%LET CUBE_META_PATH=
   C:\SAS\Config\Lev1\AppData\SASHumanCapitalManagement5.2\Cubes;
```

## CUBEPATH

### Description

Use the CUBEPATH macro variable to specify the physical path to the OLAP cubes on the file system. The default physical path (referenced by **`&hcmdefault_cubepath`**) is set when SAS Human Capital Management is installed. The default path is ***`SAS-config-dir`*****`\Lev1\AppData\SASHumanCapitalManagement5.2\Cubes`**.

### Example

```
%LET CUBEPATH=C:\SAS\Config\Lev1\AppData\SASHumanCapitalManagement5.2\Cubes;
```

## DEBUG

### Description

Use the DEBUG macro variable to specify whether to turn on the DEBUG option. When this macro variable is set to Y, additional output is written to the SAS Human Capital Management build log for debugging purposes.

### Syntax

```
%LET DEBUG = option;
```

**`option`** is either Y (for additional output) or blank (to exclude additional output).

### Example

Write additional output to the SAS Human Capital Management build log for debugging purposes:

```
%LET DEBUG = Y;
```

## DIRECT

### Description

DIRECT is an optional macro variable that specifies the text to display whenever an employee record in the HCM Data Mart has no value in a hierarchy column. This can happen if an employee is located at the lowest level of a certain branch of the organization hierarchy, but not at the lowest level of the entire hierarchy.

### Syntax

```
%LET DIRECT = text;
```

**text** is the text that will be displayed.

### Example

Specify that the phrase “Direct Report” should be displayed whenever an employee record in the HCM Data Mart has no value in a hierarchy column:

```
%LET DIRECT = Direct Report;
```

## EDU

### Description

Use the EDU macro variable to specify the site-specific path to the National Center for Education Statistics (NCES) data files that hold the input for the Education Enrollment cube. To create the Education Enrollment cube in the HCM Data Mart, first set the value of EDU appropriately, and then run the following jobs in SAS Data Integration Studio:

1. Load HCM Education Enrollment Table
2. Create HCM Education Enrollment Cube

### Syntax

```
%LET EDU = valid_path;
```

### Example

For a Windows server:

```
%LET EDU=C:\sas\Config\Lev1\Data\HCMData\NCES;
```

For a UNIX server:

```
%LET EDU=/usr/local/SAS/Config/Lev1/Data/HCMData/NCES;
```

## EMPPOP

### Description

Use the EMPPOP macro variable to specify whether the EMPMAST table and the EMPCUBE cube should include all employees or only active employees.

### Syntax

```
%LET EMPPOP = <option>;
```

**option** is either ACTIVE or blank. Specify ACTIVE if only active employees are to be included in the tables. If the tables are to include all employees, then leave this option blank. In both cases, these tables contain the most recent record for each included employee.

### Example

Populate the EMPMAST and EMPCUBE tables with active employees only:

```
%LET EMPPOP = ACTIVE;
```

Populate the tables with all employees:

```
%LET EMPPOP =;
```

## FCTRVARS

### Description

Use the FCTRVARS macro variable to specify any additional columns to include as metric factors that are not found in the headcount summary table. These columns are brought in from the ACTHMAST table. If ONPAYRL and STECLASS are not in the HCM Data Mart, then remove them from the value of this macro variable.

### Syntax

```
%LET FCTRVARS = column-1 < ... column-n>;
```

### Example

```
%LET FCTRVARS = onpayrl steclass service_start_dt;
```

## FMT_UPDATE

### Description

Use the FMT_UPDATE macro variable to indicate whether the Load HCM Formats Table job should replace the HCMFORMATS table or update it.

### Syntax

```
%LET FMT_UPDATE = N|Y;
```

N causes the table to be replaced. Y causes the table to be updated.

### Example

```
%LET FMT_UPDATE = N;
```

## HCMGROUPNAME

### Description

Use the HCMGROUPNAME macro variable to specify the name of the Metadata Users group to which SAS Human Capital Management users are assigned. The default group name is HCM Solution Users.

### Example

```
%LET HCMGROUPNAME=HCM Solution Users;
```

## HCMLIB

### Description

Use the HCMLIB macro variable to specify the name of the SAS libref that is allocated for the HCM library. The default value is HCMData.

### Example

```
%LET HCMLIB=HCMData;
```

## HIERORDS

### Description

Use the HIERORDS macro variable to specify the respective sort orders for the hierarchies in the cubes.

### Syntax

```
%LET HIERORDS = ASCENDING|DESCENDING|ASCFORMATTED|DESFORMATTED|DSORDER;
```

### Example

Suppose that the HIERS macro variable is set as follows:

```
%LET HIERS = INTORG_HR INTORG_MGR;
```

Then the following line would sort the INTORG_HR hierarchy in ASCENDING order and the INTORG_MGR hierarchy in ASCFORMATTED order:

```
%LETHIERORDS = ASCENDING ASCFORMATTED;
```

## HIERS

### Description

Use the HIERS macro variable to specify the names of the organization hierarchies that will be used to build the HCM tables. The first hierarchy that is listed is the default hierarchy. The default hierarchy is used to build tables that contain only one hierarchy. The number of hierarchies in the list must be specified by the NUMBER_OF_HIERS macro variable.

These hierarchies must be defined in the INTERNAL_ORG_ASSOC_TYPE Detail Data Store table. For a detailed discussion of defining and loading organization hierarchies, see “Loading Members and Hierarchies into a Dimension” on page 193.

### Syntax

```
%LET HIERS = variable-1 < ... variable-n>;
```

### Example

```
%LET HIERS = INTORG_HR INTORG_MGR;
```

## INCOPEN

### Description

Use the INCOPEN macro variable to specify whether to include open positions in the EMPMAST table.

### Syntax

```
%LET INCOPEN = option;
```

**`option`** is either Y (to include open positions) or blank (to exclude open positions).

### Example

Include open positions in the EMPMAST table:

```
%LET INCOPEN = Y;
```

## INTORG_DIMENSION_CD

### Description

Use the INTORG_DIMENSION_CD macro variable to specify the code of the organization dimension that contains the organization hierarchies that are specified by the HIERS macro variable.

### Syntax

```
%LET INTORG_DIMENSION_CD = code;
```

**`code`** is the code of the appropriate organization dimension.

### Example

```
%LET INTORG_DIMENSION_CD = ORG;
```

## MAXLEVL

### Description

Use the MAXLEVL macro variable to specify the maximum number of hierarchy levels that are allowed. This number should be slightly greater than the known number of hierarchy levels in the organization. If you set this number too low, the build process will not create the WRKGRP table correctly. If you set this number too high, the build process will use more resources than necessary when it creates the WRKGRP table.

### Syntax

```
%LET MAXLEVL = number;
```

**`number`** is the number of hierarchy levels allowed.

### Example

Ensure that no more than 15 hierarchical levels are created:

```
%LET MAXLEVL = 15;
```

## MEAS_LEV

### Description

If you use the Create HCM Metric Table with Org and Load to the SDM job to compute HCM metrics, then you must use the MEAS_LEV macro variable to name the additional columns from ACTHMAST that are used to control summarization for the computation of the metrics.

Do not assign a value to MEAS_LEV if you compute HCM metrics with the Create HCM Metric Table and Load to the SDM job.

### Syntax

```
%LET MEAS_LEV = column-1 < ... column-n>;
```

### Example

If you are running the Create HCM Metric Table with Org and Load to the SDM job, which summarizes by time and organization, then specify the organization column:

```
%LET MEAS_LEV = INTORG_HR;
```

If you are running the Create HCM Metric Table and Load to the SDM job, which summarizes only by time, then do not assign a value to MEAS_LEV.

## MPRINT

### Description

Use the MPRINT macro variable to determine whether the SAS statements that are generated by the execution of the macros in the HCM build code are displayed in the HCM build log.

### Syntax

```
%LET MPRINT = MPRINT | NOMPRINT;
```

**MPRINT** displays the generated statements in the log.

**NOMPRINT** does not display the generated statements in the log.

### Example

Display the statements generated by the execution of macros in the log:

```
%LET MPRINT = MPRINT;
```

## NOTES

### Description

Use the NOTES macro variable to determine whether notes and warnings are displayed in the HCM build log.

### Syntax

```
%LET NOTES = NOTES | NONOTES;
```

**NOTES** writes notes and warnings to the log.

**NONOTES** does not write notes and warnings to the log.

### Example

Write notes and warnings to the log:

```
%LET NOTES = NOTES;
```

## NUMBER_OF_HIERS

### Description

Use the NUMBER_OF_HIERS macro variable to specify the number of organization hierarchies that will be used in building the HCM tables. This number must equal the number of hierarchies that are listed in the value of the HIERS macro variable. It must be equal to or less than the number of organization hierarchies that you load into the SDM.

### Syntax

```
%LET NUMBER_OF_HIERS = number;
```

### Example

```
%LET NUMBER_OF_HIERS = 2;
```

## ONPAYRL

### Description

Use the ONPAYRL macro variable to specify whether a given employee is on or off the payroll for the purpose of computing certain metrics for which Saratoga benchmarks exist.

### Syntax

```
%LET ONPAYRL = Y|N;
```

### Example

```
if EMPLOYEE_STATUS_CD='A' then ONPAYRL='Y';
   else ONPAYRL='N';
```

Here the value of ONPAYRL is set based on the value of EMPLOYEE_STATUS_CD.

## ORGMEMBER_EXCLUDES

### Description

Use the ORGMEMBER_EXCLUDES macro variable to specify organization members in the Detail Data Store INTERNAL_ORG table that are to be excluded from the WRKGRP table.

### Syntax

```
%LET ORGMEMBER_EXCLUDES = member-code-1<...member-code-n>;
```

### Example

```
%LET ORGMEMBER_EXCLUDES = ALL EXT;
```

ALL and EXT are required members of every organization hierarchy. They play an essential role in SAS Financial Management, but SAS Human Capital Management does not use them.

## PERIOD_TYPE_CD

### Description

Use the PERIOD_TYPE_CD macro variable to specify the length of the time periods for which HCM metrics are computed.

### Syntax

```
%LET PERIOD_TYPE_CD = code;
```

**code** is one of the following codes in the predefined SAS_PERIOD_TYPE table:

- DAY
- WK
- MO or MTH
- QTR
- YR

The time period length that you specify with PERIOD_TYPE_CD must be the same as the time period length that you specify with HEDUNIT and SALUNIT, although the character strings that you use are different.

### Example

```
 %LET PERIOD_TYPE_CD = YR;
```

## POSDIR

### Description

Use the POSDIR macro variable to specify whether the organizational hierarchy levels are built in ascending or descending order.

### Syntax

```
%LET POSDIR = A | D;
```

**A** represents ascending order.

**D** represents descending order.

### Example

Specify that the hierarchy levels are to be built in descending order:

```
%LET POSDIR = D;
```

## POSVAR

### Description

Use the POSVAR macro variable to specify the name of the column in the WRKGRP table, if any, which contains the hierarchy level of the reporting group in the record that is currently being read. If this value is blank, then SAS Human Capital Management determines the hierarchy levels dynamically while it creates the WRKGRP table.

### Syntax

```
%LET POSVAR = column
```

**`column`** is the name of a column that contains a hierarchy level.

### Example

Specify the column HIERLEV as the indicator of the hierarchy level:

```
%LET POSVAR = HIERLEV;
```

## POSVDIR

### Description

Use the POSVDIR macro variable to specify whether the values in the POSVAR indicator represent ascending or descending values. This direction is based on the transactional data structure.

### Syntax

```
%LET POSVDIR = A | D;
```

**`A`** represents ascending order.

**`D`** represents descending order.

### Example

The transactional data (HIERLEV) that is identified as POSVAR specifies the order of the transactional hierarchy. For example, if the organization has the four levels identified in the following list, the 4, 3, 2, 1 order has a POSVDIR value of A, although ascending and descending appear to be reversed. Furthermore, a department might report directly to a company. In this case, by using the POSVAR and POSVDIR variables, the hierarchy represents this skip in levels.

- Corp (top most level) transactional hierlev 4
- Company (second level) transactional hierlev 3
- Group (third level) transactional hierlev 2
- Department (fourth level) transactional hierlev 1

```
%LET POSVDIR = A;
```

## SALCVAR

### Description

Use the SALCVAR macro variable to specify which columns in the ACTHMAST table indicate whether a given record signals a change in the employee's pay. The HCM build code looks for changes in the value of the specified column. When a change is found, the record is written to the salary history table.

### Syntax

```
%LET SALCVAR = column-1 < ... column-n>;
```

**`column-1 <... column-n>`** are one or more columns that indicate a salary change.

### Example

When the value of SALARY changes for an employee, write the record to the salary history table:

```
%LET SALCVAR = salary;
```

## SOURCE

### Description

Use the SOURCE macro variable to determine whether the HCM build code is displayed in the HCM build log.

### Syntax

```
%LET SOURCE = SOURCE | NOSOURCE;
```

**`SOURCE`** writes the source code to the log.

**`NOSOURCE`** does not write the source code to the log.

### Example

Write the source code to the log:

```
%LET SOURCE = SOURCE;
```

## SRVFRACT

### Description

Use the SRVFRACT macro variable to specify whether to include decimal places in the SRVYRS column.

### Syntax

```
%LET SRVFRACT = option;
```

**`option`** is either Y (to include decimal places in the SRVYRS column) or blank (for no decimal places).

### Example

Include decimal places in the SRVYRS column:

```
%LET SRVFRACT = Y;
```

## STGACUT and STGACUTA

### Description

Use STGACUT and STGACUTA to specify the demographic segments for the Saratoga benchmark data to include in metric tables that are loaded with HCM metrics.

### Syntax

```
%LET STGACUT = demographic segment;
%LET STGACUTA = demographic segment;
```

### Example

```
%LET STGACUT = ALL;
%LET STGACUTA = ALL;
```

The following table shows the valid values for STGACUT and STGACUTA.

*Table 25.1 Valid Values for STGACUT and STGACUTA*

| Demographic Cut STGACUTA (variable STAT_TYPE_DS) | Demographic Cut STGACUT (variable STAT_DS) |
|---|---|
| All | All |
| Long Reporting Industry | Banking |
| | Chemicals & Petroleum Products |
| | Computer Products |
| | Computer Software & Services |
| | Consumer Products / Food & Beverage / Retail |
| | Government Agency |
| | Hospitals / Healthcare |
| | Insurance - All Lines & P, C, P |
| | Insurance - Healthcare Only |
| | Manufacturing |
| | Non-Bank Financials |
| | Pharmaceuticals / R and D |
| | Semiconductors |
| | Services |
| | Telecommunications |
| | Utilities |
| Region | Central |
| | East |
| | West |
| Revenue Growth | High (20%>) |
| | Low (<5%) |

| | |
|---|---|
| | Medium (5%–20%) |
| | Negative Loss |
| Short Reporting Industry | Banking, Non Bank Financial |
| | Chemicals / Petroleum Products |
| | Consumer Prod. / Food & Bev. / Retail |
| | Government Agency |
| | Hospitals / Healthcare |
| | Insurance |
| | Manufacturing |
| | Pharmaceuticals / R and D |
| | Services |
| | Technology |
| | Telecommunications |
| | Utilities |
| Size | 1–500 |
| | 501–1,000 |
| | 1,001–2,000 |
| | 2,001– 5,000 |
| | 5,001–10,000 |
| | 10,001–25,000 |
| | 25,001– 50,000 |
| | 50,000+ |

## STGAMAX

### Description

Use the STGAMAX macro variable to specify the most recent year of data to include in computations of metrics. You might want to exclude a recent year from the computation of metrics because you are comparing the computed metrics against Saratoga benchmarks that do not reflect data for that year.

### Syntax

```
%LET STGAMAX = number|ALL;
```

### Example

To exclude data for 2006 and subsequent years from the computation of metrics:

```
%LET SGTAMAX = 2005;
```

To base metric computations on all the data in the HCM Data Mart:

```
%LET SGTAMAX = ALL;
```

*Chapter 26*

# Internal Formats

**Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 329

**IACTION** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 329

**ICHURN** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 330

**IEEOCL** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 330

**IEMPSTA** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 331

**IEXEMPT** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 331

**IONPAYRL** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 331

**IPAYPER** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 332

**IREGTMP** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 332

**ISTECLS** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 332

**ITERM** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 333

## Introduction

This chapter describes the internal formats that are used by SAS Human Capital Management. These internal formats map values in your site's data to keywords in the SAS Human Capital Management software. The following tables describe each format and its mappings.

Modify the Start and End values of the internal formats so that they correspond to the data at your site. (Do not modify the format labels.) For information about editing internal formats, see “Auxiliary Files and Information Sources” on page 274.

## IACTION

IACTION maps certain personnel actions. If necessary, there can be more than one entry mapping to the same format label. This format is applied to the ACTION_TYPE_CD column.

Here are the format mappings from loaded data values to internal format labels. Additional format labels can be added, with their own mappings, for other site-defined personnel actions.

| Data Values for the Following Actions | Format Label Mapping |
|---|---|
| Involuntary terminations | ITERM |
| New hires | NHIRES |
| Pay changes | PAY |
| Voluntary terminations | VTERM |

## ICHURN

ICHURN maps Personnel Action codes to the CHURN format label. The input data values covered by this format are the job action codes that represent an employee voluntarily leaving one position to take another position that is within the organization in a different reporting group. As many action codes as necessary can map to the CHURN format label. This format is applied to the ACTION_TYPE_CD column.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following Job Actions | Format Label Mapping |
|---|---|
| Internal job changes or churning | CHURN |

## IEEOCL

IEEOCL maps EEO class codes. This format is applied to the EEO_CLASS_CD column and is used in the calculation of certain HCM metrics. The internal format labels associated with this format are the EEO classifications that are used in Saratoga Institute data.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Management | Mgmt |
| Professional Staff | Prof |
| Operations Staff | Op |
| Sales Staff | Sales |
| Office and Clerical Staff | OandC |

## IEMPSTA

IEMPSTA is used to determine whether an employee is active or inactive. This format is applied to the EMPLOYEE_STATUS_CD column. It is required; at least one employee status code must be mapped for each format label.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Active Employees | A |
| Inactive Employees | IN |

## IEXEMPT

IEXEMPT is used to classify employees as exempt or nonexempt according to the United States Fair Labor Standards Act. This format is applied to the EXEMPT_STATUS_CD column and is used in the calculation of certain HCM metrics.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Exempt | Ex |
| Non-Exempt | Nonex |

## IONPAYRL

IONPAYRL determines whether an employee is currently on the organization's payroll. This format is applied to the ONPAYRL column. It is used in the calculation of certain HCM metrics.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| On payroll | WFOnP |
| Off payroll | WFOffP |

## IPAYPER

IPAYPER is used to determine an employee's normal pay period, which is used to calculate an employee's total annual compensation. This internal format is applied to the PAY_GRADE_FREQUENCY_CD and PAY_FREQUENCY columns. It is required.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Hourly | HR |
| Annually | YR |
| Monthly | MNTH |
| Bi-Weekly | BW |
| Semi-Monthly | SM |
| Weekly | WEEK |
| Daily | DAY |

## IREGTMP

IREGTMP is used to determine whether an employee is a regular or temporary employee. As many entries as necessary can map to each format label. This format is applied to the PERMANENCE_CD column and is used in the calculation of certain HCM metrics.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Regular | R |
| Temporary, contract, or seasonal | T |

## ISTECLS

ISTECLS is used to determine the Saratoga employee class. This format is applied to the STECLASS column. It is required in order to calculate certain HCM metrics.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Executive Staff | Exec |
| Management, but not executive staff | Mgr |
| Non-management or non-executive staff | Staff |

# ITERM

ITERM determines the job action codes that indicate that an employee has left the organization, whether voluntarily or involuntarily. This format is applied to the ACTION_TYPE_CD column.

Here are the format mappings from loaded data values to internal format labels:

| Data Values for the Following | Format Label Mapping |
|---|---|
| Termination | TERMS |

# Part 3

# System Administration

*Chapter 27*
***Introduction to System Administration*** . . . . . . . . . . . . . . . . . . . . . . . . . . *337*

*Chapter 28*
***Post-Configuration Steps*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *341*

*Chapter 29*
***Assigning Groups and Roles*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *363*

*Chapter 30*
***Portal and Content Administration*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *373*

*Chapter 31*
***Administering the Middle Tier*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *377*

*Chapter 32*
***Viewing and Configuring the Log Files*** . . . . . . . . . . . . . . . . . . . . . . . . . . *381*

*Chapter 33*
***Administering MySQL Server*** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *383*

*Chapter 27*

# Introduction to System Administration

**Overview of SAS Solutions Services** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 337

**What's Changed in SAS Solutions Services** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 338

**Required Skills** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 338

**Documentation Conventions** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 339

**Additional Documentation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 339

## Overview of SAS Solutions Services

SAS Solutions Services is a set of services that provide common functionality and a framework for specific solutions. SAS Solutions Services builds on the SAS Intelligence Platform and includes the following areas of functionality:

- Document management allows users to create, organize, and secure documents of disparate types based on their own folder structures.

  A My Favorites portlet is available in the SAS Information Delivery Portal (referred to in this book as “the portal”). It provides shortcuts to the folders or the documents themselves, and some documents can also be viewed within a portlet.

  Document Manager, a Web application, supports management and viewing of the documents.
- Key performance indicator (KPI) management enables the user to create and manage KPIs for various levels within an organization. Based on security authorization, a user can create, manage, and modify KPI projects.

  A Web application, KPI Viewer, enables the user to open KPI projects. The Performance Dashboard portlet enables the user to put KPIs on a dashboard.
- Dimension Management provides the ability to create, manage, and add values to dimensions and hierarchies. A Java client application, SAS Solutions Dimension Editor, allows the user to interactively create and modify the dimensions.
- Microsoft Office integration provides the ability to integrate documents from SAS Solutions Services within the Microsoft Office suite of applications.
- A role-based user interface provides a means of associating user capabilities with the roles a user has (for example, administrator or analyst).

The Web applications of SAS Solutions Services are available as tasks in a My Favorites portlet.

## What's Changed in SAS Solutions Services

SAS Solutions Services 5.3 includes the following changes from SAS Solutions Services 1.4:

- Some features are now a part of the SAS Intelligence Platform or the Web Infrastructure Platform. As such, they are still available to users when they are using SAS Solutions Services and the solutions. They include:
  - the Solutions Web Administration application (with some changes in functionality)
  - the alerts service and the Alerts portlet
  - the directives service
  - the Configuration Manager plug-in of SAS Management Console
  - the Comment Manager application
- Users now log on via a common Logon Manager, which is part of the Web Infrastructure Platform.
- There are changes to the user identities, roles, and groups that are used in SAS Solutions Services as well as the solutions. For more information, see “Assigning Groups and Roles” on page 363.
- Document Manager has the following changes:
  - SAS Human Capital Management objects (such as a geographic analysis or an organization analysis) are not supported in Document Manager (or in a My Favorites portlet). For these objects, use the SAS Human Capital Management workspace.

    The My Favorites portlet does support a link to the Home page of SAS Human Capital Management.
  - The **Search** tab is removed. Use the portal **Search** button instead.
  - To run a stored process, users now click the stored process name. The **Refresh** menu option is not supported, and stored process reports (STO objects) are not supported.
  - Deleting a document no longer moves it to the **Trashcan**. Instead, it is completely deleted.
- The Measure Manager application is not included with SAS Solutions Services 5.3.
- Quick Help must be enabled or disabled in SAS Management Console for a site or for one or more software components. It cannot be configured at the end-user level.

For information about a particular solution, see the documentation for that solution.

## Required Skills

To administer the solutions software, you must be familiar with the operating system on which it is installed. For example, you must know how to create folders, run scripts (.bat files or .sh files), and update environment variables. On Microsoft Windows, you must be an administrator of the machine.

## Documentation Conventions

This book uses the following documentation conventions to identify paths in the solutions configuration:

| Path | Refers to | Example |
|---|---|---|
| `!SASROOT` | Path to the SAS root directory | Windows: `C:\Program Files\SAS\SASFoundation\9.2`<br>UNIX: `/usr/local/SAS/SAS_9.2` |
| ***`SAS-config-dir`*** | Path to the SAS configuration directory | Windows: `C:\SAS\Config`<br>UNIX: `/usr/local/SAS/Config` |
| ***`MySQL-install-dir`*** | Path to the MySQL installation directory | Windows: `C:\mysql`<br>UNIX: `/usr/local/mysql` |
| ***`WebSphere-install-dir`*** | Path to the installation directory for IBM WebSphere | Windows: `C:\Program Files\IBM\WebSphere\AppServer`<br>UNIX: `/usr/IBM/WebSphere/AppServer` |

*Note:*

- The name of the configuration directory and the level number might be different at your site.
- If your configuration is the result of a migration from the previous release of SAS Solutions Services, the **`SASApp`** directory might be called **`SASMain`** instead (for example, **`C:\SAS\Config\Lev1\SASMain`** rather than **`C:\SAS\Config\Lev1\SASApp`**). Please make the appropriate substitutions as you read this book.
- File system pathnames are typically shown with Windows separators ("\"); for UNIX, substitute a forward slash ("/").
- Some code examples contain line breaks (indicated by an underscore "_" at the end of the line) so that the code fits on the line. If you copy the code, remove the underscores and line breaks.

## Additional Documentation

For additional information about SAS Human Capital Management, see **`http://support.sas.com/documentation/onlinedoc/hcm`**.

*Note:* This site is password-restricted. You can find the user name and password in the preinstallation checklist or by contacting SAS Technical Support at `http://support.sas.com/techsup/contact`.

For information about administering the SAS Intelligence Platform, see the following documents (and others) at `http://support.sas.com/93administration`:

- *SAS Intelligence Platform: System Administration Guide*
- *SAS Intelligence Platform: Web Application Administration Guide*
- *SAS Intelligence Platform: Installation and Configuration Guide*
- *What's New in SAS 9.3 Intelligence Platform*

For information about administering third-party software, such as the Web application servers, see `http://support.sas.com/resources/thirdpartysupport/v93`.

# *Chapter 28*
# Post-Configuration Steps

**Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **342**
About the Post-Configuration Tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 342
Migration from SAS 9.1.3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 342
Upgrade in Place . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 342

**General Modifications** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **343**
Increase the Permanent Generation Size for SASServer1 . . . . . . . . . . . . . . . . . . . 343
Configure PC Files Server . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 343
(Optional) Configure Quick Help Display . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 345

**SAS Human Capital Management Modifications** . . . . . . . . . . . . . . . . . . . . . . . . . **346**
Modify Permissions for Remote Services . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 346
Add Permissions for the OLAP Schema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 346
Modify the Diagnostics Configuration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 347
(UNIX) Change Default Permissions for Cubes . . . . . . . . . . . . . . . . . . . . . . . . . . 348
(Optional) Modify SAS Web Report Studio Properties . . . . . . . . . . . . . . . . . . . . . 349

**KPI Viewer Modifications** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **350**
(KPI Viewer) Modify the login.config File to Support Export
to Information Map . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 350

**Secure Your Installation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **350**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 350
(Windows) Configure Security Settings for Folders and Files . . . . . . . . . . . . . . . . 351
(Windows, Optional) Secure Access to MySQL . . . . . . . . . . . . . . . . . . . . . . . . . . 352
(UNIX) Configure Security Settings for Folders and Files . . . . . . . . . . . . . . . . . . . 352

**(Optional) Verify Using Sample Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **353**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 353
Load the Sample Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 354
Verify the Installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 355
Reset the DDS, stageDDS, and HCM Databases . . . . . . . . . . . . . . . . . . . . . . . . . 356

**Create the Site's Users and Groups** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **358**

**Load Production Data** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **358**

**Localization** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **359**
SAS Human Capital Management Localization . . . . . . . . . . . . . . . . . . . . . . . . . . 359
KPI Viewer Localization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 361

**Add-On Configuration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **361**
Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 361
Import Data Source Definition Files for SAS BI Dashboard . . . . . . . . . . . . . . . . . 362

# Overview

## *About the Post-Configuration Tasks*

This chapter describes the post-configuration tasks that need to be performed for SAS Solutions Services and SAS Human Capital Management. It also describes how to load sample data that can be used to verify the installation and demonstrate the software.

It contains the following information:

- tasks that need to be performed
- optional tasks
- information about securing the installation
- information about verifying the installation
- tasks that apply to localization
- tasks that apply if you are installing one of the solutions as an add-on to an existing installation

For more information about the solutions, see the online Help and user's and administrator's guides, as well as the Data Administration part of this book. (See "Additional Documentation" on page 339.)

We recommend that you check the SAS Notes for additional information and support fixes. Go to **`support.sas.com/notes`**.

## *Migration from SAS 9.1.3*

The topics in this chapter apply to migrations from the *X*.4 version of the solutions, as well as new installations. However, do not install the sample data on a migrated system. Doing so would overwrite your existing data.

## *Upgrade in Place*

If you are upgrading from SAS Human Capital Management 5.1 (in other words, if you are performing an upgrade in place), please check the instructions in this chapter and perform any applicable steps that you have not already performed. Do not install the sample data; doing so would overwrite your existing data. (The 5.2 sample data folder is purposely not provided for an upgrade in place.)

In particular, please note the following topics:

- "Configure PC Files Server" on page 343
- "Add-On Configuration" on page 361

# General Modifications

## *Increase the Permanent Generation Size for SASServer1*

In the JVM options for the SASServer1 managed server, modify the PermSize and MaxPermSize options as follows:

```
-XX:PermSize=768m -XX:MaxPermSize=768m
```

Then restart SASServer1 and the other managed servers.

These options apply to the IBM WebSphere Server (Solaris only) and to the Oracle WebLogic Server.

For information about setting JVM options, see *SAS 9.2 Web Applications: Tuning for Performance and Scalability*, available at `http://support.sas.com/resources/thirdpartysupport/index.html`. Select **`Web Application Servers and HTTP Servers`** and then select the application server that applies to your site.

## *Configure PC Files Server*

### *Overview*

Installing SAS PC Files Server is mandatory for 64-bit Windows installations and optional for UNIX installations. It enables users to load data from 32-bit PC files into 64-bit SAS. With this configuration, data administrators can use Microsoft Excel or Microsoft Access files as input to jobs in SAS Data Integration Studio.

The following sections explain how and why to change the port number for SAS PC Files Server and how to use a Microsoft Office file as a data source in SAS Data Integration Studio. It assumes that you have already installed SAS PC Files Server as a service on a Windows machine.

### *Change the Port Number for SAS PC Files Server*

SAS PC Files Server uses port 8621 by default. This is also the default port for the SAS Stored Process Server. If you kept the default for SAS Stored Process Server, modify the port number for SAS PC Files Server, as follows:

1. Stop the Windows service that runs the SAS PC Files Server by typing the following command at a command prompt:

   ```
   net stop service-name
   ```

2. From the Windows Start menu, select **SAS** ⇨ **PC Files Server**.
3. In the application window, click **Change Options**.
4. Change the port number from **8621** to **9621** or another unused port number. Save your changes.
5. Click **Shutdown Server** to stop the desktop application.
6. Restart the service:

   ```
   net start service-name
   ```

### Install the Office 2007 ODBC Driver

On the machine where you installed SAS PC Files Server, install the Microsoft Office 2007 ODBC driver, which works with both Office 2007 and Office 2003 files. To download the driver, see the following SAS Note: `http://support.sas.com/kb/37/521.html`.

### Set HCMConfig.xml Property

If your installation is on a 64-bit Windows operating system, you must also set the following property to **`true`** in the HCMConfig.xml file:

```
<Property Id="OSVersion" Name="Is 64 bit OS" Value="true" ReadOnly="false"/>
```

The HCMConfig.xml file is located on the middle tier, in the ***`SAS-config-dir`*** **`\Lev1\AppData\SASHumanCapitalManagement5.2`** directory.

### Validate the Configuration Changes

To validate the changes that you made, follow these instructions for importing a Microsoft Excel file:

*Note:* The file to be imported must reside on the machine where SAS PC Files Server is installed, and it must be accessible from the data tier.

Add a library for the imported files, as follows:

1. Log on to SAS Data Integration Studio as a data administrator.
2. On the **Inventory** tab, right-click the **Library** folder and select **New Library**.
3. Select **Resource Templates** ⇨ **Libraries** ⇨ **Generic Library**. Click **Next**.
4. Enter a name for the library.
5. Click the **Browse** button and select a location for the library. Click **Next**.
6. From the **Available Servers** select **SASApp**. Click **Next**.
7. On the Library properties page, enter the following values:
   - **Libref**: Enter a libref with a maximum of 8 characters.
   - **Engine Type**: Enter **`pcfiles`**.
   - **Other Options**: Enter the path to the file that you want to import and the port number for SAS PC Files Server. If the file is on a machine other than the data tier, enter the server name. For example:

     ```
     path="C:\MyFiles\myfile.xlsx" port=9621 server=servername
     ```

   For more information, see *SAS/ACCESS(R) 9.2 Interface to PC Files: Reference* at `http://support.sas.com/documentation`
8. Click **Next**.
9. Review your selections and click **Finish**.

Register the Excel file that you want to import:

*Note:* You cannot register tables as the unrestricted user.

1. Right-click the new library and select **Register tables**.
2. Select the following options:
   - **Enable case-sensitive**
   - **Enable special characters**

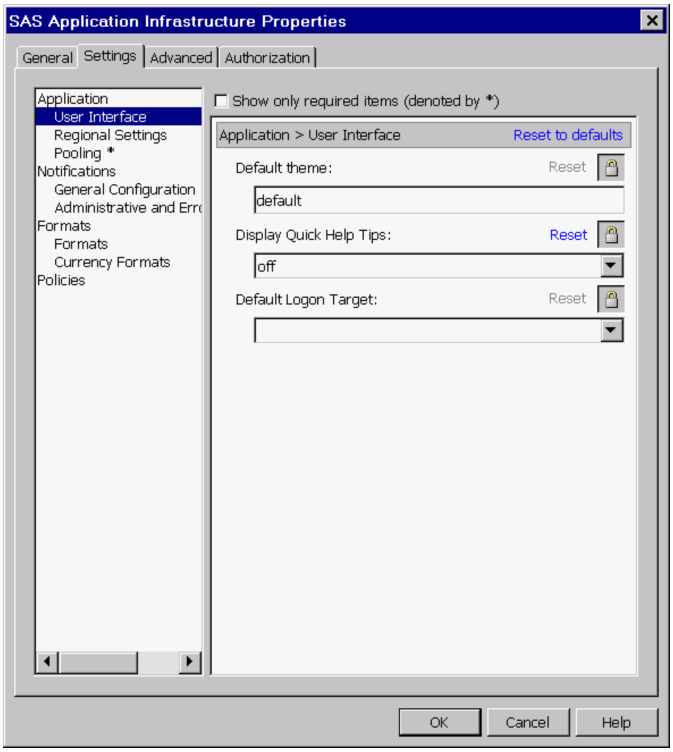
3. Select the tables to register.
4. Click **Next**.
5. Review your selections and click **Finish**.

## *(Optional) Configure Quick Help Display*

In the solutions, Quick Help is a short Help topic that can be automatically displayed on a Web application page. By default, Quick Help display is disabled. As an administrator, you can enable or disable Quick Help display for all the solutions applications, or you can enable or disable it for individual applications.

To enable Quick Help display:

1. Log on to SAS Management Console as the SAS Administrator.
2. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager**.
3. Right-click **SAS Application Infrastructure** and open its properties.
4. Click the **Settings** tab:



SAS Application Infrastructure Properties

General | Settings | Advanced | Authorization

Application
- User Interface
- Regional Settings
- Pooling *

Notifications
- General Configuration
- Administrative and Erro

Formats
- Formats
- Currency Formats

Policies

□ Show only required items (denoted by *)

Application > User Interface — Reset to defaults

Default theme: default

Display Quick Help Tips: off

Default Logon Target:

5. Click the Lock button 🔒 for **Display Quick Help Tips**.

   By default, this property is locked. Unlocking the property makes it possible to change its value in components that inherit it.
6. Click **OK**.
7. Right-click **Solutions Svc 5.3** and select **Properties**.
8. Click the **Settings** tab.

9. From the **Display Quick Help Tips** drop-down box, select **Yes** to enable Quick Help display.
10. Click **OK**.
11. Restart the managed servers.

*Note:* Do not re-lock the quick help configuration property for SAS Application Infrastructure. Doing so erases the changes that you made to the solutions applications, and they would once again inherit their setting from SAS Application Infrastructure.

For details about the Configuration Manager, see “Administering the SAS Web Infrastructure Platform” in the *SAS Intelligence Platform: Web Application Administration Guide*.

# SAS Human Capital Management Modifications

## Modify Permissions for Remote Services

You must modify permissions for the remote services so that the event broker service works correctly. Follow these steps:

1. Log on to SAS Management Console as the SAS Administrator.
2. On the **Plug-ins** tab, navigate to **Environment Management** ⇨ **Foundation Services Manager.**
3. Right-click **Remote Services** and select **Properties**.
4. On the **Authorization** tab of the properties window, click **Add**.
5. Add HCM Solution Users to the **Selected Identities** list, and click **OK**.
6. Grant Read Metadata permission to the HCM Solution Users group. Deny Write Metadata and CheckInMetadata permissions to this group.
7. Click **OK**.

## Add Permissions for the OLAP Schema

If members of the HCM Solution Users group do not have permissions for the OLAP schema, add those permissions as follows:

1. Log on to SAS Management Console as the SAS Administrator.
2. On the **Folders** tab, navigate to **Shared Data** ⇨ **SASApp - OLAP Schema**.
3. In the right pane, right-click the OLAP schema name and select **Properties**.
4. On the **Authorization** tab, grant ReadMetadata permission to the HCM Solution Users group. (If necessary, add this group to the list of identities.)

   Grant ReadMetadata and WriteMetadata permission to users who will administer data for SAS Human Capital Management.
5. Click **OK**.

For more information about the metadata permissions that are required for using cubes, see the “Authorization Model” chapter of the *SAS Intelligence Platform: Security Administration Guide*. For information about operating system permissions for cubes, see “(Windows) Configure Security Settings for Folders and Files” on page 351 and “(UNIX) Configure Security Settings for Folders and Files” on page 352.

## *Modify the Diagnostics Configuration*

### *Add the JUnit JAR File*

Before running the diagnostics, you must copy the junit-4.5.jar file to the ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\Diagnostics\lib`** directory on the middle tier.

You should have downloaded this JAR file during the installation process. For details, see the *Installation Instructions for Release 5.2 of the SAS Performance Management Solutions*.

*Note:* If your installation has the ***`SAS-config-dir`*** **`\Lev1\Web\SASDomain\lib`** directory, and the junit-4.5.jar file is stored there, remove the junit-4.5.jar file from that directory. Not all installations will have this directory.

### *(If Necessary) Modify Diagnostics Configuration Files*

Before using the HCM Diagnostics utility, you might need to modify the configuration files used by the utility. If your installation is a migration from a previous version or an upgrade in place, you must modify the configuration files.

The diagnostics configuration files are located in the ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\Diagnostics`** directory on the middle tier.

Follow these steps:

*Note:* Line breaks (“_”) are added to some example code for readability.

1. Open the HCMDiagnosticsConfig.xml file for editing.
2. If the site changed the names of components such as logical servers, the EAR filename, WAR filename, or table names, modify those names in the configuration file.

   For example, a migrated system uses **`SASMain`** rather than **`SASApp`**.
3. Find the **<Configuration>** ⇨ **<Connection>** ⇨ **<Properties>** ⇨ **<FileSystem>** ⇨ **<DataTier>** section of the XML file.
4. Within that section, modify the paths to any folders or files on the data tier.

   *Note:* Use local paths rather than UNC paths.
5. By default, there are no files listed for the data tier. When you run the diagnostics, a warning message is displayed, noting the absence of files and advising you to check the configuration file.

   If the **`<Files>`** section is indeed empty, you can safely ignore this warning message. To prevent the warning message from being displayed, add a file to be checked, as follows:

   a. Find the **<Configuration>** ⇨ **<Connection>** ⇨ **<Properties>** ⇨ **<FileSystem>** ⇨ **<DataTier>** ⇨ **<Files>** section of the XML file.

b. Add a **<File>** node that specifies a file on the data tier. Here is an example:

```
<Files>
   <File absolutepath="C:\SAS\Config\Lev1\AppData\  _
   SASHumanCapitalManagement5.2\StoredProcesses\absmean.sas"  _
   name="HCM Stored Process"/>
</Files>
```

*Note:* Alternatively, on the initial screen for the diagnostics, you can disable the FileSystem diagnostics for the data tier.

6. Save the file.
7. Open the PlatformDiagnosticsConfig.xml file for editing.
8. If necessary, modify the component names in this file.
9. Find the following section:

```
<SoftwareComponent keyProperties="[bid.WrsSoftwareComponentName] _
[bid.LogFile]" name="BI Dashboard 4.3" propertiesToPrint="[Name] [bid.LogFile]" _
verifyKeyProperties="true"/>
```

10. Remove the two references to the **[bid.LogFile]** property. The result should look like this:

```
<SoftwareComponent keyProperties="[bid.WrsSoftwareComponentName]"  _
name="BI Dashboard 4.3" propertiesToPrint="[Name]"  _
verifyKeyProperties="true"/>
```

11. Save the file.

*Note:* The diagnostics utility currently supports single-tier and two-tier configurations. A two-tier configuration is assumed to be composed of a data tier and a middle tier, with the diagnostics being deployed on the middle tier.

When you run the diagnostics utility, you might see a message saying that the sastrust user has no SAS Human Capital Management role or employee mapping. You can ignore this message. This identity does not log on to SAS Human Capital Management and so does not need a role or employee mapping.

For more information about running the diagnostics utility and interpreting the results, see "The Diagnostic Utility" on page 87.

## (UNIX) Change Default Permissions for Cubes

By default, in a UNIX installation of the solutions, users do not have write permission for files that they did not create. However, frequently you want one administrator to be able to rebuild a cube that was originally created by another administrator. (If your site has a single administrator, you can skip these instructions.)

The following steps set the default permissions to be applied when a user creates a file via code that is running on the workspace server.

1. In the operating system, create a group that includes all SAS Human Capital Management administrators who will be building cubes. Make it the primary group for these users.
2. On the data tier, open the WorkspaceServer_usermods.sh file for editing. This file is located in the ***SAS-config-dir*/Lev1/SASApp/WorkspaceServer** directory.
3. Just before the **USERMODS_OPTIONS=** line, add code similar to the following:

```
CURR_GID=`/usr/bin/id -g`
GID=hcm-group-id
if [ $CURR_GID -eq $GID ]; then
   umask 002
fi
```

Replace *hcm-group-id* with the ID for the group that you created in Step 1. Use back ticks (not single quotes) around **/usr/bin/id -g**.

4. Save the file.

The umask settings apply only to this group and give the group read and write permission for new files.

*Note:* Use a version of the **id** command that supports the **-g** option. For example, under Solaris, you need to use **/usr/xpg4/bin/id** instead.

Make a similar change to the PooledWorkspaceServer_usermods.sh file. This file is located in the ***SAS-config-dir*/Lev1/SASApp/PooledWorkspaceServer** directory.

If the code shown above does not work in your installation, try replacing it with the following code, which grants write permission for new files to all users, rather than to a specified group:

```
umask 002
```

## (Optional) Modify SAS Web Report Studio Properties

SAS Human Capital Management includes a set of reports for viewing in SAS Web Report Studio. These reports are automatically loaded during the installation. If a report query fails with an error saying that the data set is too large, you might need to increase the MAX_TUPLE_COUNT value. This value limits the amount of data that can be retrieved from a query.

To set the MAX_TUPLE_COUNT property, follow these steps:

1. Open SAS Management Console as the SAS Administrator.
2. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager**.
3. Right-click **Web Report Studio 4.3** and select **Properties**.
4. Click the **Advanced** tab.
5. Click **Add** to add a new property.
6. In the **Property Name** field, enter this value:

   ```
   .vmwide.com.sas.iquery.dataservices.ProcSummaryROLAPBuilder.MAX_TUPLE_COUNT
   ```
7. In the **Property Value** field, enter a value. The default is 2500.
8. Click **OK** to save the new definition, and then click **OK** again to close the properties window.
9. Restart the managed servers.

For more information about modifying properties for SAS Web Report Studio, see the “Configuring SAS Web Report Studio” chapter of the *SAS Intelligence Platform: Web Application Administration Guide*.

### See Also

- For localization information, see "SAS Human Capital Management Localization" on page 359.
- If you installed SAS Human Capital Management as an add-on to an existing installation of the SAS Intelligence Platform, see "Add-On Configuration" on page 361.

## KPI Viewer Modifications

### (KPI Viewer) Modify the login.config File to Support Export to Information Map

In order to use the **Export to Information Map** feature of the KPI Viewer, you must modify the login.config file for the remote services, as follows:

1. Open the login.config file for editing.

   This file is located on the middle tier, in the ***SAS-config-dir*\Lev1\Web\Common** directory.

2. Modify the **PFS** block that contains **com.sas.services.security.login.OMILoginModule**.

   Change **"aliasdomain"="DefaultAuth"** to **"aliasdomain"="MidtierInternal"**.

   Do not modify the **SCS** block.

3. Save the file.

4. Restart the remote services and the managed servers.

*Note:* After a migration or an upgrade in place, re-export any existing, non-working information maps from scorecard or KPI projects. Follow the instructions in the online Help for saving a scorecard as an information map.

## Secure Your Installation

### Overview

This section contains information about setting operating system protection for files and folders, for both Windows and UNIX. It also contains instructions for securing access to MySQL Server.

Additional information:

- For an overview and detailed information about security in the SAS Intelligence Platform, see the *SAS Intelligence Platform: Security Administration Guide*.
- If you installed SAS Web Report Studio, see "Configuring SAS Web Report Studio" in the *SAS Intelligence Platform: Web Application Administration Guide*. This

chapter includes information about securing the folders that are used by SAS Web Report Studio, including folders that hold temporary files.

These books are available at **support.sas.com/92administration**.

## *(Windows) Configure Security Settings for Folders and Files*

To configure security for configuration directories, follow these steps:

1. Apply the operating system protections that are recommended for configuration directories on the SAS Intelligence Platform. For instructions, see “What to Do Next: Administration Tasks” in the *SAS Intelligence Platform: System Administration Guide*, which is available at **support.sas.com/92administration**.
2. Apply the additional protections that are recommended in Table 28.1 on page 351. All of these directories are located in ***SAS-configuration-directory*\Lev1** on the data tier.
3. Provide the operating system protections in Table 28.2 on page 351 to the MySQL directories.

***Table 28.1*** *Windows: Protections That Apply to Solutions Directories*

| **Directories** | **Permissions** |
|---|---|
| **Appdata \SASHumanCapitalManagement5.2\Cubes** | Grant Modify permission to the SAS Server Users group. |
| Under **SASApp\SASEnvironment\ [SolutionsServices, FinancialManagement, HumanCapitalManagement, StrategyManagement]:**:<br>**SASCode\Jobs**<br>**SASFormats**<br>**SASMacro** | Grant Modify permission to the SAS Server Users group. |
| **SASApp\Data** and its subdirectories | Grant Full Control to SAS General Server User (sassrv) .<br>Grant Read/Write/Modify permission to users who run SAS Data Integration Studio jobs to update data in the data warehouse. These users should include the Solutions Host User (sassln). |

***Table 28.2*** *Windows: Recommended Operating System Protections for the MySQL Directories*

| **Directories** | **Permissions** |
|---|---|
| ***MySQL-install-dir*** | Grant Full Control to SYSTEM and Administrators only. |

| Directories | Permissions |
|---|---|
| ***MySQL-install-dir*\bin** | Grant Read and Execute permissions to the Everyone group.<br>(During installation and configuration, you were asked to give this group Read, Execute, and Modify permissions. After installation and configuration, restrict these permissions.) |

## (Windows, Optional) Secure Access to MySQL

On Windows, MySQL is installed as a system service by default. Consequently, the service has access to all directories. MySQL can be used only with its own user IDs.

*Note:* During the configuration process, several MySQL users are created, and the root user for MySQL is deleted after it is no longer needed.

To restrict the IP address that MySQL uses, perform these steps after the configuration has been validated:

1. On the machine where MySQL resides, create a file (grant.sql) with these contents (line breaks are inserted for readability):

   ```
   revoke all privileges, grant option from 'sqladmin'@'%';
   GRANT ALL PRIVILEGES ON *.* TO sqladmin@'datatier'
       IDENTIFIED BY 'mysqlpassword'
       WITH GRANT OPTION;
   GRANT ALL PRIVILEGES ON *.* TO sqladmin@'midtier'
       IDENTIFIED BY 'mysqlpassword'
       WITH GRANT OPTION;
   GRANT ALL PRIVILEGES ON *.* TO sqladmin@'localhost'
       IDENTIFIED BY 'mysqlpassword'
       WITH GRANT OPTION;
   flush privileges;
   ```

   In this file, make the following changes:

   a. Replace *mysqlpassword* with the password for sqladmin.

   b. Replace *datatier* and *midtier* with the fully qualified host names of the data tier and middle tier, respectively.

   Save the file.

2. At a command prompt, execute this command (replacing *mysqlpassword* with the password for sqladmin):

   ```
   mysql -usqladmin -pmysqlpassword < grant.sql
   ```

Follow the same procedure for additional MySQL users (depending on the products that you installed): sassdmdbadm, hcmdbadm, and spmdbadm.

## (UNIX) Configure Security Settings for Folders and Files

To configure security for configuration directories on UNIX, follow these steps:

1. Apply the operating system protections that are recommended for configuration directories on the SAS Intelligence Platform. For instructions, see “What to Do Next:

Administration Tasks” in the *SAS Intelligence Platform: System Administration Guide*, which is available at support.sas.com/92administration.

2. Apply the additional protections that are recommended in the following table. All of these directories are located in ***SAS-configuration-directory*/Lev1** on the data tier.

***Table 28.3*** *UNIX: Protections That Apply to Solutions Directories*

| Directories | Permissions |
|---|---|
| **Appdata/SASHumanCapitalManagement5.2/ Cubes** | Permit full access for the sas user ID and the sas user group. |
| Under **SASApp/SASEnvironment/ [SolutionsServices, HumanCapitalManagement]:**:<br>**SASCode/Jobs**<br>**SASFormats**<br>**SASMacro** | Permit full access for the sas user ID and the sas user group |
| **SASApp/Data** and its subdirectories | Permit full access for the sas user ID and the sas user group. |

# (Optional) Verify Using Sample Data

## *Overview*

Sample data is provided to help you verify the correct operation of the software and to demonstrate functionality of the solutions. After verification, you can run batch programs to load the DDS, stageDDS, and HCM databases again. These batch programs reset the databases to their default state (the state that they were in right after installation and configuration).

***CAUTION:***

If this installation is the result of a migration or an upgrade in place, or if you have already stored data that you want to keep, do not load the sample data or run the batch programs.

The installer or data administrator should load the sample data. Here is an overview of the procedure.

1. On the data-tier server, load the sample data for the Cross Industry Detail Data Store (DDS) and the staging tables (stageDDS).
2. Load the sample data for the HCM Data Mart.
3. Restart the managed servers.
4. Rebuild cubes and information maps.
5. Verify the installation.
6. Reset the databases to their default state.

7. Restart the managed servers.

## Load the Sample Data

### Load Sample Data for the Cross Industry Detail Data Store

Running the LoadDDSSampleData script replaces all the tables in the Cross Industry Detail Data Store (DDS) as well as the staging tables (stageDDS).

1. At a command prompt, change directory to ***`SAS-config-dir`*** **`\Lev1\Applications\SASSolutionsServices5.3\SampleData`**.

   *Note:* Logs for these commands are written to a subdirectory of the current working directory.

2. Run one of the following scripts to load the DDS and stageDDS sample data:

   - Windows: LoadDDSSampleData.bat
   - UNIX: LoadDDSSampleData.sh

3. When prompted, enter the following connection information:

| Prompt | Description |
|---|---|
| **SAS Metadata User ID** | User ID of the unrestricted user (for example, sasadm@saspw) |
| **SAS Metadata User Password** | User password |

### Load the HCM Sample Data

If you have installed SAS Human Capital Management, you can also run the LoadHCMSampleData command, which populates tables in the HCM database with fictitious employee data.

1. At a command prompt, change directory to ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\SampleData`**.

2. Run one of the following scripts to load the sample data:

   - Windows: LoadHCMSampleData.bat
   - UNIX: LoadHCMSampleData.sh

3. When prompted, enter the following connection information:

| Prompt | Description |
|---|---|
| **MySQL Host Name** | Name of the host machine where MySQL is running |
| **MySQL DB Name** | **hcm** |
| **MySQL Port (default:3306)** | MySQL port number (default: 3306) |
| **MySQL DB User ID** | User ID for accessing the HCM database |

| Prompt | Description |
|---|---|
| **MySQL DB User Password** | Password for accessing the HCM database |
| **SAS Metadata User ID** | User ID of the unrestricted user (for example, sasadm@saspw) |
| **SAS Metadata User Password** | User password |

If you installed MySQL Server on a different machine, the batch file prompts you to run the scripts manually. Follow these steps:

1. Copy the hcmdata.sql file to the machine where MySQL is running.

   This script is located in the ***`SAS-install-dir`*** **`\SASHumanCapitalManagementDataTier\5.2\SampleData`** directory on the data tier.

2. On the MySQL machine, run the command as you were instructed in the batch file output.

   *Note:* Replace the source directory name in the command with the new source directory on the MySQL machine.

3. On the data-tier machine, run the register_hcm_sample_data.sas command as you were prompted by the batch file output.

### Copy the %CODENODE Macro

After you load the sample data for SAS Human Capital Management, copy the %CODENODE macro so that it can be used by the ETL job that loads the MODELSCORES table. Follow these steps:

1. On the data tier, find the codenode.sas file in the ***`SAS-install-dir`*** **`\SASHumanCapitalManagementDataTier\5.2\SampleData`** directory.

2. Copy the file to the ***`SAS-config-dir`*** **`\Lev1\SASApp\SASEnvironment\HumanCapitalManagement\SASMacro`** directory.

For details about the related ETL job, see the Data Administration part of this book.

## Verify the Installation

### Prepare for Verification

After you load the sample data:

1. Restart SASServer3, the ODCS servers, and the Web Data Entry (WDE) server (if there is one).

2. Rebuild the cubes and information maps. The Administration application of SAS Human Capital Management has utilities to rebuild these objects. There are also jobs for building cubes and information maps in SAS Data Integration Studio (see the Data Administration part of this book).

3. In SAS Management Console, make sure that the SAS Demo User belongs to the HCM Solution Users group and has the HCM Administrator role.

### Run the Diagnostic Utility

Before running any of the SAS Human Capital Management applications, we recommend running the Diagnostic utility:

1. Make sure that you have installed the JUnit.jar file. (See "Add the JUnit JAR File" on page 347.)

2. If necessary, modify the diagnostics configuration file. (See "Modify the Diagnostics Configuration" on page 347.)

3. On the middle tier, run the diagnostics.

   From the Windows Start menu, use the shortcut at **All Programs** ⇨ **SAS** ⇨ **SAS Configuration** ⇨ **Config - Lev1**.

   On UNIX, use this command: ***`SAS-config-dir`*****`/Lev1/Applications/SASHumanCapitalManagement5.2/Diagnostics/launchDiagnostics.sh`**.

For details about running the diagnostics, see "The Diagnostic Utility" on page 87.

### Log On to SAS Human Capital Management

To verify the Web applications that are part of SAS Human Capital Management, follow these steps:

1. Log on to SAS Human Capital Management using the URL in the Instructions.html file.

2. On the Home page, enter a string such as **`smith`** and click **Search**.

3. On the search results page, click the SAS logo to return to the Home page.

4. From the **Tasks** list, try one or more of the following links:

   - **My Employee Profile**
   - **New Organization Analysis**
   - **New Geographic Analysis**
   - **My Portal**

5. From the **Manage** list, try the following links:

   - **Workspace**
   - **Administration**

6. Log off.

## Reset the DDS, stageDDS, and HCM Databases

On the data tier, follow these steps to reset the DDS, stageDDS, and HCM databases to their default state:

1. At a command prompt, change directory to ***`SAS-config-dir`*****`\Lev1\Applications\SASHumanCapitalManagement5.2\SampleData`**.

2. You must update some variables the restore script before executing it. The scripts are as follows:

   - Windows: restore_original_install_data_fm.bat
   - UNIX: restore_original_install_data_fm.sh

Make a backup copy and then edit the appropriate script, modifying the following variables as necessary for your site configuration.

*Note:* When you modify the script, replace *SAS-config-dir* and **`!SASROOT`** with the appropriate file system path. Replace **`SASApp`** and **`Lev1`** as necessary for your site. In the SET statements, do not use spaces around the equal sign.

| **Variable** | **Description** |
|---|---|
| SAS_EXE | Full path (including filename) to the sas executable, located in the **`!SASROOT`** directory.<br>On Windows, a typical path is **`C:\Program Files\SAS\SASFoundation\9.2`**.<br>On a UNIX system, a typical value is **`/usr/local/SAS/SAS_9.2`**. |
| SAS_DIR (UNIX) | Path to the **`!SASROOT`** directory. In UNIX scripts, this variable is used in the definition of SAS_EXE. |
| SAS_CONFIG (Windows) or SOL_CFG (UNIX) | Full path (including filename) to the SASV9.cfg file that is located in the ***`SAS-config-dir`*** **`\Lev1\SASApp`** directory.<br>On Windows, a typical location is **`C:\SAS\Config\Lev1\SASApp\sasv9.cfg`**.<br>On UNIX, a typical location is **`/usr/local/SAS/Config/sasv9.cfg`** |
| REG_STAGEDDS_DDS | Full path (including filename) to the register_stagedds_and_dds.sas file, which is located in the ***`SAS-config-dir`*** **`\Lev1\Applications\SASSolutionsServices5.3\SASCode`** directory. |
| REG_HCM | Full path (including filename) to the register_hcm.sas file, which is located in the ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\SASCode`** directory. |

3. Run one of the following scripts:

- Windows: restore_original_install_data_hcm.bat
- UNIX: restore_original_install_data_hcm.sh

When prompted, enter the following connection information:

| **Prompt** | **Description** |
|---|---|
| **DB Host Name** | Name of the host machine where MySQL is running |
| **DB PORT** | Port number for MySQL (default: 3306) |
| **DB NAME** | Name of the MySQL database to be restored (HCM) |

| Prompt | Description |
|---|---|
| **DB User ID** | User ID for accessing the HCM database |
| **DB Password** | Password for accessing the HCM database |
| **META USER ID** | User ID of the unrestricted user (for example, sasadm@saspw) |
| **META USER PASSWORD** | Metadata user password |

The batch program runs several SAS registration programs. The logs from these programs are in the ***SAS-config-dir*\Lev1\Logs\Configure** directory.

If you installed MySQL Server on a different machine, the batch file prompts you to run the scripts manually. Follow these steps:

1. On the machine where MySQL Server is installed, run the command to drop and then create the HCM database, as you were prompted by the batch file output.
2. On the data-tier machine, run the following .sas commands as you were prompted by the batch file output.
   - register_stagedds_and_dds.sas
   - register_hcm.sas

After you run the scripts, restart SASServer3, the ODCS servers, and the Web Data Entry (WDE) server.

## Create the Site's Users and Groups

Register users at the site and assign them to groups and roles. For instructions, see "Assigning Groups and Roles" on page 363.

*Note:* On Windows, in order for users to access a standard workspace server using credential-based host authentication, they need the local **Log on as a batch job** right on that machine. For more information, see "Windows Privileges" in the *SAS Intelligence Platform: Security Administration Guide*.

## Load Production Data

For instructions about loading production data, see the Data Administration part of this book.

If you installed SAS Human Capital Management: After loading production data, rebuild the cubes and information maps, either in the Administration application of SAS Human Capital Management or via SAS Data Integration Studio jobs.

# Localization

## *SAS Human Capital Management Localization*

### *Set the Correct Date Format*

If you installed the HCM MySQL database using a stored date format other than the default (**yyyy-MM-dd**), you must modify the HCM configuration file, so that conversions from stored date formats to displayed date formats are correct. Follow these steps:

1. Open the HCMConfig.xml file for editing.

   This file is located in the ***`SAS-config-dir`*****`\Lev1\AppData\SASHumanCapitalManagement5.2`** directory.

2. Find this property:

   ```
   <Property Id="date_format_database" Name="Date format of database"
       Value="yyyy-MM-dd" ReadOnly="false"/>
   ```

3. Replace the value with the correct date format.

4. Save the file.

### *Set the Default Font for PDF Files*

If your site supports DBCS languages, set the font that is used when a user saves data to a PDF file from the Employee Browser, a geographic analysis, an organization analysis, or the general search results.

You can set this value on the **Configuration** tab of the Administration application in SAS Human Capital Management. For details, see "Configuring SAS Human Capital Management" on page 87 or the online Help for the **Configuration** tab.

### *Support User Locale Preferences*

Follow these guidelines to support user locale preferences:

- When users are logged on to SAS Human Capital Management, they can select a locale by clicking **Preferences**. For the best user experience, advise your users to select a locale to match the locale that you selected when you installed the SAS software. Otherwise, some messages might not be displayed in the localized text. Users should specifically select a locale, rather than selecting **Browser Default**.

  *Note:* In terms of data display, the user locale preference affects only stored process reports.

- In the MySQL HCM database, check the SAS_DEFAULT_PROPERTIES table to make sure that the value of PropFilePath is correct. This value should point to the **`!SASROOT\hrds\sasmisc`** directory on the data tier. It tells the %SETLOCS macro where to find the hcmtitles.properties and hcmlabels.properties files. These properties files are used for localization (for example, for titles in stored process output and for some of the labels in an OLAP cube).

  In deciding which properties file to use, the %SETLOCS macro first looks for an exact match to the user preference, including both the language-code and the country-code (for example, hcmtitles_ja_jp.properties). If an exact match cannot be found, the %SETLOCS macro looks for a match with the same language-code (for example, hcmtitles_ja.properties). If no match exists, the properties files for the installed HCM locale are used. For more information about these properties files, see "Managing the Data Sources" on page 6.

### Localize the Diagnostics Utility

*Note:* The user interface of the diagnostics utility currently supports only the English language. However, you can localize the test names by editing the diagnostics configuration files.

If you installed SAS Human Capital Management in a language other than English, you must modify the following server names in the diagnostics configuration files. Follow these steps:

1. Log on to SAS Management Console as the SAS Administrator.
2. On the **Plug-ins** tab, navigate to **Environment Management** ⇨ **Server Manager**.
3. Right-click **SAS Content Server** and select **Properties**.
4. On the **General** tab of the Properties dialog box, copy the name of the SAS Content Server and save it for later use.
5. Navigate to **Environment Management** ⇨ **Server Manager** ⇨ **SASApp**.
6. Copy and temporarily store the names of the following servers:
   - SASApp - Logical OLAP Server
   - SASApp - Logical Workspace Server
   - SASApp - Logical Stored Process Server
7. Open the HCMDiagnosticsConfig file for editing.

   This file is located in the ***`SAS-config-dir`*** **`\Lev1\Applications\SASHumanCapitalManagement5.2\Diagnostics`** directory on the middle tier.
8. Find the following line:

   ```
   <ContentServer name="SAS Content Server"/>
   ```
9. In that line, substitute the name of your site's SAS Content Server for **`SAS Content Server`**.
10. Find the following section:

    ```
    <LogicalServers>
       <WorkspaceServers>
          <Server name="SASApp - Logical Workspace Server"/>
           </WorkspaceServers>
       <StoredProcessServers>
          <Server name="SASApp - Logical Stored Process Server"/>
       </StoredProcessServers>
       <OLAPServers>
          <Server name="SASApp - Logical OLAP Server"/>
       </OLAPServers>
    </LogicalServers>
    ```
11. Substitute the names of the corresponding servers that you copied from SAS Management Console.
12. Save the file.

13. Open the PlatformDiagnosticsConfig.xml file for editing.
14. Repeat steps 8 through 11.
15. Save the file.

## *KPI Viewer Localization*

### *Modify the %SPMEXPSC Macro*

If your installation is in a language other than English, you must modify the SAS autocall macro %SPMEXPSC as follows:

1. Log on to SAS Management Console as the SAS Administrator.
2. On the **Plug-ins** tab, navigate to **Environment Management** ⇨ **Server Manager**.
3. Right-click **SAS Content Server** and select **Properties**.
4. On the **General** tab of the Properties dialog box, copy the name of the SAS Content Server and store it for later use.
5. Open the spmexpsc.sas file for editing.

   This file is located on the data tier, in the **`!SASROOT\scorecard\sasmacro`** directory (Windows) or the **`!SASROOT/sasautos`** directory (UNIX).

   *Note:* We recommend that you first make a backup copy of this file.
6. Find the following line:

```
ss = "omsobj:TCPIPConnection?TCPIPConnection[Source/ServerComponent
   [@Name='SAS Content Server' and
   @ClassIdentifier='DAC0D7F0-10DA-11D6-8816-AA0004006D06']]";
```

7. In that line, substitute the name of your site's SAS Content Server for **`SAS Content Server`**.
8. Make the same change to the following line:

```
ss = "omsobj:Directory?Directory[DeployedComponents/ServerComponent
[@Name='SAS Content Server' and @ClassIdentifier=
'DAC0D7F0-10DA-11D6-8816-AA0004006D06']]";
```

9. Save the file.

# Add-On Configuration

## *Introduction*

An add-on configuration occurs when you install the SAS Intelligence Platform (and possibly one or more solutions), and then install a solution at a later time. For example, you might install the SAS Intelligence Platform and SAS Strategy Management and later install SAS Financial Management as an add-on. Or you might install the SAS Intelligence Platform and later install SAS Human Capital Management.

When you install the add-on, you must perform some manual configuration steps, as described in this section.

## Import Data Source Definition Files for SAS BI Dashboard

As of SAS BI Dashboard 4.3, data source definition files (.DSX files) and contributor files (.CDX files) must be stored in the WebDAV repository. If you install SAS BI Dashboard and at a later time install a solution, that solution might have .DSX and .CDX files that are not stored in WebDAV. You might also create additional .DSX and .CDX files at a site.

If necessary, follow these steps to import the files. The files are typically located in subdirectories of ***SAS-config-dir*\Lev1\AppData\SASBIDashboard4.3** on the middle-tier machine where SAS BI Dashboard is installed.

1. On the **Folders** tab in SAS Management Console, navigate to **SAS Folders** ⇨ **System** ⇨ **Applications** ⇨ **SAS BI Dashboard** ⇨ **SAS BI Dashboard 4.3** ⇨ **DataSourceDefinitions** .
2. Right-click and select **Add Content From External File(s) or Directories**.
3. Select and import the .DSX file.

   The .DSX files are typically located in the ***SAS-config-dir*\Lev1\AppData \SASBIDashboard4.3\DataSourceDefinitions** on the middle-tier machine where SAS BI Dashboard is installed.
4. In SAS Management Console, navigate to **SAS Folders** ⇨ **System** ⇨ **Applications** ⇨ **SAS BI Dashboard** ⇨ **SAS BI Dashboard 4.3** ⇨ **ContributorDefinitions** .
5. Right-click and select **Add Content From External File(s) or Directories**.
6. Select and import the .CDX file.

   The .CDX files are typically located in the ***SAS-config-dir*\Lev1\AppData \SASBIDashboard4.3\ContributorDefinitions** directory on the middle-tier machine.

For more information, see “Working with Data Source XML (DSX) Files” in the *SAS Intelligence Platform: Web Application Administration Guide*.

# Chapter 29
# Assigning Groups and Roles

**Overview of Users, Groups, and Roles** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **363**
Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 363
Role Membership in SAS Solutions Services . . . . . . . . . . . . . . . . . . . . . . . . . . 364

**SAS Solutions Services Users, Groups, and Roles** . . . . . . . . . . . . . . . . . . . . . . . **364**
SAS Solutions Services User Identities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 364
SAS Solutions Services Groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 365
SAS Solutions Services Roles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 366

**SAS Human Capital Management Groups and Roles** . . . . . . . . . . . . . . . . . . . . . **367**
About SAS Human Capital Management Groups and Roles . . . . . . . . . . . . . . . . 367
SAS Human Capital Management Groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 367
SAS Human Capital Management Roles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 367
Example: SAS Human Capital Management Users . . . . . . . . . . . . . . . . . . . . . . . 368

**Registering Users** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **370**
About Registering Users . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 370
Synchronizing Users, Groups, and Roles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 370

## Overview of Users, Groups, and Roles

### Introduction

A metadata identity is created when you define an individual user or group in the User Manager plug-in to the SAS Management Console, or when you import user and group definitions from an enterprise source by using SAS bulk-load macros. The authorization facility uses identity metadata to define who is granted or denied permission to access a resource.

The users of a solution are typically the business users in a particular domain, such as finance or human resources. A site's administrator must load all of the appropriate information for each user who requires access to the solution. This chapter describes the default metadata identities (representing users, groups, and roles) that are required by SAS Solutions Services and the solutions, as well as the identities that need to be created on site.

Additional information:

- For detailed information about authentication and authorization, see the *SAS Intelligence Platform: Security Administration Guide*.

- The SAS Intelligence Platform configures a default set of users, groups, and roles during the deployment process. For information about those identities, see “Understanding the State of Your System” in the *SAS Intelligence Platform: System Administration Guide*.

These books are available at `http://support.sas.com/92administration`.

## Role Membership in SAS Solutions Services

### About Groups and Roles

It is important to understand the difference between groups and roles, as well as the privileges that each conveys. In SAS Solutions Services, group membership determines the content a user has access to, whereas role membership determines the actions a user can perform with this content. Role assignments can control the menus and links that are displayed in an application, and they can determine a user's ability to perform a task such as deleting a file in Document Manager or approving a form in a workflow.

Roles do not provide data security. Use role assignments to enable the actions that a user can perform and the menu items that are available to a user. Use data security, such as metadata permissions, to restrict the data that a user can access.

*Note:* Unlike groups, roles are not hierarchical; they do not inherit properties from other roles.

### How Role Permissions Are Enforced

Permissions that are based on roles are enforced in two different ways:

- **Document manager**. For each content type, such as WebDocument or ExcelReport, there is a defined set of actions, such as Move, AddtoPortlet, and Comment. In Document Manager, roles are granted permission to perform various actions based on content type. These pre-defined permissions should not be changed at a site.

  Each user of Document Manager must belong to one of these roles: Solutions Services: Information Consumer, Solutions Services: Analyst, or Solutions Services: System Administrator.

  *Note:* If a user belongs to none of these roles, then role permissions are not enforced in Document Manager. If a user belongs to more than one of these roles, results are undetermined.

- **The solutions**. In the solutions, roles are enforced by the application. Each application determines the functionality that is permitted to various roles.

# SAS Solutions Services Users, Groups, and Roles

## SAS Solutions Services User Identities

SAS Solutions Services has two default user identities: Solutions Host User and SAS Solutions Administrator.

| User | Description |
|---|---|
| Solutions Host User | This identity has JDBC access to the MySQL databases, depending on group membership and operating system permissions. It is an external identity. |
| SAS Solutions Administrator | The SAS Solutions Administrator is an internal identity that is used for cases in which a user must perform a query as a part of a larger process, but the query requires a role that the user does not generally need. Rather than requiring the user to be assigned to that role, the application recognizes the SAS Solutions Administrator as a user with the proper role to successfully complete the process.<br>In a system that was migrated from SAS 9.1.3, this user is called Solutions Role Administrator. |

## SAS Solutions Services Groups

To log on to Document Manager, users must belong to the Solutions Users group or the Administrators group.

***Table 29.1*** *SAS Solutions Services Groups*

| Group | Description |
|---|---|
| Solutions Users | Members of the Solutions Users group are able to access the Document Manager, are configured to run SAS Solutions Services stored processes, and have default portal customization capabilities, such as adding a portlet or modifying its content.<br>Any users who run metric jobs or access the metric tables must be a member of the Solutions Users group, which has access to the SDMMET library. |
| Administrators | Membership in this group grants certain default privileges in Document Manager. It also grants superuser status in SAS Financial Management Studio. Membership in this group should be granted sparingly. |
| SASSDM MySQL Users | This group grants access to users who run stored processes and ETL processes that reference MySQL tables in the SASSDM database. The group has a login to the SASSDM database.<br>*Note:* The Solutions Users group is a member of SASSDM MySQL Users. |

Additional groups that are associated with roles are listed in Table 29.2 on page 366.

You can add users directly to a group, or you can make the assignment indirectly, via one or more custom groups. For more information about metadata groups, see the *SAS Intelligence Platform: Security Administration Guide*.

*Note:* In SAS Solutions Services 1.4, the Administrators group was a subgroup of Solutions Users. In a system that was migrated from SAS 9.1.3, that is still the case.

If you want to emulate that behavior on a new system, add the Administrators group to the Solutions Users group. Otherwise, assign members of the Administrators group to the Solutions Users group.

## SAS Solutions Services Roles

Each user who logs on to Document Manager must have one of these roles:

- Solutions Services: Information Consumer
- Solutions Services: Analyst
- Solutions Services: System Administrator

Assign additional roles as necessary. See the table below.

*Note:* Rather than assigning users directly to a role, assign users to the corresponding group, if one exists.

***Table 29.2*** *SAS Solutions Services Roles and Corresponding Groups*

| Group Name (if one exists) | Role Name | Display Name | Description |
|---|---|---|---|
| Analyst Group | Analyst | Solutions Services: Analyst | In Document Manager, users with this role can view, edit, move, and delete authorized content. Users with this role cannot delete folders.<br>In SAS Strategy Management and in SAS Financial Management, this role confers an additional set of privileges. The role is the same, but its functionality depends on the application that is being used. See later sections of this chapter for details. |
| Not applicable | Dimension Modeler | Solutions Services: Dimension Administrator | This role gives users access to the SAS Solutions Dimension Editor, a Java client application for creating and modifying dimensions and hierarchies. |
| Data Administrators | Data Administrator | Solutions Services: Data Administrator | Users who run jobs in SAS Data Integration Studio should have this role. |
| Information Consumer Group | Information Consumer | Solutions Services: Information Consumer | In Document Manager, users with this role can view content. These users cannot create, move, or delete content. |
| Solutions Administrators Group | System Administrator | Solutions Services: System Administrator | Users with this role have access to all functionality within Document Manager. |

# SAS Human Capital Management Groups and Roles

## About SAS Human Capital Management Groups and Roles

SAS Human Capital Management users must meet the following qualifications:

- membership in the HCM Solution Users group
- membership in one of the following roles:
  - HCM User
  - HCM Analyst
  - HCM Administrator

  *Note:* Assign users, not groups, to SAS Human Capital Management roles.
- a valid entry in the SAS_USER_EMPLOYEE table of the HCM database

  For information about the ETL job that loads the SAS_USER_EMPLOYEE table, see the Data Administration part of this book. You can also update this table manually, from the Administration application of SAS Human Capital Management. See the Administration Application part of this book for details.

## SAS Human Capital Management Groups

The following groups are part of SAS Human Capital Management:

***Table 29.3*** *SAS Human Capital Management: Groups*

| Group | Description |
|---|---|
| HCM Users | To log on to SAS Human Capital Management, users must belong to the HCM Solution Users group. |
| HCM MySQL Users | This group grants access to users who run stored processes and ETL processes that reference MySQL tables in the HCM database. It has a default login to the HCM database. |
| HR | The HR group is the default power user group. Members of this group are not subject to hierarchical filters, which are part of row-level security in SAS Human Capital Management. (Additional filters for users, groups, or roles still apply.) Each site can define its own power user group.For details, see "Securing Objects and Tables" on page 66. |

## SAS Human Capital Management Roles

SAS Human Capital Management users must be a member of one of these roles:

***Table 29.4*** *SAS Human Capital Management: Roles*

| Role | Description |
|---|---|
| HCM User | Users with the HCM User role view employee, organizational, and geographic data, and create presentations and reports. These users can print information but cannot export it. They cannot create a document such as a geographic analysis, organization analysis, or a SAS report.<br>Users with the HCM User role have these capabilities:<br>• Employee Browser: view employee detail (profile view), search for employees, and edit the category list.<br>• organization analysis: open and print organization charts; launch a linked scorecard; create a presentation view.<br>• geographic analysis: open a geographic analysis document and drill down into the content; print a map or employee list.<br>• general search: conduct a simple, advanced, or history search. |
| HCM Analyst | Users with the HCM Analyst role create the documents that are viewed by other users, including organization and geographic analyses. They can print, save, and export data.<br>Users with the HCM Analyst role have these capabilities:<br>• Employee Browser: all functions.<br>• organization analysis: all functions. In addition to the HCM User privileges, these users can add and remove measures, create a new organization analysis, modify the display options, and simulate a reorganization.<br>• geographic analysis: all functions. These users can create or open a geographic analysis document and drill down into the content; print a map or employee list.<br>• general search: all functions, including saving and exporting search results. |
| HCM Administrator | Users with the HCM Administrator role configure SAS Human Capital Management and manage data security. These users have full access to all functionality within SAS Human Capital Management.<br>In addition to the tasks that are described for the HCM Analyst role, these users can perform administration and configuration tasks, including importing tables, mapping hierarchies, configuring application defaults, and creating employee profiles. |

## Example: SAS Human Capital Management Users

The following list includes some hypothetical users at a SAS Human Capital Management site: general users (managers), report creators, HR analysts, and administrators, along with the groups and roles they might belong to.

*Note:* Membership in the appropriate groups and roles does not guarantee access to data or actions. For more information, see "Securing Objects and Tables" on page 66.

***Table 29.5*** *SAS Human Capital Management: Typical Users*

| **Typical Users** | **Tasks** | **Groups and Roles (Optional*)** |
|---|---|---|
| Managers | These managers view and interact with reports. They can log on to SAS Human Capital Management, browse employees, and interact with a geographic analysis or organization analysis. They can print but cannot save data from those applications. They can also execute the standard stored processes.<br>Optional:<br>• With membership in the Scorecard Data Entry Group, these users can view KPIs and scorecards (depending on permissions).<br>• With the Financial Management: Form Submitter role or the Financial Management: Form Approver role, users can participate in the budgeting process. (This capability requires SAS for Workforce Planning & Budgeting.)<br>• With the appropriate role, users can view and interact with reports in SAS Web Report Studio.<br>• With the appropriate permissions, users can view a BI Dashboard on the Home page or in a portlet. | Groups:<br>• HCM Solution Users<br>• Scorecard Data Entry Group<br>Roles:<br>• HCM User<br>• Financial Management: Form Submitter*<br>• Financial Management: Form Approver* |
| Report Creators | These users create reports for others to view and interact with. They can log on to SAS Human Capital Management and browse employees, create a geographic analysis, and create an organization analysis. They can save or export data from those applications. They can also execute the standard stored processes.<br>Optional:<br>• With membership in the Scorecard Modeler Group and the SPM Users group, these users can create and manage KPIs and scorecards. With membership in the Analyst Group, they can customize a scorecard or KPI but cannot create or manage them.<br>• With the appropriate role and permissions, users can create and administer dashboards in SAS BI Dashboard.<br>• With the appropriate role, users can create reports in SAS Web Report Studio. | Groups:<br>• HCM Solution Users<br>• SPM Users<br>• Analyst Group or Scorecard Modeler Group*<br>Roles:<br>• HCM Analyst |
| HR analysts | These users perform statistical analyses of the data in SAS Human Capital Management. Because they need access to all the data, they might need to belong to the HR (superuser) group. Membership in the HR group enables a user to access records for all employees, regardless of hierarchical filters. However, other filters might apply.<br>*Note:* Because some data is quite sensitive, use care in assigning membership in the HR group.<br>If these users need to create reports as well as analyze them, they need the same groups and roles as report creators. | Groups:<br>• HCM Solution Users<br>• Scorecard Data Entry Group<br>Roles:<br>• HCM User |

| Typical Users | Tasks | Groups and Roles (Optional*) |
|---|---|---|
| Administrators | These users have access to the full functionality of SAS Human Capital Management. They might perform several types of administrative tasks at a site, including content administration, data administration, and IT administration. If the site has SAS for Workforce Planning & Budgeting, these users can create planning measures for use in the budgeting process.<br>Optional:<br>• With membership in the Data Administrators group, the users can run jobs in SAS Data Integration Studio.<br>• If these users need to create or modify reports, they will also require the additional groups and roles that report creators have.<br>• Any users who run metric jobs or access the metric tables must be a member of the Solutions Users group, which has access to the SDMMET library. | Groups:<br>• HCM Solution Users<br>• Administrators<br>• Solutions Users*<br>• Data Administrators<br>Roles:<br>• HCM Administrator |

For information about roles and permissions for SAS BI Dashboard, see “Managing Security for SAS BI Dashboard” in the *SAS Intelligence Platform: Web Application Administration Guide*.

For information about roles and permissions for SAS Web Report Studio, see “Managing SAS Web Report Studio Content and Users” in the *SAS Intelligence Platform: Web Application Administration Guide*.

# Registering Users

## About Registering Users

For information about registering users, see the *SAS Intelligence Platform: Security Administration Guide*.

When you define a user, be sure to include the user's e-mail address. E-mail notifications are often sent to users. For the successful processing of some functions, you must define an e-mail address for every user.

*Note:* On Windows, in order for users to access a standard workspace server using credential-based host authentication, they need the local **Log on as a batch job** right on that machine. For more information, see “Windows Privileges” in the *SAS Intelligence Platform: Security Administration Guide*.

## Synchronizing Users, Groups, and Roles

### About Synchronizing Users, Groups, and Roles

*Note:* These procedures are not required for SAS Human Capital Management.

Information for users, groups, and roles is stored in database tables that must be kept in synchronization with the metadata. As a part of best practices, it is recommended that you set up a SAS Data Integration Studio job as a scheduled process to synchronize data tables.

### Running a Batch Job

The typical way to update user and group assignments is by using a batch job:

1. In SAS Data Integration Studio, create a batch job to update the Solutions Data Mart tables.
2. Then, schedule this job to be performed on a regular basis. Include these three jobs:
   - Load Users
   - Load Groups
   - Load User_x_Group

For more information about these jobs, see the Data Administration part of this book. For information about creating batch jobs and about scheduling jobs, see the *SAS Intelligence Platform: System Administration Guide*.

### Clear User Cache

The **Clear user cache** utility is used to clear a cache that is used only by SAS Financial Management planning security. For performance reasons, roles are cached when the Web application server is started. If you have changed any role assignments in SAS Management Console, you must flush the cache in order to use the new assignments when you assign form authors or reviewers in SAS Financial Management Studio.

To flush the cache without restarting the Web application server, follow these steps:

1. In the My Favorites portlet, click **Clear user cache**.

   If this task is not already available, add it to a My Favorites portlet. For more information about the My Favorites portlet, see the online Help for SAS Solutions Services.

2. On the page that is displayed, click **Clear user cache**.

# Chapter 30
# Portal and Content Administration

**Portal Administration** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **373**
About Customizing the Portal . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 373
My Favorites Portlets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 373
Alerts Portlets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 374
KPI and Scorecard Portlets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 374

## Portal Administration

### About Customizing the Portal

This chapter includes information about customizing the portal for the solutions, including the portlets that are provided with SAS Solutions Services. For detailed information about portal administration, consult the following references:

- the online Help for the SAS Information Delivery Portal
- the *SAS Intelligence Platform: Web Application Administration Guide*, available at `support.sas.com/92administration`

These references explain how to administer portal content, how to create page templates, and how to administer portal authorization.

*Note:* Membership in the Portal Admins group is no longer recommended.

### My Favorites Portlets

The My Favorites portlet has many uses. The portlet allows users to create lists of documents, files, folders, links, and tasks.

The following tasks can be added to a My Favorites portlet:

***Table 30.1*** *Tasks for a My Favorites Portlet*

| Task | Description |
|---|---|
| **Manage Strategy Management Scorecard Projects and Templates** | Opens the KPI Viewer. |

| Task | Description |
|---|---|
| **New Strategy Management Scorecard Project** | Opens the KPI Viewer. |
| **Human Capital Management Home** | Opens the Home page of SAS Human Capital Management. |

## Alerts Portlets

An alert is a notification of an event that the user might need to respond to. Opt-in alerts are alerts that users choose to receive by setting properties on a document or a folder. For example, a user might ask to be informed of a document being added to a folder, or of a comment being added to a document.

Workflow alerts are notifications of tasks that the user has to perform, such as approving a budget form. Users cannot choose not to receive these alerts.

To add an Alerts portlet to a page, follow these steps:

1. From the **Customize** menu, select **Edit Page** ⇨ **Edit Page Content**.
2. On the Edit Page Content page, select **Add Portlets** and add an Alerts portlet.

For more information about adding portlets to a page, see the online Help for the portal.

## KPI and Scorecard Portlets

The following portlets are available for KPIs and scorecards. (You must have licensed SAS Strategy Management to view scorecard elements.)

***Table 30.2*** *Portlets Available with SAS Solutions Services*

| Portlet Type | Description |
|---|---|
| Performance Dashboard portlet | Displays KPIs and scorecard elements in graphical format. Each element is represented by a dashboard that displays, in graphical format, the data ranges that have been defined. In addition to ranges, you can display comments, history data, and element properties from a dashboard. |
| Performance Data Entry portlet | Displays a Web data-entry form. |
| Performance Table portlet | Displays data for the selected KPI or scorecard in tabular form. |
| Strategy Management portlet | Displays multiple strategy views in a single portlet.<br>KPI data can be displayed in the Table View, Aggregate View, Gauge View, and Trend Analysis tiles of this portlet.<br>Scorecard data can be displayed using any of its tiles, including Association View and Scorecard Hierarchy. |

If you have licensed SAS Strategy Management, the following portlets are also available:

***Table 30.3*** *Portlets Available with SAS Strategy Management*

| Portlet Type | Description |
|---|---|
| Performance Aggregate Table portlet | Displays data for the selected scorecard and all of its children. |
| Performance Association portlet | Displays the hierarchical relationship between scorecard elements of a single scorecard or project. |
| Performance Diagram portlet | Displays data in the form of diagrams, to illustrate the relationships between elements. The data can be based on project element types or scorecard element types. |

For information about these portlets, see the online Help.

# Chapter 31
# Administering the Middle Tier

**Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **377**
About the Managed Servers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 377
Additional Information . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 378
**Oracle WebLogic Server Modifications** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **378**
**IBM WebSphere Server Modifications** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **379**
Set Total Transaction Lifetime Time-out . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 379
Increase the Log File Size . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 380

## Overview

### *About the Managed Servers*

Depending on the solutions that you installed and your choices during the configuration steps, your system will have several managed servers. The following table shows the default servers, listen ports, and a partial listing of the contents, consisting of enterprise archive (EAR) files. For complete information, see the WebLogic Console and the ***SAS-config-dir*\Lev1\Web\Common** directory.

| Server (Default Name) | Partial Contents (EAR Files) | Default Listen Port |
|---|---|---|
| SASServer1 | The Web Infrastructure Platform (WIP), the Logon Manager, the SAS Information Delivery Portal, SAS BI Dashboard | 7001 |
| SASServer2 | The SAS WebDoc application, SAS Web Report Studio, SAS Web OLAP Viewer | 7101 |
| SASServer3 | SAS Solutions Services and the solutions | 7201 |
| SASServer4 | ODCS | 7301 |
| SASServer5 | Planning Data Entry | 7401 |

You might also install one or more secondary ODCS servers on additional managed servers.

The SAS Remote Services application must be started before the managed servers. If you restart SASServer1 (where sas.wip.services9.2.ear is deployed), you must restart the other managed servers as well.

### Additional Information

For detailed information about administering Web applications and the Web application server, see the *SAS Intelligence Platform: Web Application Administration Guide*, available at `http://support.sas.com/92administration`. That book also contains information about topics such as the following:

- tuning the Web application servers
- administering the SAS remote services
- installing a custom theme and setting the default theme
- modifying application configuration properties in the Configuration Manager plug-in of SAS Management Console
- administering the SAS Information Delivery Portal, SAS Web Report Studio, and SAS BI Dashboard
- WebDAV administration

*Note:* WebLogic clustering is not supported by SAS Solutions Services and the solutions.

For additional information, see *SAS 9.2 Web Applications: Tuning for Performance and Scalability* at `http://support.sas.com/resources/thirdpartysupport/v92`. This document includes information about performance tuning and setting JVM options.

## Oracle WebLogic Server Modifications

The following modifications might be necessary for the Oracle WebLogic managed servers, depending on your system configuration:

- **URL Mapping:** WebLogic appears to treat domains differently if they are referenced differently (for example, **http://Dxxx/yyy** and **http://Dxxx.mycompany.com/yyy**). This situation causes problems when a Web application stores information in the HttpSession context. There is a configuration parameter called **Frontend Host** that addresses this issue. According to the WebLogic documentation, this parameter should be set when the Host information coming from the URL might be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is used instead.

  The **Frontend Host** parameter is part of the HTTP Protocols for a managed server. For instructions about modifying these protocols, see the WebLogic documentation.

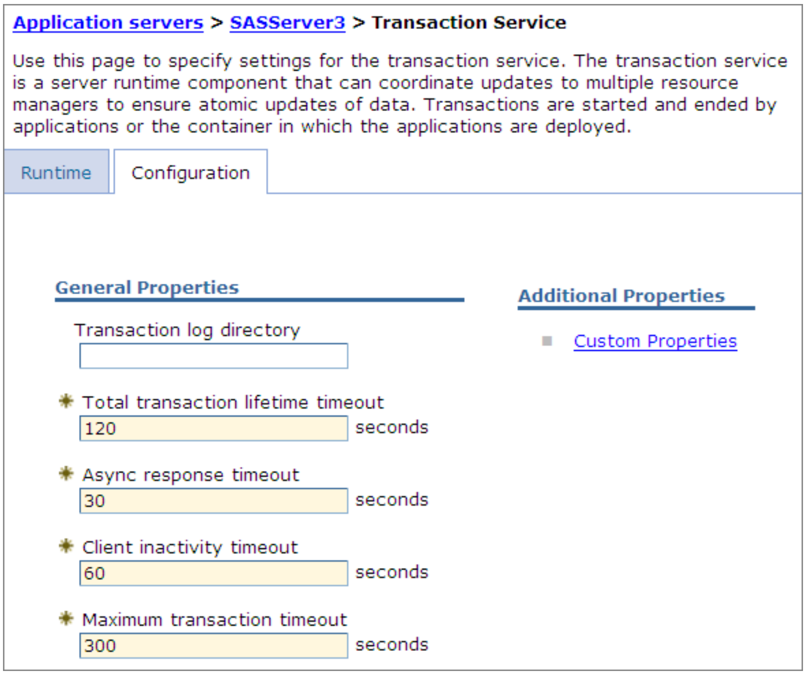
- **If you installed SAS Human Capital Management:** To ensure best performance, restart the managed servers, as well as the SAS application servers, once a week.

# IBM WebSphere Server Modifications

## Set Total Transaction Lifetime Time-out

The total transaction lifetime time-out value specifies the length of time, in seconds, for a transaction to be completed before it is rolled back. The default value that is configured by the solutions is 120 seconds. If your configuration needs additional time, increase this value as follows:

1. Log on to the WebSphere administrative console.
2. In the navigation tree, select **Servers** ⇨ **Application Servers** (WebSphere 6) or **Servers** ⇨ **Server Types** ⇨ **WebSphere Application Servers** (WebSphere 7).
3. On the Application Servers page, click the name of the managed server for SAS Solutions Services and the solutions (typically, SASServer3).
4. Under **Container Services**, select **Transaction Service**.



Application servers > SASServer3 > Transaction Service

Use this page to specify settings for the transaction service. The transaction service is a server runtime component that can coordinate updates to multiple resource managers to ensure atomic updates of data. Transactions are started and ended by applications or the container in which the applications are deployed.

Transaction log directory

* Total transaction lifetime timeout: 120 seconds

* Async response timeout: 30 seconds

* Client inactivity timeout: 60 seconds

* Maximum transaction timeout: 300 seconds

5. Specify a new value for **Total transaction lifetime timeout**.

   You might need to experiment by increasing the value considerably. If that works, try decreasing the value until you find a limit that works at your site.

6. If necessary, modify the value for **Maximum transaction timeout**.

   This value specifies the length of time, in seconds, for all transactions that run in this server. Its value should be greater than or equal to the total transaction lifetime time-out value.

7. If necessary, modify the value for **Client inactivity timeout**.

   This value specifies the length of time, in seconds, between transactional requests from a remote client.

8. Save your changes.

9. Make similar changes to the ODCS managed server (typically, SASServer4).
10. Restart SASServer3, the ODCS managed servers, and the Web Data Entry (WDE) managed server (typically, SASServer5).

For more information about these parameters, see the online Help for the WebSphere administrative console.

### Increase the Log File Size

The default sizes for WebSphere log files and history files might be too small to capture substantial logging. To change the log settings:

1. Log on to the WebSphere administrative console.
2. In the navigation tree, select **Troubleshooting Logs and Trace**.
3. On the Logging and Tracing page, click the server name.
4. Click **JVM Logs**.
5. For the System.out log, find the **File Size** for the **Log File Rotation**, and change the **Maximum Size** from 1 MB to 10 MB. You can adjust this value to suit your configuration.
6. To save log files that have been rotated, increase the value of **Maximum Number of Historical Log Files**.
7. Make the same changes for the System.err log.
8. Save your changes.

# Chapter 32
# Viewing and Configuring the Log Files

**Log File Locations and Configuration Files** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **381**
**Dynamically Configuring Logging Levels** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **382**
SAS Human Capital Management . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 382
**Additional Log Files** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **382**

## Log File Locations and Configuration Files

The following log files are located on the middle tier, where you installed the Web application server.

***Table 32.1*** *Middle-Tier Log Files*

| Application | Log File Information |
|---|---|
| SAS Solutions Services<br>SAS Human Capital Management | The logs for these applications are configured using the *application*-log4j.xml files in the ***`SAS-config-dir`*****`\Lev1\Web\Common\LogConfig`** directory.<br>The default log file location is ***`SAS-config-dir`*****`\Lev1\Web\Logs`**. |
| Remote Services | For local services and the remote services use the logging framework from the SAS Web Infrastructure Platform. You can modify the logging configuration in the Foundation Services Manager of SAS Management Console. For more information, see “Administering SAS Web Applications” in the *SAS Intelligence Platform: Web Application Administration Guide*. The default log file location is ***`SAS-config-dir`*****`\Lev1\Web\Logs`**. |
| Oracle WebLogic servers | By default, the log files for the Oracle WebLogic servers are located in subdirectories under ***`SAS-config-dir`*****`\Lev1\Web\SASDomain\servers\`*****`server-name`*****`\logs`** directory, where *server-name* is the name of the managed server. Output to the WebLogic console is written to these log files. |

| Application | Log File Information |
|---|---|
| IBM WebSphere servers | By default, the log files for the IBM WebSphere servers are located in the `WebSphere-install-dir/profiles/profile-name/logs/server-name` directory. |

## Dynamically Configuring Logging Levels

### *SAS Human Capital Management*

For SAS Human Capital Management, you can dynamically change the logging levels on the **Configuration** tab of the SAS Human Capital Management Administration application. For details, see “Configuring SAS Human Capital Management” on page 87.

## Additional Log Files

- For information about log files for other Web applications, such as SAS Web Report Studio, see the *SAS Intelligence Platform: Web Application Administration Guide*.
- For information about log files that are generated by the SAS servers, see “Enabling Server Logging” in the *SAS Intelligence Platform: System Administration Guide*.

Both books are available at `support.sas.com/92administration`.

*Chapter 33*

# Administering MySQL Server

**MySQL Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **383**

**Windows: MySQL Installation and Configuration** . . . . . . . . . . . . . . . . . . . . . . . . **383**
Installing and Configuring the MySQL Server . . . . . . . . . . . . . . . . . . . . . . . . . . 383
Reconfiguring MySQL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 383

**UNIX: MySQL Installation and Configuration** . . . . . . . . . . . . . . . . . . . . . . . . . . . **384**

## MySQL Overview

SAS Solutions Services stores common data in a MySQL database that is created during the installation process. Support for INNODB tables must be enabled within MySQL to provide transaction support, which is required by a number of SAS Solutions Services components.

## Windows: MySQL Installation and Configuration

### *Installing and Configuring the MySQL Server*

For installation and configuration information, see the readme.pdf file that is located in the SAS Software Depot, in the appropriate subdirectory of the **`third_party\MySQL_Database_Server\5_0_86`** directory.

### *Reconfiguring MySQL*

The MySQL server is configured to read its configuration settings from the ***`MySQL-install-dir`*****`\my.ini`** configuration file. If you need to adjust your MySQL configuration, you can modify these configuration settings in the MySQL Administrator, or you can edit the my.ini file directly. Before you make any changes, be sure to make a backup copy of the my.ini file. After making your changes, restart the service.

The MySQL client reads its configuration information from a copy of the my.ini file that is located in the Windows root directory (for example, **`C:\WINNT\my.ini`**). If you modify the ***`MySQL-install-dir`*****`\my.ini`** file, be sure to copy your modified file to the Windows root directory.

## UNIX: MySQL Installation and Configuration

The path to the MySQL executable (typically, **/usr/local/mysql/bin**) must be on the user's path. For information about installing and configuring MySQL on UNIX, see the readme.pdf file that is located in the SAS Software Depot, in the appropriate subdirectory of the **third_party\MySQL_Database_Server\5_0_86** directory.

Excessive I/O in MySQL might be an indicator that sorts are not occurring in memory. In that case, you might consider increasing sort_buffer_size. See "Tuning Server Parameters" in the MySQL Reference Manual for considerations in changing this parameter, which is allocated per thread. As a test, you might temporarily set sort_buffer_size as high as 32M. However, a lower permanent setting might be more appropriate. To modify a parameter setting, you edit the ***MySQL-install-dir/*****my.cnf** file and restart the MySQL server.

At Solaris sites with heavy data usage, you can improve performance by editing the my.cnf file to set the thread_concurrency value. This value is used in determining the number of threads to be run simultaneously. The recommended value is as follows:

```
number-of-cpus * (2..4)
```

# Part 4

# The Data Model

*Chapter 34*
**Introduction to the Data Model** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *387*

*Chapter 35*
**Descriptions of DDS Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *389*

*Chapter 36*
**Descriptions of DDS Table Columns** . . . . . . . . . . . . . . . . . . . . . . . . . . . . *401*

*Chapter 37*
**More Information about DDS Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . *509*

*Chapter 38*
**Solutions Data Mart Jobs** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *571*

*Chapter 39*
**HCM Data Mart Jobs** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *573*

*Chapter 40*
**HCM Data Mart Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . *591*

*Chapter 41*
**Descriptions of HCM Data Mart Table Columns** . . . . . . . . . . . . . . . . . . *597*

*Chapter 34*

# Introduction to the Data Model

**Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 387

## Introduction

This data model reference contains supplementary information that you might find helpful as you follow the instructions in the Data Administration part of this book.

# *Chapter 35*
# Descriptions of DDS Tables

**Detail Data Store Table Descriptions** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 389

## Detail Data Store Table Descriptions

The following table contains descriptions of all the tables in the SAS Cross Industry Detail Data Store. The tables are listed in alphabetical order.

| Name | Comment |
|---|---|
| ABSENCE_TYPE | This table defines the types of absence that an employee can take. Examples are sickness, vacation, disability, family leave, and bereavement. This table is used to build the HCM formats for ABSENCE_TYPE_CD. |
| ACADEMIC_CREDIT | This table defines types of credit that are awarded by educational institutions. Examples: A = adult credits, C = continuing education units, N = no credit, Q = quarter hour credit. This table will be used by the future SAS Human Capital Management Education Model. |
| ACADEMIC_HONORS | This table defines academic honors. Example: magna cum laude. This table will be used by the future SAS Human Capital Management Education Model. |
| ACTION_REASON | This table describes the reason for a specific action associated with an employee. This table is used to build the SAS Human Capital Management formats for ACTION_REASON_CD. |
| ACTION_TYPE | This table defines the type of employee action, such as pay increase, probation, or suspension. This table is used to build the SAS Human Capital Management formats for ACTION_TYPE_CD. |
| ADDRESS_TYPE | This table contains address types, such as business, shipping, or residential. |

| Name | Comment |
|---|---|
| APPLICATION_STATUS | This table contains status codes for employment applications. This table is used to build the SAS Human Capital Management formats for APPLICANT_STATUS_CD. |
| ATTENDANCE_STATUS | This table defines attendance statuses, such as "Current" or "Prior." This table will be used by the future SAS Human Capital Management Education Model. |
| CODE_LANGUAGE | This table contains a row for each language/locale used throughout the DDS. It is used as input into the SDM. This table is required for SAS Human Capital Management. |
| COMPENSATION | This table defines a history of additional compensation paid to an employee (other than base pay). Examples are bonuses and one-time awards. Required table when populating SAS Human Capital Management compensation history data. |
| COMPENSATION_TYPE | This table defines types of additional compensation, such as commissions or bonuses. This table is used to build the SAS Human Capital Management formats for COMPENSATION_TYPE_CD. |
| COMPETENCY | This table defines competencies. A competency is a specific identifiable, definable, and measurable skill, ability, knowledge, or other characteristic necessary for the performance of an activity or job within a specific business context. Competencies are measurable and can be quantifiable. Examples include a requirement for Java Programming, Masters Degree, a willingness to relocate as identified by the organization. Through the use of the other COMPETENCY tables, competencies are identified overall, assigned, and tracked for the jobs and employees. Competencies can be recursive. A competency can include other competencies, up to one level. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_ASSOC | This association table contains the parent/child relationships that make up the Competency hierarchy. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_ASSOC_TYPE | This table identifies the COMPETENCY hierarchies. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_CATALOG | This table defines catalogs that are used to track competencies. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_CATEGORY | This table defines categories that are used to track competencies. This table will be used by the future SAS Human Capital Management Competency Model. |

| Name | Comment |
|---|---|
| COMPETENCY_CATEGORY_CLASS | This table contains the type of the competency. Types can include skills, abilities, and knowledge. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_CLASS | This table defines classes that are used to track competencies. Classes can include Business, Communication, Humanities, and so on. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_EVIDENCE | This table is used to capture information, or evidence, required for the competency. Examples are performance appraisals, reports, or certificates. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_TYPE | This table defines competency classes. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_WEIGHT_BOUNDS | This table contains the threshold bounds for the competency by type. Bounds can be numeric or a string value. For example, a Required (type) for Java programming can have a range bound of 1 to 5 with an interval of 1. This means that the job or employee can be rated a 1, 2, 3, 4, or 5. Non-numeric bounds examples are Interested or Not Interested. This table will be used by the future SAS Human Capital Management Competency Model. |
| COMPETENCY_X_TAXONOMY_SOURCE | This table is a crosswalk between the COMPETENCY column in the COMPETENCY table and the TAXONOMY_ID column in the TAXONOMY table. This mapping is used to associate internal competencies with external competencies for comparison and measurement. This table will be used by the future SAS Human Capital Management Competency Model. |
| CONTACT_ROLE | This table defines contact roles. |
| COST_CENTER | This table contains a list of categories that identify the division of functional areas within an organization. These categories are typically entities to which costs are assigned. The table includes child and parent members. |
| COST_CENTER_ASSOC | This table contains the parent/child relationships that make up the COST_CENTER hierarchies. |
| COST_CENTER_ASSOC_TYPE | This table identifies the COST_CENTER hierarchies. |
| COST_CENTER_NLS | This table supplements the COST_CENTER table with localized member names and descriptions. |

| Name | Comment |
|---|---|
| COUNTRY | This table contains two-character country codes that conform to the ISO 3166 standard. |
| COUNTY | This table contains codes used to identify counties in a state or region. This table is used to build the SAS Human Capital Management formats for |
| COUNTY_CD. | Information in this table used by SAS Human Capital Management Geographic Analysis. |
| COURSE_LEVEL | This table defines course levels that are established by the National Center for Education Statistics to indicate the level or type of work, which is reflected in the grade average and the credit hours. Examples: 1 = Remedial; 2 = Basic; 3 = Teacher's Aide; 4 = General; A = Summary of all. SAS Human Capital Management reference table for the NCES information. |
| CURRENCY | This table contains the list of standard three-character ISO 4217 codes used for identifying currency codes. Required by SAS Human Capital Management. |
| DATES_OF_ATTENDANCE | This table captures the period during which a person attended a school or institution. This table will be used by the future SAS Human Capital Management Education Model. |
| DDS_SOLUTIONS_VERSION | This table contains the details from loading various types of data: version of the data, load method of the data (installation or migration), and the status of the load (successful or failed). |
| DEGREE_CONCENTRATION | This table defines concentrations within a major or field of study. Example: masters degree in business administration with a concentration in personnel. This table will be used by the future SAS Human Capital Management Education Model. |
| DEGREE_OPTION | This table contains the official names of major degree options. Example: A particular institution might require their Civil Engineering majors to select either a Mechanical Engineering option or a Structural Engineering option. This table will be used by the future SAS Human Capital Management Education Model. |
| DEGREE_PROGRAM | This table defines degree programs. This table will be used by the future SAS Human Capital Management Education Model. |
| DEGREE_TYPE | This table defines degree types. Examples: doctorate, masters, bachelors, associate certificate. This table will be used by the future SAS Human Capital Management Education Model. |

| Name | Comment |
|---|---|
| EDUCATION_HISTORY | This table lists the employee's history of formal education and certification. Examples: High School, College, University, or Technical School. This table will be used by the future SAS Human Capital Management Education Model. |
| EDUCATION_LEVEL | This table defines the employee's level of education. Required table when tracking SAS Human Capital Management education history. |
| EDUCATION_VALUE | This table contains the measurement standards relevant for tracking an employee's education history. This includes the minimum and maximum bounds and the employee's value. Examples of values tracked are GPA and class rank. Tracked values can be numeric or string. This table will be used by the future SAS Human Capital Management Education Model. |
| EDUCATION_VALUE_SYSTEM | This table defines scales against which the educational measure can be compared. Business rules: For a GPA, this might contain the highest possible GPA attainable at this institution. For a class rank, this might contain the size of the class. This table will be used by the future SAS Human Capital Management Education Model. |
| EDUCATION_VALUE_TYPE | This table contains information measuring the educational performance of the person. Examples: If the measurement is grade point average (GPA), then this might contain a value of 3.5. If the measurement is rank in class, this might contain a value of 130 (ranked 130 out of class of 5000). This table will be used by the future SAS Human Capital Management Education Model. |
| EEO_CLASS | This table contains the standard EEO classification codes. EEO1 classifications are used for corporations and EEO4 are used for government reporting at the federal, state and local level. This table is used to build the SAS Human Capital Management formats for EEO_CLASS_CD. Many standard SAS Human Capital Management reports use this information. |
| EMPLOYEE | This table defines current and historical information pertaining to employees. Required table for SAS Human Capital Management. |
| EMPLOYEE_ABSENCE | This table defines and tracks the information about an employee absence. Required table when populating SAS Human Capital Management absence history data. |
| EMPLOYEE_ACTION | This table contains particular employee actions, such as pay increases, suspensions, or probations of particular employees. This table represents valid combinations of Action Type Codes and Reason Type Codes. An employee has to have at least 1 record in this table to be counted in headcount from that start date forward. Required table for SAS Human Capital Management. |

| Name | Comment |
|---|---|
| EMPLOYEE_INVOLVEMENT_TYPE | This table is not used. |
| EMPLOYEE_STATUS | This table defines the employment status of an employee. For example: active, terminated, inactive. This table is used to build the SAS Human Capital Management formats for EMPLOYEE_STATUS_CD. |
| EMPLOYEE_TYPE | This table contains unique codes to indicate an employee's employment type. This table is used to build the SAS Human Capital Management formats for EMPLOYEE_TYPE_CD. |
| EMPLOYEE_UNION | This table contains codes for labor unions. This table is used to build the SAS Human Capital Management formats for UNION_TYPE_CD. |
| EMPLOYEE_X_COMPETENCY | This table is the intersection of the EMPLOYEE table and the COMPETENCY table. This table contains the competencies held by an employee independent of any job he or she holds. This table will be used by the future SAS Human Capital Management Competency Model. |
| EMPLOYEE_X_EVIDENCE | The intersection of the EMPLOYEE table and the COMPETENCY_EVIDENCE table. This table contains the information about the evidence the employee has for a competency. For example, a Medical Doctor has to have a license. The information in this table would contain the information supporting the evidence of that license. This table will be used by the future SAS Human Capital Management Competency Model. |
| EMPLOYEE_X_INTERNAL_ORG | This table identifies the employees associated with an internal organization and vice versa. Required table for SAS Human Capital Management. |
| EMPLOYEE_X_JOB | This table associates an employee to a position or job. Required table for SAS Human Capital Management. |
| EMPLOYEE_X_JOB_X_COMPETENCY | This table contains the employee's job information with the competencies. It identifies the competencies held by the employee associated with the job position, and the employee's competency value and the bounds that the employee holds. This table will be used by the future SAS Human Capital Management Competency Model. |
| EMPLOYMENT_APPLICATION | This table contains employment application data. Required table when populating SAS Human Capital Management applicant history data. |
| ENROLLMENT_STATUS | This table defines enrollment statuses. Examples: regular, night, continuing education. This table will be used by the future SAS Human Capital Management Education Model. |

| Name | Comment |
|---|---|
| ETHNICITY | This table is used to define an individual's ethnic origin. This table is used to build the SAS Human Capital Management formats for ETHNICITY_CD. |
| EVIDENCE_TYPE | The type classification of competency evidence. Examples: CRT is Certificate, DEG is Degree, CMP is Completion, or OBS is Manager Observation. This table will be used by the future SAS Human Capital Management Competency Model. |
| EXEMPT_STATUS | This table defines a position as "exempt" or "non-exempt." An exempt position is purely management, administrative, or professional. Employees in exempt positions perform without close supervision. Consequently, they are exempt from coverage under the Fair Labor Standards Act. This table is used to build the SAS Human Capital Management formats for EXEMPT_STATUS_CD. Many standard SAS Human Capital Management reports use this information. |
| EXTERNAL_ORG | This table defines organizations that are not part of your enterprise. Some examples are suppliers, vendors, and customers. Compare INTERNAL_ORG. Required table for SAS Human Capital Management if tracking competency taxonomy. |
| EXTERNAL_ORG_ADDRESS | This table has the address associated with an external organization. |
| EXTERNAL_ORG_ASSOC | This table contains the parent/child relationships that make up the EXTERNAL_ORG hierarchies. |
| EXTERNAL_ORG_ASSOC_TYPE | This table identifies the EXTERNAL_ORG hierarchies. |
| EXTERNAL_ORG_CONTACT | This table contains contact information and associations to the EXTERNAL_ORG and EXTERNAL_ORG_ADDRESS tables. |
| EXTERNAL_ORG_NLS | This table supplements the EXTERNAL_ORG table with localized member names and descriptions. |
| FICE | This table contains Federal Interagency Committee on Education (FICE) codes. FICE codes are assigned by the Department of Education and are used as the primary identifier for each academic institution in the Graduate Student Survey. The FICE table represents the Institution FICE code information as identified by the Federal Interagency Committee on Education. FICE code table data is used as the primary identifier for each academic institution. This table will be used by the future SAS Human Capital Management Education Model. |

| Name | Comment |
|---|---|
| FLSA_STATUS | This table defines the Employee FLSA (Fair Labor Standards Act) status. The FLSA status might be the same as the exempt status. This table is used to build the SAS Human Capital Management formats for FLSA_STATUS_CD. |
| GENDER | This table defines the codes for gender. This table is used to build the SAS Human Capital Management formats for GENDER_CD. |
| GRADUATING_DEGREE | This table defines codes that indicate whether a diploma is issued at the completion of the education. Examples: Graduating or Qualifying. This table will be used by the future SAS Human Capital Management Education Model. |
| HONORS_PROGRAM | This table defines honors programs that are part of degree programs. This table will be used by the future SAS Human Capital Management Education Model. |
| INDUSTRY | This table defines industries. |
| INTERNAL_ORG | This table defines organizations that are part of your enterprise. Some examples are departments, divisions, and subsidiaries. Compare EXTERNAL_ORG. Required table for SAS Human Capital Management. |
| INTERNAL_ORG_ASSOC | This table contains the parent/child relationships that make up the INTERNAL_ORG hierarchies. Required table for SAS Human Capital Management. |
| INTERNAL_ORG_ASSOC_TYPE | This table identifies the INTERNAL_ORG hierarchies. Required table for SAS Human Capital Management. |
| INTERNAL_ORG_NLS | This table supplements the INTERNAL_ORG table with localized member names and descriptions. |
| ITEM_CATEGORY | This table contains the item or product hierarchy names for each level in that hierarchy. |
| ITEM_CATEGORY_ASSOC | This table contains the parent/child relationships that make up the ITEM_CATEGORY hierarchies. |
| ITEM_CATEGORY_ASSOC_TYPE | This table identifies the ITEM_CATEGORY hierarchies. |
| JOB | This table contains the various jobs that the company has. Required table when populating SAS Human Capital Management job information data. |
| JOB_GROUP | This table defines the various job groups that a company has. The job group is usually associated with an EEO classification. This table is used to build the SAS Human Capital Management formats for JOB_GROUP_CD and to load the JOBS table. Required table when populating SAS Human Capital Management job information data. |

| Name | Comment |
|---|---|
| JOB_POSITION | This table defines a company's job positions. It can include open and closed positions. Required table when populating SAS Human Capital Management job position information data. |
| JOB_X_COMPETENCY | This table is the intersection of the JOB and COMPETENCY tables. It identifies what competencies are required to perform the duties of the job. It also contains the value within the competency bounds that the job is associated with. This table will be used by the future SAS Human Capital Management Competency Model. |
| MARITAL_STATUS | This table defines the codes used for marital status. This table is used to build the SAS Human Capital Management formats for MARITAL_STATUS_CD. |
| MEASURE | This table contains measures that can be used in scorecards. A default source table is supplied. |
| MILITARY_EXPERIENCE | This table defines an employee's military experience. Required table when populating SAS Human Capital Management military experience data. |
| MILITARY_EXPERIENCE_TYPE | This table defines the types of military experience. This table is used to build the SAS Human Capital Management formats for MILITARY_EXPERIENCE_TYPE_CD. |
| ORG_TYPE | This table contains codes for organization types. Examples are division, department, and subsidiary. |
| OTHER_HONORS | This table defines honors that students can receive that are not in the ACADEMIC_HONORS table. Example: Phi Beta Kappa. This table will be used by the future SAS Human Capital Management Education Model. |
| PAY_LEVEL | This table is used to define the company's pay structure. The pay structure can be referred to pay grades or bands. The pay levels might be associated with a job or position or with the employee. Required table when populating SAS Human Capital Management pay level information. |
| PAY_LEVEL_STRUCTURE | This table is used to define unique codes in the company's pay level structure. Required table when populating SAS Human Capital Management pay level information. |
| POSITION_PERMANENCE | This table indicates whether the position is regular or temporary. This table is used to build the SAS Human Capital Management formats for PERMANENCE_CD. |
| POSITION_STATUS | This table defines unique position status codes. This table is used to build the SAS Human Capital Management formats for POSITION_STATUS_CD. |

| Name | Comment |
|---|---|
| PROFIT_CENTER | This table contains categories that identify the division of functional areas within an organization. These categories are typically nonphysical entities to which revenues and costs are assigned. The table includes child and parent members. |
| PROFIT_CENTER_ASSOC | This table contains the parent/child relationships that make up the PROFIT_CENTER hierarchies. |
| PROFIT_CENTER_ASSOC_TYPE | This table identifies the PROFIT_CENTER hierarchies. |
| PROFIT_CENTER_NLS | This table supplements the PROFIT_CENTER table with localized member names and descriptions. |
| RECRUITMENT_SOURCE | This table defines the recruitment source for the application. For example, Internet or advertisement. This table is used to build the SAS Human Capital Management formats for RECRUITMENT_SOURCE_CD. |
| REJECTION_REASON | This table defines the reason that the application was rejected. For example, withdrawal or not qualified. This table is used to build the SAS Human Capital Management formats for REJECTION_REASON_CD. |
| RETAINED_EARN_ROLL_FWD_METHOD | This table contains the predefined methods for computing the value of a Retained Earnings account that are in the SAS_RETAINED_EARN_ROLL_FWD_METH table. |
| SASOP_DETAIL | This table contains the Operational Planning transactional data. |
| SCHOOL_DEPT | This table contains a listing of the departments, schools, or other organizational units for the educational institution. Examples: department, branch, group, team. This table will be used by the future SAS Human Capital Management Education Model. |
| SCHOOL_DEPT_TYPE | This table defines the school department types. Examples: department, branch, group or team. This table will be used by the future SAS Human Capital Management Education Model. |
| SCHOOL_NAME_TYPE | This table contains codes for different types of educational institutions. This table will be used by the future SAS Human Capital Management Education Model. |
| SCHOOL_OR_INSTITUTION | This table contains a listing of educational institutions tracked by the organization. This can be an internal or external listing depending on the site. This table will be used by the future SAS Human Capital Management Education Model. |

| Name | Comment |
|---|---|
| SCHOOL_TYPE | This table contains codes for different types of educational institutions. Examples: community college, trade school, university. This table will be used by the future SAS Human Capital Management Education Model. |
| SOURCE_SYSTEM | This table contains the predefined source system codes that are in the SAS_SOURCE_SYSTEM table. |
| SPECIAL_REF_COMP | This table contains special reference competency information. This table will be used by the future SAS Human Capital Management Competency Model. |
| STATE_REGION | This table contains unique codes that are associated with states or regions. Examples include "AK" for Alaska, "AL" for Alabama, and "AR" for Arkansas. This table is used to build the SAS Human Capital Management formats for STATE_REGION_CD. |
| TAXONOMY | This table contains the codes that are used to cross-reference the internal competencies to external sources. Example: Corpx Java Programming competency with a code of JAV references to CorpExternal for Java Programming with a code of ABC. This table will be used by the future SAS Human Capital Management Competency Model. |
| TAXONOMY_CATALOG | This table defines taxonomy catalogs. This table will be used by the future SAS Human Capital Management Competency Model. |
| TAXONOMY_COMP_CLASS | This table defines competency classes for taxonomies. This table will be used by the future SAS Human Capital Management Competency Model. |
| TAXONOMY_SOURCE | This table contains the listing of external or internal competency comparisons. For example, the site wants to compare its software development internal competency listings to other organizations. This table identifies the external source and along with the crosswalk table provides the ability for comparison. This table will be used by the future SAS Human Capital Management Competency Model. |
| TAXONOMY_SPECIAL_REF_COMP | This table defines special reference competency codes for the taxonomy. This table will be used by the future SAS Human Capital Management Competency Model. |
| TIME_FREQUENCY | This table contains a list of codes that quantify the number of hours that span a particular time period. For example, a time frequency code of "WK" would equate to hours_per_period_qty of 168 hours. This table is used to build the SAS Human Capital Management formats for TIME_FREQUENCY_CD. |

| Name | Comment |
|---|---|
| TIME_PERIOD | This table contains a list of time periods that are used to represent time as a hierarchy. The table would include both the child and parent members that would be used in traditional hierarchy/dimension. Examples include AllYears, YR2002, 1stQtr2002, and Jan2002. Used for SAS Human Capital Management metrics. |
| TIME_PERIOD_ASSOC | This table contains the parent/child relationships that make up the TIME_PERIOD hierarchies. Used for SAS Human Capital Management metrics. |
| TIME_PERIOD_ASSOC_TYPE | This table identifies the TIME_PERIOD hierarchies. Used for SAS Human Capital Management metrics. |
| TIME_PERIOD_NLS | This table supplements the TIME_PERIOD table with localized member names and descriptions. |
| TIME_UNIT_OF_MEASURE | This table contains a list of unique codes that are used to categorize types of time frequencies into units of measure. |
| WEIGHT_TYPE | This table defines weighting types that are used for measurement. This table will be used by the future SAS Human Capital Management Competency Model. |

# *Chapter 36*
# Descriptions of DDS Table Columns

**Descriptions of Detail Data Store Table Columns** . . . . . . . . . . . . . . . . . . . . . . . . . 401

## Descriptions of Detail Data Store Table Columns

Each table in this chapter contains descriptions of all the columns in a particular DDS table. The tables are in alphabetical order.

Each staging table contains a subset of the columns in the corresponding DDS table. The DDS table columns that are not in the staging tables are the PROCESSED_DTTM columns and the columns that have names that end with _RK:

- The PROCESSED_DTTM columns tell when a table was most recently loaded.
- The _RK columns contain unique integer key values that map to the business key values.

The time stamps and integer key values are generated by the jobs that load the DDS tables from the corresponding staging tables.

***Table 36.1*** *ABSENCE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| ABSENCE_TYPE_CD | VARCHAR(10) | Code for a type of absence. Typical absence types are sickness, vacation, disability, family leave, and bereavement. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the absence type is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the absence type is valid to. |
| ABSENCE_TYPE_DESC | VARCHAR(255) | Description of a type of absence. Typical absence types are sickness, vacation, disability, family leave, and bereavement. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.2*** *ACADEMIC_CREDIT Table*

| Name | Data Type | Comment |
|---|---|---|
| ACADEMIC_CREDIT_CD | VARCHAR(10) | Code indicating the type of credit used or awarded by the institution. Examples: A = adult credits, C = continuing education units, N = no credit, Q = quarter hour credit. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ACADEMIC_CREDIT_DESC | VARCHAR(255) | Description of the academic credit. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.3** ACADEMIC_HONORS Table*

| Name | Data Type | Comment |
|---|---|---|
| ACADEMIC_HONORS_CD | VARCHAR(10) | Code for any academic honors associated with the degree. Example: magna cum laude. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ACADEMIC_HONORS_DESC | VARCHAR(255) | Description of the academic honors. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.4** ACTION_REASON Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_ACTION_REASON_CD | VARCHAR(10) | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the action reason is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the action reason is valid to. |

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_ACTION_REASON_DESC | VARCHAR(255) | Description of a reason for an employee action. For example, "promotion" is a reason for a pay raise. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.5*** *ACTION_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| ACTION_TYPE_CD | VARCHAR(10) | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the action type is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the action type is valid to. |
| ACTION_TYPE_DESC | VARCHAR(255) | Description of a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.6*** *ADDRESS_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| ADDRESS_TYPE_CD | VARCHAR(10) | Code for an address type. Typical address types are business, shipping, mailing, and primary residence. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ADDRESS_TYPE_DESC | VARCHAR(255) | Description of an address type. Typical address types are business, shipping, mailing, and primary residence. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.7*** *APPLICATION_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| APPLICATION_STATUS_CD | VARCHAR(10) | Code for the status of an application for employment. This column is used for data validation in DDS ETL jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the employment application is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the employment application is valid to. |

| Name | Data Type | Comment |
|---|---|---|
| APPLICATION_STATUS_DESC | VARCHAR(255) | Description of the application status that is identified in the APPLICATION_STATUS_CD column. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.8*** *ATTENDANCE_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| ATTENDANCE_STATUS_CD | VARCHAR(10) | A code that identifies the attendance status of the student within the organization unit. Example: A student transfers from the School of Business to the School of Journalism. The first instance shows "School of Business, prior" and the second instance shows "School of Journalism, current." |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ATTENDANCE_STATUS_DESC | VARCHAR(255) | Description of the attendance status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.9*** *CODE_LANGUAGE Table*

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies a language/ locale. An example is "en" for English. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| LANGUAGE_DESC | VARCHAR(255) | Description of the character code that identifies the languages. |
| DEFAULT_LANGUAGE_FLG | CHAR(1) | Flag indicating the default language code used in the DDS. This language/locale is used for the names and descriptions in the tables without LANGUAGE_CD as part of the key. |
| LOCALE_LANGUAGE_CD | VARCHAR(2) | Two-character ISO0639 language code associated with the locale. |
| LOCALE_VARIANT_CD | VARCHAR(32) | Blank by default. This column can be used as the third parameter in the aggregate locale string used by the SDM. (Locale_language_cd + locale_country_cd + locale_variant_cd.) |
| LOCALE_COUNTRY_CD | VARCHAR(2) | This three-character column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.10*** *COMPENSATION Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. This column should be populated with the EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management COMPHIST table. |
| COMPENSATION_TYPE_CD | VARCHAR(10) | Unique identifier for the COMPENSATION TYPE occurrence. |

| Name | Data Type | Comment |
|---|---|---|
| PAYMENT_DT | DATE | Date that the compensation was paid. This column is required for the build of the SAS Human Capital Management COMPHIST table. |
| COMPENSATION_AMT | NUMERIC(10,2) | Amount of compensation paid to the employee. |
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.11*** *COMPENSATION_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPENSATION_TYPE_CD | VARCHAR(10) | Unique identifier for the COMPENSATION TYPE occurrence. This column is used for data validation in DDS ETL jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the compensation type is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the compensation type code is valid to. |
| COMPENSATION_TYPE_DESC | VARCHAR(255) | Description of the type of compensation. |
| BASE_SALARY_FLG | CHAR(1) | Indicates whether the compensation type identifies base salary. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.12 COMPETENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the COMPETENCY table data is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY table data is valid to. |
| COMPETENCY_ID | VARCHAR(32) | Code that identifies a particular competency. COMPETENCY_TYPE_CD and COMPETENCY_ID are the two components of this table's business key. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | Code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. |
| COMPETENCY_NM | VARCHAR(40) | Short name of the competency. |
| COMPETENCY_DESC | VARCHAR(255) | Description of the competency. |
| COMPETENCY_OWNER_NM | VARCHAR(40) | Owner name for the competency. |
| COMPETENCY_LEVEL_VAL | VARCHAR(20) | Level designation for the competency. |
| COMPETENCY_CATEGORY_CLASS_CD | VARCHAR(10) | Code from the COMPETENCY_CATEGORY_CLASS table. The code translates to several competency classification categories. |
| COMPETENCY_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Indicates whether this member's values roll up into its parent’s values. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This timestamp is supplied by the job. |

*Table 36.13 COMPETENCY_ASSOC Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| PARENT_COMPETENCY_RK | NUMERIC(10) | Surrogate key for parent competency record in the hierarchy represented by this association. A competency can include other competencies. One competency can consist of several component competencies, each of which might be separately measurable. A maximum of one parent level is possible in this model. |
| COMPETENCY_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |
| COMPETENCY_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This timestamp is supplied by the job. |

***Table 36.14*** *COMPETENCY_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| LANGUAGE_CD | VARCHAR(3) | Code identifying the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| COMPETENCY_ASSOC_TYPE_DESC | VARCHAR(255) | Description of a hierarchy. |
| COMPETENCY_ADK | VARCHAR(32) | This column is not used. |
| DEFAULT_COMPETENCY_RK | NUMERIC(10) | Since source data for COMPETENCY_ASSOC_TYPE can come from multiple systems, the business-supplied keys might not be unique. A surrogate key is added in the ETL process to ensure a unique identifier for COMPETENCY_ASSOC_TYPE. This key is used with VALID_FROM_DTTM for versioning of rows. |
| PROCESSED_DTTM | DATE | Timestamp for the last time a record was processed, typically by ETL load processing. The record also could be updated when inter-ETL cycle modifications are made to a record. |

*Table 36.15 COMPETENCY_CATALOG Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_CATALOG_CD | VARCHAR(10) | When the COMPETENCY table information is identified by means of a catalog, the COMPETENCY_CATALOG_CD column represents the catalog reference table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COMPETENCY_CATALOG_DESC | VARCHAR(255) | Description of the competency catalog code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.16 COMPETENCY_CATEGORY Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_CATEGORY_CD | VARCHAR(10) | Code that identifies a competency class. |
| LANGUAGE_CD | VARCHAR(3) | Code identifying the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COMPETENCY_CATEGORY_DESC | VARCHAR(255) | Description of the competency class. |

| **Name** | **Data Type** | **Comment** |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This timestamp is supplied by the job. |

***Table 36.17*** *COMPETENCY_CATEGORY_CLASS Table*

| **Name** | **Data Type** | **Comment** |
|---|---|---|
| COMPETENCY_CATEGORY_CLASS_CD | VARCHAR(10) | Code that identifies a Competency Category Class. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the COMPETENCY_TYPE table data is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY table record is valid to. |
| COMPETENCY_CATALOG_CD | VARCHAR(10) | Catalog code for the competency. If the competency refers to an internal or external published catalog, this is where that reference is located. |
| COMPETENCY_CLASS_CD | VARCHAR(10) | Code that identifies a competency class. |
| COMPETENCY_TYPE_CD | VARCHAR(10) | Code that identifies a competency class. |
| COMPETENCY_CATEGORY_CD | VARCHAR(10) | Code that identifies a competency class. |
| COMPETENCY_JOB_GROUP_CD | VARCHAR(10) | Job group code. This is usually a grouping of jobs. It is usually associated with only one EEO category. This column is used for data validation in DDS jobs. This column is required for the build of the SAS Human Capital Management JOBS table. |
| SPECIAL_REF_COMP_CD | VARCHAR(10) | Special reference competency code. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This timestamp is supplied by the job. |

*Table 36.18 COMPETENCY_CLASS Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_CLASS_CD | VARCHAR(10) | Code that identifies a competency class. |
| LANGUAGE_CD | VARCHAR(3) | |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| COMPETENCY_CATEGORY_CD | VARCHAR(10) | A code that identifies a competency class. |
| LANGUAGE_CD__3370975 | VARCHAR(3) | Code identifying the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM__3370978 | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COMPETENCY_CLASS_DESC | VARCHAR(255) | Description of the competency class. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This timestamp is supplied by the job. |

*Table 36.19 COMPETENCY_EVIDENCE Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_RK | NUMERIC(10) | Reference key of the COMPETENCY associated with the COMPETENCY_EVIDENCE. |
| EVIDENCE_TYPE_CD | VARCHAR(10) | A code that identifies a type of evidence. Examples: CRT=certificate, DEG=degree, OBS=observation, CMP=completion. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY is valid to. |
| COMPETENCY_ID | VARCHAR(32) | A COMPETENCY_ID value from the COMPETENCY table. |
| EVIDENCE_NM | VARCHAR(40) | Name of the evidence associated with the COMPETENCY table data. |
| EVIDENCE_DESC | VARCHAR(255) | Description of the evidence needed to demonstrate the competency. For bounds used in the metric ranges high/low, a site can include the values in the description. For example, for college degree, using a string value, the description could include BS, BA, MS, or PHD. |
| EVIDENCE_OWNER_NM | VARCHAR(40) | Owner name of the evidence. |
| EVIDENCE_REQUIRED_FLG | CHAR(1) | Flag to indicate whether the competency evidence is mandatory for a particular position or given context. |

| Name | Data Type | Comment |
|---|---|---|
| EVIDENCE_NUM_MINIMUM_VAL | NUMERIC(15,2) | The minimum numeric value of the rating scale. This column is populated when the range is numeric. For example, if the bounds are 1 to 5 with an interval of 1, then the job or employee's evidence rating is 1, 2, 3, 4, or 5, and the minimum value in this column is 1. |
| EVIDENCE_NUM_MAXIMUM_VAL | NUMERIC(15,2) | The maximum numeric value of the rating scale. This column is populated when the range is numeric. For example, if the bounds are 1 to 5 with an interval of 1, then the job or employee's evidence rating is 1, 2, 3, 4, or 5, and the maximum value in this column is 5. |
| EVIDENCE_NUM_INTERVAL_VAL | NUMERIC(15,2) | The numeric increment or step for the relevant scale. This column is populated when the range is numeric. For example, if the bounds are 1 to 5 with an interval of 1, then the job or employee's evidence rating is 1, 2, 3, 4, or 5, and the interval value in this column is 1. |
| EVIDENCE_STR_MINIMUM_VAL | VARCHAR(20) | The string minimum value of the rating scale. This column is populated when the range is a character string instead of numeric. Example: For a college degree the EVIDENCE_STRING_MINIMUM_VAL value is BS and for EVIDENCE_STRING_MAXIMUM_VAL value is PHD. |
| EVIDENCE_STR_MAXIMUM_VAL | VARCHAR(20) | The string maximum value of the rating scale. This column is populated when the range is a character string instead of numeric. Example: For a college degree the EVIDENCE_STRING_MINIMUM_VAL value is BS and the EVIDENCE_STRING_MAXIMUM_VAL value is PHD. |
| EVIDENCE_STR_INTERVAL_VAL | VARCHAR(20) | The string increment or step based on the scale being used for measurement. This column is populated for a character-string range that has an interval identified. It can be left blank. |

| Name | Data Type | Comment |
|---|---|---|
| LAST_USED_DT | DATE | The date on which the competency evidence was last used. |
| SUPPORTING_INFORMATION_TXT | VARCHAR(100) | Additional descriptive information to substantiate or clarify a rating, measure, value, and so on. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.20*** *COMPETENCY_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_TYPE_CD | VARCHAR(10) | A code that identifies a competency class. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COMPETENCY_TYPE_DESC | VARCHAR(255) | Description of the competency class. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.21*** *COMPETENCY_WEIGHT_BOUNDS Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_RK | NUMERIC(10) | Reference key of the competency associated with the COMPETENCY_WEIGHT_BOUNDS table row data. |

| Name | Data Type | Comment |
|---|---|---|
| WEIGHT_TYPE_CD | VARCHAR(10) | A code for a weight type. The COMPETENCY_WEIGHT_BOUNDS table can have multiple entries based on weight type. For example, there can be multiple rows representing a required bound, level of interest bound, or skill bound. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY_WEIGHT_BOUNDS table data is valid to. |
| COMPETENCY_ID | VARCHAR(32) | The business key of the COMPETENCY table. |
| WEIGHT_DESC | VARCHAR(255) | Description of the COMPETENCY_WEIGHT_BOUNDS table data. For the range between the minimum and maximum, the description must include the values. Example: If the WEIGHT_NUMERIC_MINIMUM_VAL is 0 and the WEIGHT_NUMERIC_MAXIMUM_VAL is 10 and the WEIGHT_NUMERIC INTERVAL_VAL is 5 then the description would include 0=0's description, 5=5's description, and 10=10's description. |
| WEIGHT_NUM_MINIMUM_VAL | NUMERIC(15,2) | Numeric minimum value for the rating scale. Populated when numeric bounds are used. For example, if the minimum value is 0 and the maximum value is 10 and the interval is 5, then the WEIGHT_NUMERIC_MINIMUM_VAL value is 0. |
| WEIGHT_NUM_MAXIMUM_VAL | NUMERIC(15,2) | Numeric maximum value for the rating scale. Populated when numeric bounds are used. For example: If the minimum value is 0 and the maximum value is 10 and the interval is 5, then the WEIGHT_NUMERIC_MAXIMUM_VAL value is 10. |

| Name | Data Type | Comment |
|---|---|---|
| WEIGHT_NUM_INTERVAL_VAL | NUMERIC(15,2) | The numeric increment or step for the relevant scale. Populated when numeric bounds are used. For example, if the minimum value is 0 and the maximum value is 10 and the interval is 5, then the WEIGHT_NUMERIC_INTERVAL_VAL value is 5. |
| WEIGHT_STR_MINIMUM_VAL | VARCHAR(20) | String minimum value for the rating scale. Populated when character bounds are used. For example, if the minimum value is "Not Interested" and the maximum value is "Strongly Interested," then the WEIGHT_STRING_MINIMUM_VAL value is "Not Interested." |
| WEIGHT_STR_MAXIMUM_VAL | VARCHAR(20) | String maximum value for the rating scale. Populated when character bounds are used. For example, if the minimum value is "Not Interested" and the maximum value is "Strongly Interested," then the WEIGHT_STRING_MAXIMUM_VAL value is "Strongly Interested." |
| SUPPORTING_INFORMATION_TXT | VARCHAR(100) | Additional descriptive information to substantiate or clarify a rating, measure, value, and so on. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.22*** *COMPETENCY_X_TAXONOMY_SOURCE Table*

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_RK | NUMERIC(10) | Reference key used to establish the relationship of the Competency to the Taxonomy Source. |
| TAXONOMY_SOURCE_RK | NUMERIC(10) | Reference key used to establish the relationship of the Taxonomy Source to the Competency. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY is valid to. |
| COMPETENCY_ID | VARCHAR(32) | The business key of the COMPETENCY table. |
| TAXONOMY_SOURCE_ID | VARCHAR(32) | Business key for taxonomy source. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.23*** *CONTACT_ROLE Table*

| Name | Data Type | Comment |
|---|---|---|
| CONTACT_ROLE_CD | VARCHAR(3) | Code for a contact role. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| CONTACT_ROLE_DESC | VARCHAR(255) | Description of the contact role that is identified in CONTACT_ROLE_ID. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.24*** *COST_CENTER Table*

| Name | Data Type | Comment |
|---|---|---|
| COST_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COST_CENTER_ID | VARCHAR(32) | Unique ID for a functional area within an organization, to which costs are assigned. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| COST_CENTER_NM | VARCHAR(50) | Short name for describing categories that identify the division of functional areas within an organization. These categories are typically non-physical entities to which costs are assigned. |
| COST_CENTER_DESC | VARCHAR(255) | Long name for describing categories that identify the division of functional areas within an organization. These categories are typically non-physical entities to which costs are assigned. |
| RESPONSIBLE_EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. |
| COST_CENTER_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Identifies if this member's values rolls up into its parent. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.25 COST_CENTER_ASSOC Table*

| Name | Data Type | Comment |
|---|---|---|
| COST_CENTER_RK | NUMERIC(10) | The COST_CENTER_RK value from the COST_CENTER table for the child in a parent-child relationship. |

| Name | Data Type | Comment |
|---|---|---|
| PARENT_COST_CENTER_RK | NUMERIC(10) | The COST_CENTER_RK value from the COST_CENTER table for the parent in a parent-child relationship. |
| COST_CENTER_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |
| COST_CENTER_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.26*** *COST_CENTER_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| COST_CENTER_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COST_CENTER_ASSOC_TYPE_DESC | VARCHAR(255) | Description of a hierarchy. |
| COST_CENTER_ADK | VARCHAR(32) | This column is not used. |

| Name | Data Type | Comment |
|---|---|---|
| DEFAULT_COST_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.27*** *COST_CENTER_NLS Table*

| Name | Data Type | Comment |
|---|---|---|
| COST_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COST_CENTER_NM | VARCHAR(50) | Short name for describing functional areas within an organization, to which costs are assigned. This column represents the specific language/locale identified. |
| COST_CENTER_DESC | VARCHAR(255) | Long name for describing functional areas within an organization, to which costs are assigned. This column represents the language/locale description. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.28 COUNTRY Table*

| Name | Data Type | Comment |
|---|---|---|
| COUNTRY_CD | VARCHAR(3) | This column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. This column is used for data validation in DDS ETL jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the country code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the country code is valid to. |
| COUNTRY_DESC | VARCHAR(255) | Country name/description for the unique two-character code, as defined by ISO 3166 standard. |
| COUNTRY_REGION_DESC | VARCHAR(255) | Regional location of the country. Examples are North America, Eastern Europe, Asia. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.29 COUNTY Table*

| Name | Data Type | Comment |
|---|---|---|
| STATE_REGION_CD | VARCHAR(10) | Unique code for a state or region. Examples include "AK" for Alaska, "AL" for Alabama, "AR" for Arkansas. |
| COUNTY_NM | VARCHAR(50) | County name. This column is used for data validation in DDS ETL jobs. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the county name is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the county name is valid to. |
| COUNTY_DESC | VARCHAR(255) | Long description of the county. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.30*** *COURSE_LEVEL Table*

| Name | Data Type | Comment |
|---|---|---|
| COURSE_LEVEL_CD | VARCHAR(10) | Code established by the National Center for Education Statistics to indicate the level and/or the type of work, which is reflected in the grade average and the credit hours. Examples: 1 = Remedial; 2 = Basic; 3 = Teacher's Aide; 4 = General; A = Summary of all courses taken at all institutions; AR = Academic Renewal. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| COURSE_LEVEL_DESC | VARCHAR(255) | Description of the course level that is identified by COURSE_LEVEL_CD. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.31*** *CURRENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. This column is used for data validation in DDS ETL jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the currency code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the currency code is valid to. |
| CONVERTED_TO_EURO_FLG | CHAR(1) | Indicates that the currency was converted to Euro. |
| CURRENCY_DESC | VARCHAR(255) | Currency name/description for the unique three-character code, as defined by ISO 4217 standard. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.32*** *DATES_OF_ATTENDANCE Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_HISTORY_RK | NUMERIC(10) | Reference key of the education history associated with the dates of attendance. |

| Name | Data Type | Comment |
|---|---|---|
| ATTENDANCE_START_DT | DATE | Beginning date of attendance at an educational institution. |
| ENROLLMENT_STATUS_CD | VARCHAR(10) | Code for enrollment status. Examples: regular, night, continuing education. |
| EDUCATION_HISTORY_ID | VARCHAR(32) | Business key for the EDUCATION_HISTORY table data record. |
| CURRENTLY_ENROLLED_FLG | CHAR(1) | Indicates whether the person is currently enrolled as a student. |
| STUDENT_IN_GOOD_STANDING_FLG | CHAR(1) | Indicates whether the person in question is a student in good standing. This is frequently a value returned in response to checks to verify a dependent student's eligibility for insurance under a parent's insurance policy. |
| ANTICIPATED_GRADUATION_DT | DATE | Date on which the student is scheduled to graduate. |
| ATTENDANCE_END_DT | DATE | Ending date of attendance at an educational institution. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.33*** *DDS_SOLUTIONS_VERSION Table*

| Name | Data Type | Comment |
|---|---|---|
| VERSION_NO | VARCHAR(32) | Version of the specified type of data. |
| SOLUTION_NM | VARCHAR(50) | Type of data that was attempted to be loaded. Valid values are "StageDDS", "DDS", "Common", "ODCS", "FM", and "HCM". |
| VERSION_STATUS_DESC | VARCHAR(255) | Status of the installation or migration for the specified type of data. Valid values are "Successful" or "Failed". |
| LOAD_METHOD_DESC | VARCHAR(255) | Method by which the specified type of data reached its current state. Valid values are "Installation" or "Migration". |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The timestamp for the last time a record was processed, typically by ETL load processing. The timestamp could also be updated when inter-ETL cycle modifications are made to a record. |

***Table 36.34*** *DEGREE_CONCENTRATION Table*

| Name | Data Type | Comment |
|---|---|---|
| DEGREE_CONCENTRATION_CD | VARCHAR(10) | Code for the emphasis associated with a major or field of study. Example: A masters degree in mathematics with a concentration in Bioinformatics. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| DEGREE_CONCENTRATION_DESC | VARCHAR(255) | Description of the degree concentration. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.35*** *DEGREE_OPTION Table*

| Name | Data Type | Comment |
|---|---|---|
| DEGREE_OPTION_CD | VARCHAR(10) | Code for the official name of a major option associated with the student's degree title. Examples: A particular institution might require their Civil Engineering majors to select either a mechanical engineering option or a structural engineering option. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| DEGREE_OPTION_DESC | VARCHAR(255) | Description of the degree option. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.36*** *DEGREE_PROGRAM Table*

| Name | Data Type | Comment |
|---|---|---|
| DEGREE_PROGRAM_CD | VARCHAR(10) | Code reflecting the program associated with the course of study. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| DEGREE_PROGRAM_DESC | VARCHAR(255) | Description of the degree program. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.37 DEGREE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| DEGREE_TYPE_CD | VARCHAR(10) | Code for the type of degree. Examples: doctorate, masters, bachelors, associate certificate. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| DEGREE_TYPE_DESC | VARCHAR(255) | Description of the degree type. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.38 EDUCATION_HISTORY Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_HISTORY_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the COMPETENCY table data is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the COMPETENCY is valid to. |
| EDUCATION_HISTORY_ID | VARCHAR(32) | Business key for the EDUCATION_HISTORY table data record. |
| EMPLOYEE_RK | NUMERIC(10) | Reference key value for the employee. |
| EMPLOYEE_ID | VARCHAR(32) | Business key identifying the employee. |

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_DEPT_RK | NUMERIC(10) | Reference key of the SCHOOL_DEPARTMENT table associated with the EDUCATION_HISTORY employee data. |
| EDUCATION_ACHIEVEMENT_NM | VARCHAR(40) | Name of the educational achievement attained. |
| TERMINAL_DEGREE_FLG | CHAR(1) | A flag to indicate the highest academic degree in a field of study. If a doctorate is the highest degree that a field offers, then it is the terminal degree. If a bachelors or masters is the highest degree available in the field, then that is the terminal. |
| ACADEMIC_HONORS_CD | VARCHAR(10) | Code for any academic honors associated with the degree. Example: magna cum laude. |
| HONORS_PROGRAM_CD | VARCHAR(10) | Code for any honors program associated with the degree. |
| OTHER_HONORS_CD | VARCHAR(10) | Code for any other honors that the student received. Example: Phi Beta Kappa. |
| GRADUATION_DT | DATE | Date the student graduated from the academic program. |
| DEGREE_TYPE_CD | VARCHAR(10) | Code for the type of degree. Examples: doctorate, masters, bachelors, associate certificate. |
| DEGREE_DESCRIPTION | VARCHAR(255) | Description of the degree that was attained. |
| DEGREE_CONCENTRATION_CD | VARCHAR(10) | Code for the emphasis associated with a major or field of study. Example: A masters degree in business administration with a concentration in personnel. |
| DEGREE_PROGRAM_CD | VARCHAR(10) | Code reflecting the program associated with the course of study. |
| DEGREE_OPTION_CD | VARCHAR(10) | Code for the official name of a major option associated with the student's degree title. Example: A particular institution might require their civil engineering majors to select either a mechanical engineering option or a structural engineering option. |

| Name | Data Type | Comment |
|---|---|---|
| GRADUATING_DEGREE_CD | VARCHAR(10) | Code that indicates whether a diploma or certification is issued at the completion of the education. Examples: Graduating, Qualifying, Certification. |
| EXAM_REQUIRED_FLG | CHAR(1) | Flag indicating if an exam is required to complete the education. |
| EXAM_PASSED_FLG | CHAR(1) | Flag indicates the individual passed an exam related to or required for the education. |
| COMMENTS_TXT | VARCHAR(100) | Comments relating to education. |
| ATTENDANCE_STATUS_CD | VARCHAR(10) | Code for the status of the student within the SCHOOL_DEPT table. Examples: A student transfers from the School of Business to the School of Journalism. The first record shows School of Business with an attendance status code of TRNSFR and the second record shows the School of Journalism with an attendance status code of ATTENDING. |
| TUITION_ASSISTANCE_FLG | CHAR(1) | Flag indicating if the student received tuition assistance to attain the education. |
| MAJOR1_NM | VARCHAR(40) | Name of the first major degree. |
| MAJOR2_NM | VARCHAR(40) | Name of the second major degree. |
| MINOR1_NM | VARCHAR(40) | Name of the first minor degree. |
| MINOR2_NM | VARCHAR(40) | Name of the second minor degree. |
| EDUCATION_VALUE_SYSTEM_CD | VARCHAR(10) | An EDUCATION_VALUE_SYSTEM_CD value from the EDUCATION_VALUE_SYSTEM table. |
| EXCESSIVE_VALUE_FLG | CHAR(1) | Flag that indicates whether an excessive value is possible on the relevant scale or rating system. Examples: A "False" value indicates it is not possible to have a higher GPA than indicated in range maximum. A "True" value indicates it is possible to have a higher GPA than indicated. |

| Name | Data Type | Comment |
|---|---|---|
| GOOD_STUDENT_FLG | CHAR(1) | Flag that indicates whether the person is a good student under the criteria established by the school. Good students might be eligible for insurance discounts. |
| CUMULATIVE_SUMMARY_FLG | CHAR(1) | Flag that indicates whether this is a summary of all work included in the record. |
| ACADEMIC_CREDIT_CD | VARCHAR(10) | Code that indicates the type of credit awarded by the institution. Examples: A = adult credits, C = continuing education units, N = no credit, Q = quarter-hour credit. |
| COURSE_LEVEL_CD | VARCHAR(10) | A COURSE_LEVEL_CD value from the COURSE_LEVEL table. |
| ACADEMIC_CREDIT_HRS_INCLD_NO | NUMERIC(15) | Total number of credits or credit hours included in the grade point average for this particular summary. Inclusion or exclusion of certain credits depends on the policy of the sending institution. |
| ACADEMIC_CREDIT_HRS_ATTPD_NO | NUMERIC(15) | Total number of credits or credit hours attempted and earned and included in this summary. Examples: Credit hours to be included, if available, are credits for which non-punitive grades such as "I" or "W" or "Audit" were awarded. |
| ACADEMIC_CREDIT_HRS_ERND_NO | NUMERIC(15) | Total number of credits or credit hours included on the record for this particular summary. This is normally all credits for which the student paid, whether the credits were used to calculate the grade point average. |
| CLASS_RANK_NO | NUMERIC(15) | A student's numerical class rank with the highest student in the class having a rank or position of 1. |
| GPA_NO | NUMERIC(10,3) | A student's numerical Grade Point Average. |
| TOTAL_STUDENTS_NO | NUMERIC(15) | Total number of students in the class to help position the student's rank. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.39 EDUCATION_LEVEL Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_LEVEL_CD | VARCHAR(10) | Unique identifier for education level. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the education level code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the education level code is valid to. |
| EDUCATION_LEVEL_DESC | VARCHAR(255) | Description of the education-level code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.40 EDUCATION_VALUE Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_HISTORY_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| EDUCATION_VALUE_TYPE_CD | VARCHAR(10) | Code for type of education value. Examples: GPA, rank in class. Used as part of the measurement of the academic performance of the person who has the education. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the EDUCATION_VALUE table data is valid to. |
| EDUCATION_HISTORY_ID | VARCHAR(32) | Business key for the EDUCATION_HISTORY table data record. |

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_NUM_MIN_VAL | NUMERIC(15,2) | When the rating scale is numeric, the EDUCATION_NUMERIC_MINIMUM_VAL column contains the lowest possible numeric value on the scale or rating system. Used in conjunction with the EDUCATION_NUMERIC_MAXIMUM_VAL column to define the measurement bounds. |
| EDUCATION_NUM_MAX_VAL | NUMERIC(15,2) | When the rating scale is numeric, the EDUCATION_NUMERIC_MAXIMUM_VAL column contains the highest possible numeric value on the scale or rating system. Used in conjunction with the EDUCATION_NUMERIC_MINIMUM_VAL column to define the measurement bounds. |
| EDUCATION_NUM_VAL | NUMERIC(15,3) | Numeric value of the educational performance of the person, within the bounds set in the EDUCATION_NUMERIC_MAXIMUM_VAL column and EDUCATION_NUMERIC_MINIMUM_VAL column. Examples: If Education Value Type is Grade Point Average, GPA, then this column might contain a value of 3.5. If Education Value Type is Rank in Class, then this column might contain a value of 130, ranked 130 out of 5000. |
| EDUCATION_STR_MIN_VAL | VARCHAR(20) | When the rating scale is character, the EDUCATION_STRING_MINIMUM_VAL column is used in conjunction with the EDUCATION_STRING_MAXIMUM_VAL column to identify the low and high measurement bounds. The EDUCATION_STRING_MINIMUM_VAL column contains the lowest possible string value on the scale or rating system. Example: If Education Value Type is Pass/Fail, then this column contains FAIL. |
| EDUCATION_STR_MAX_VAL | VARCHAR(20) | When the rating scale is character, the EDUCATION_STRING_MINIMUM_VAL column is used in conjunction with the EDUCATION_STRING_MAXIMUM_VAL column to identify the low and high measurement bounds. The EDUCATION_STRING_MAXIMUM_VAL column represents the highest possible string value on the scale or rating system. Example: If Education Value Type is Pass/Fail and the employee passed, then this column contains PASS. |

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_STR_VAL | VARCHAR(20) | When the rating scale is character, the EDUCATION_STRING_VAL column represents the value of the educational performance of the person. Example: If Education Value Type is Pass/Fail and the employee passed, this column contains PASS. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.41*** *EDUCATION_VALUE_SYSTEM Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_VALUE_SYSTEM_CD | VARCHAR(10) | Code for the scale against which the educational measure can be compared. Business Rules: For a GPA, this might contain the highest GPA that can be attained at this institution. For a class rank, this might contain the size of the class. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| EDUCATION_VALUE_SYSTEM_DESC | VARCHAR(255) | Description of the education value system. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.42 EDUCATION_VALUE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| EDUCATION_VALUE_TYPE_CD | VARCHAR(10) | Code for type of education value. Examples: GPA, rank in class. Used as part of the measurement of the academic performance of the person who has the education. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| EDUCATION_VALUE_TYPE_DESC | VARCHAR(255) | Description of the education value type. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.43 EEO_CLASS Table*

| Name | Data Type | Comment |
|---|---|---|
| EEO_CLASS_CD | VARCHAR(10) | The standard EEO classification codes. EEO1 classifications are used for corporations and EEO4 are used for government reporting at the federal, state, and local level. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the EEO class code is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the EEO class code is valid to. |
| EEO_CLASS_DESC | VARCHAR(255) | Description of the EEO classification code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.44*** *EMPLOYEE Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. This column should be populated with EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. This column is used by a look-up in the DDS jobs. This column is required for the build of the SAS Human Capital Management EMPGEN table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the employee data is valid from. This column is required for the build of the SAS Human Capital Management EMPGEN table. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_ID | VARCHAR(32) | Unique identifier for an employee occurrence. This column is required by SAS Human Capital Management. This value can represent the employee's employee number as long as it uniquely identifies only one employee for all employees brought into the DDS and SAS Human Capital Management. If employee information is brought in from multiple source systems and there are, for example, 2 employees with the same employee number 1234, one from source A and the other from source B, then the EMPLOYEE_ID must be made to be unique. This could simply be A1234 and B1234. The EMPLOYEE_NO column would contain the actual employee number 1234 for both employees as it does not have to be unique. This column is used by a look-up in the DDS jobs. This column is required for the build of the SAS Human Capital Management EMPGEN table. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| FIRST_NM | VARCHAR(50) | Employee's first name. |
| MIDDLE_NM | VARCHAR(50) | Employee's middle name. |
| LAST_NM | VARCHAR(50) | Employee's last name. |
| MIDDLE_INITIAL | CHAR(1) | Employee's middle initial. |
| ADDRESS_LINE_1_TXT | VARCHAR(100) | First line of the address. |
| ADDRESS_LINE_2_TXT | VARCHAR(100) | Second line of the address. |
| ADDRESS_LINE_3_TXT | VARCHAR(100) | Third line of the address. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| ADDRESS_LINE_4_TXT | VARCHAR(100) | Fourth line of the address. Requires job modifications when this column needs to be included in SAS Human Capital Management. |

| Name | Data Type | Comment |
|---|---|---|
| CITY_NM | VARCHAR(50) | City name for the address. This column must be populated for full functionality of the SAS Human Capital Management GEO Analysis. |
| STATE_REGION_CD | VARCHAR(10) | Unique code associated with states or regions. Examples include: "AK" for Alaska, "AL" for Alabama, and "AR" for Arkansas. |
| COUNTRY_CD | VARCHAR(3) | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| CITIZENSHIP_COUNTRY_CD | VARCHAR(3) | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| DISCIPLINARY_ACTION_FLG | CHAR(1) | Flag indicating if an employee is on disciplinary action. |
| EDUCATION_LEVEL_CD | VARCHAR(10) | Unique identifier for an education level. This is the surrogate key to join to the EDUCATION_LEVEL table. |
| ETHNICITY_CD | VARCHAR(10) | Code used to define the ethnicity. This is the surrogate key to join to the Ethnicity table. |
| MARITAL_STATUS_CD | VARCHAR(3) | Code indicating marital status. This is the surrogate key to join to the MARITAL_STATUS table. |
| GENDER_CD | VARCHAR(3) | Code used to specify the gender. This is the surrogate key to join to the Gender table. |
| PRIMARY_LANGUAGE_CD | VARCHAR(3) | Code identifying the language/locale for names and descriptions within this record. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| PERMANENCE_CD | VARCHAR(10) | A PERMANENCE_CODE value from the POSITION_PERMANENCE table. This indicates whether a person is regular, permanent, temporary, and so on. |

| Name | Data Type | Comment |
|---|---|---|
| UNION_CD | VARCHAR(10) | Indicates whether the employee belongs to a union. |
| HOME_TELEPHONE_NO | VARCHAR(20) | Employee's home telephone number. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| DAYTIME_TELEPHONE_NO | VARCHAR(20) | Employee's daytime telephone number. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| MINORITY_FLG | CHAR(1) | Indicates whether the employee is in a minority classification. |
| SOCIAL_SECURITY_NO | VARCHAR(20) | Employee's Social Security number. Specific to the United States. |
| COUNTY_NM | VARCHAR(50) | County name. This column must be populated for full functionality of the SAS Human Capital Management Geographical Analysis. |
| POSTAL_CD | VARCHAR(20) | Postal code for the address (ZIP code in the United States). |
| HIRE_DT | DATE | Employee's hire date. |
| SERVICE_START_DT | DATE | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| MANAGER_EFFECTIVE_DT | DATE | Date the employee becomes a manager. |
| MOBILE_TELEPHONE_NO | VARCHAR(20) | Employee's mobile telephone number. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| FTE_RT | NUMERIC(9,4) | Employee's full-time equivalence value bounded by 0 and 1, where halftime = .5 and fulltime = 1. |
| EMPLOYEE_NO | VARCHAR(20) | Employee's employee number representing the source system. The value in this column does not have to be unique and is not required. |
| BIRTH_DT | DATE | Employee's birth date. |
| DISABILITY_FLG | CHAR(1) | Flag indicating if an employee has a disability. |

| Name | Data Type | Comment |
|---|---|---|
| EMAIL_ADDRESS_TXT | VARCHAR(100) | E-mail address for the employee. The site can determine whether this is to be business or personal. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| USER_NM | VARCHAR(60) | Employee's User Name. This contains the same value as the employee's User Name populated in The User Manager Plug-in in SAS Management Console. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.45*** *EMPLOYEE_ABSENCE Table*

| Name | Data Type | Comment |
|---|---|---|
| ABSENCE_ID | VARCHAR(32) | Since source data for EMPLOYEE_ABSENCE can come from multiple systems, the business-supplied keys might not be unique. A surrogate key is added by the DDS job to ensure a unique identifier for EMPLOYEE_ABSENCE. Used with VALID_FROM_DTTM for versioning of rows. This column is required for the build of the SAS Human Capital Management ABSHIST table. |
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table for an employee who had an absence. The stage table should be populated with the EMPLOYEE_ID. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management ABSHIST table. |
| ABSENCE_TYPE_CD | VARCHAR(10) | An ABSENCE_TYPE_CD value from the ABSENCE_TYPE table. |
| ABSENCE_START_DT | DATE | Effective or beginning date of the employee absence. This column is required for the build of the SAS Human Capital Management ABSHIST table. |
| ABSENCE_END_DT | DATE | Ending date of the employee absence. |

| Name | Data Type | Comment |
|---|---|---|
| DURATION_QTY | NUMERIC(8,2) | Duration or length of the employee absence. |
| DURATION_TIME_UOM_CD | VARCHAR(3) | Unique code for a type of unit of measure. This must be a valid value in the TIME_UOM table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.46*** *EMPLOYEE_ACTION Table*

| Name | Data Type | Comment |
|---|---|---|
| ACTION_DT | DATE | The date of the employee action. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table for an employee who was the object of an action. This column should be populated with the EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| ACTION_TYPE_CD | VARCHAR(10) | An ACTION_TYPE_CD value from the ACTION_TYPE table. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| EMPLOYEE_TYPE_CD | VARCHAR(10) | An EMPLOYEE_TYPE_CD from the EMPLOYEE_TYPE table. |
| EMPLOYEE_ACTION_REASON_CD | VARCHAR(10) | Code that defines the reason for the employee action. For example, promotion is a reason for a pay raise. This code should be valid in the ACTION_REASON table. |

| Name | Data Type | Comment |
|---|---|---|
| PAY_RATE_AMT | NUMERIC(10,2) | Pay rate amount for an employee. The PAY_FREQUENCY_CD column must be used with this column to identify what the PAY_RATE_AMT is. For example, if PAY_FREQUENCY_CD is "Hr" then the PAY_RATE_AMT might be an hourly rate. |
| PAY_FREQUENCY_CD | VARCHAR(3) | Unique code for a time frequency or time span. Defines the type of time frequency. For example: week, bi-weekly, month, year. |
| EXEMPT_STATUS_CD | VARCHAR(10) | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." |
| HOURS_PER_WEEK_CNT | NUMERIC(2) | Number of hours per week that an employee works. |
| EMPLOYEE_PAID_FREQUENCY_CD | VARCHAR(3) | Unique code for a time frequency or time span. Defines the type of time frequency. For example: week, bi-weekly, month, year. |
| PAY_LEVEL_CD | VARCHAR(20) | The employee's actual pay level. |
| PAY_LEVEL_STRUCTURE_CD | VARCHAR(20) | Code used to join to the PAY_LEVEL table. It is a unique identifier of a PAY_LEVEL since the PAY_LEVEL column itself cannot be guaranteed to be unique. If the pay levels are unique, this column can be loaded with just the PAY_LEVEL. |
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR=euro. |
| EMPLOYEE_STATUS_CD | VARCHAR(10) | Code that defines the employment status of an employee. |
| EVALUATION_RESULT_CD | VARCHAR(10) | An employee's evaluation result code. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.47** EMPLOYEE_INVOLVEMENT_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_INVOLVEMENT_TYPE_CD | VARCHAR(3) | This table is not used. |
| LANGUAGE_CD | VARCHAR(3) | This table is not used. |
| VALID_FROM_DTTM | DATE | This table is not used. |
| VALID_TO_DTTM | DATE | This table is not used. |
| EMPLOYEE_INVOLVEMENT_TYPE_DESC | VARCHAR(255) | This table is not used. |
| PROCESSED_DTTM | DATE | This table is not used. |

***Table 36.48** EMPLOYEE_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_STATUS_CD | VARCHAR(10) | Code that defines the employment status of an employee. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the employee status code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the employee status code is valid to. |

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_STATUS_DESC | VARCHAR(255) | Description of the employee status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.49*** *EMPLOYEE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_TYPE_CD | VARCHAR(10) | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the employee type code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the employee type code is valid to. |
| EMPLOYEE_TYPE_DESC | VARCHAR(255) | The description of the EMPLOYEE_TYPE_ID. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.50*** *EMPLOYEE_UNION Table*

| Name | Data Type | Comment |
|---|---|---|
| UNION_CD | VARCHAR(10) | Unique code indicating labor union. This column is used for data validation in DDS jobs. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the union code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the union code is valid to. |
| UNION_DESC | VARCHAR(255) | Description of the LABOR_UNION_CD. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.51*** *EMPLOYEE_X_COMPETENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | Reference key used to establish the relationship of the EMPLOYEE table to the COMPETENCY table. |
| COMPETENCY_RK | NUMERIC(10) | Reference key used to establish the relationship of the COMPETENCY table data row to the EMPLOYEE table information. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the EMPLOYEE_X_COMPETENCY table data is valid to. |
| EMPLOYEE_ID | VARCHAR(32) | Business key for employee in the EMPLOYEE table. |

| Name | Data Type | Comment |
|---|---|---|
| COMPETENCY_ID | VARCHAR(32) | The COMPETENCY_ID contains the business key associated with the COMPETENCY table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.52*** *EMPLOYEE_X_EVIDENCE Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | Reference key used to establish the relationship of the employee in the EMPLOYEE table to the evidence in the COMPETENCY_EVIDENCE table. |
| COMPETENCY_RK | NUMERIC(10) | Reference key used to establish the relationship of the employee held evidence to the competency data in the COMPETENCY table. |
| EVIDENCE_TYPE_CD | VARCHAR(10) | Code to identify the type of evidence associated to the employee holding the evidence. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the EMPLOYEE_X_EVIDENCE table data is valid to. |
| EMPLOYEE_ID | VARCHAR(32) | Business key for employee. |
| COMPETENCY_ID | VARCHAR(32) | A competency code that is defined in the COMPETENCY table. |
| INCIDENT_DT | DATE | Date on which the employee's competency evidence occurred. Example: Employee's license or certification date is 01JAN2007. |

| Name | Data Type | Comment |
|---|---|---|
| EXPIRATION_DT | DATE | Date of any applicable expiration date, such as the date that a license or certification expires. Example: Employee's license or certification expires 31DEC2009. |
| EMP_EVD_NUM_VAL | NUMERIC(10,2) | Numeric measurement factor based on the individual's rating. Used in conjunction with the numeric bounds set for the competency. |
| EMP_EVD_NUM_VALUE_DESC | VARCHAR(255) | Description of the numeric value which stores the measurement factor based on the individual's rating. |
| EMP_EVD_STR_VAL | VARCHAR(100) | String measurement factor based on the individual's rating. Used in conjunction with the character string bounds set for the competency. |
| EMP_EVD_STR_VALUE_DESC | VARCHAR(255) | Description of the string value which stores the measurement factor based on the individual's rating. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.53*** *EMPLOYEE_X_INTERNAL_ORG Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_RK | NUMERIC(10) | An INTERNAL_ORG_RK value from the INTERNAL_ORG table. This column should be populated with the INTERNAL_ORG_ID value in the stage table. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. This column should be populated with the EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management ACTHIST table. |

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the beginning date and time that the employee was associated with the INTERNAL_ORG_ID. This column is required for the build of the SAS Human Capital Management ACTHIST. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the end date and time that the employee was associated with the INTERNAL_ORG_ID. |
| EMPLOYEE_INVOLVEMENT_TYPE_CD | VARCHAR(3) | This column is not used. |
| EMPLOYEE_X_INTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.54*** *EMPLOYEE_X_JOB Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. This column should be populated with EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| JOB_CD | VARCHAR(10) | A code that represents a job within the company. This column is required for the build of the SAS Human Capital Management ACTHIST table. |

| Name | Data Type | Comment |
|---|---|---|
| START_DT | DATE | The date the employee started the job. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| POSITION_CD | VARCHAR(10) | A code that represents a position within the company. This column is required for the build of the SAS Human Capital Management ACTHIST table. |
| END_DT | DATE | The date the employee ended the job. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.55*** *EMPLOYEE_X_JOB_X_COMPETENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10,2) | Reference key used to establish the relationship of the employee held job in the EMPLOYEE_X_JOB table and competency in the COMPETENCY table. |
| JOB_CD | VARCHAR(10) | A code that represents a job within the organization as identified in the EMPLOYEE_X_JOB table. |
| START_DT | DATE | The date the employee started the job based on the data from the EMPLOYEE_X_JOB table. |
| COMPETENCY_RK | NUMERIC(10) | Reference key of the competency associated with the COMPETENCY_WEIGHT_BOUNDS table row data. |
| WEIGHT_TYPE_CD | VARCHAR(10) | A code for a weight type. The COMPETENCY_WEIGHT_BOUNDS table can have multiple entries based on weight type. For example, there can be multiple rows representing a required bound, level of interest bound, or skill bound. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the EMPLOYEE_X_JOB_X_COMPETENCY table data is valid to. |
| EMPLOYEE_ID | VARCHAR(32) | Business key for employee identified in the EMPLOYEE_X_JOB table. |
| COMPETENCY_ID | VARCHAR(32) | The COMPETENCY_ID contains the business key associated with the COMPETENCY table. |
| COMPETENCY_EMP_NUM_VAL | NUMERIC(10,2) | When used with the competency measurement scale WEIGHT_NUMERIC_MINIMUM_VAL column and WEIGHT_NUMERIC_MAXIMUM_VAL column from the COMPETENCY_WEIGHT_BOUNDS table, this represents the employee's measurement value within the job competency. |
| COMPETENCY_EMP_NUM_VAL_DESC | VARCHAR(255) | Description of the COMPETENCY_EMP_NUM_VAL column. |
| COMPETENCY_EMP_STR_VAL | VARCHAR(20) | When used with the competency measurement scale WEIGHT_STRING_MINIMUM_VAL column and WEIGHT_STRING_MAXIMUM_VAL column from the COMPETENCY_WEIGHT_BOUNDS table, this represents the employee's measurement value within the job competency. |
| COMPETENCY_EMP_STR_VAL_DESC | VARCHAR(255) | Description of the COMPETENCY_EMP_STRING_VAL column. |
| POSITION_CD | VARCHAR(10) | The code representing the employees position number within the organization as identified in the EMPLOYEE_X_JOB table. |
| END_DT | DATE | The date the employee ended the job as indicated in the EMPLOYEE_X_JOB table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.56*** *EMPLOYMENT_APPLICATION Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYMENT_APPLICATION_ID | VARCHAR(32) | A unique ID for the employment application. This column is required for the build of the SAS Human Capital Management APPHIST table. |
| POSITION_CD | VARCHAR(10) | Position for which the applicant is applying. This column is required for the build of the SAS Human Capital Management APPHIST table. |
| APPLICATION_DT | DATE | Date of the application. This column is required for the build of the SAS Human Capital Management APPHIST table. |
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. This column should be populated with EMPLOYEE_ID in the stage table. The DDS job looks up the RK value. |
| APPLICATION_STATUS_CD | VARCHAR(10) | The status of the employment application. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| APPLICANT_INTERVIEWED_FLG | CHAR(1) | Indicates whether the applicant was interviewed or not. |
| RECRUITMENT_SOURCE_CD | VARCHAR(10) | Code for the recruitment source that the applicant used. |
| FIRST_NM | VARCHAR(50) | First name of the applicant. |
| MIDDLE_NM | VARCHAR(50) | Middle name of the applicant. |
| LAST_NM | VARCHAR(50) | Last name of the applicant. |
| MIDDLE_INITIAL | CHAR(1) | Middle initial of the applicant. |
| ADDRESS_LINE_1_TXT | VARCHAR(100) | Home address line 1 for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| ADDRESS_LINE_2_TXT | VARCHAR(100) | Home address line 2 for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |

| Name | Data Type | Comment |
|---|---|---|
| ADDRESS_LINE_3_TXT | VARCHAR(100) | Home address line 3 for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| ADDRESS_LINE_4_TXT | VARCHAR(100) | Home address line 4 for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| STATE_REGION_CD | VARCHAR(10) | Unique code associated with states or regions. Examples include: "AK" for Alaska, "AL" for Alabama, and "AR" for Arkansas. |
| CITY_NM | VARCHAR(50) | City name in address of the applicant. |
| POSTAL_CD | VARCHAR(20) | Postal code (ZIP code) in the address of the applicant. |
| COUNTRY_CD | VARCHAR(3) | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| GENDER_CD | VARCHAR(3) | Code used to specify the gender of the applicant. |
| REJECTION_REASON_CD | VARCHAR(6) | A code that represents an employment application rejection reason. |
| APPLICANT_BIRTH_DT | DATE | Birth date of the applicant. |
| ETHNICITY_CD | VARCHAR(10) | A unique code to define applicant's ethnic origin. |
| EMAIL_ADDRESS_TXT | VARCHAR(100) | E-mail address of the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| HOME_TELEPHONE_NO | VARCHAR(20) | Home telephone number of the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| DAYTIME_TELEPHONE_NO | VARCHAR(20) | Daytime telephone number for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| MOBILE_TELEPHONE_NO | VARCHAR(20) | Mobile phone number for the applicant. Requires job modifications when this column needs to be included in SAS Human Capital Management. |

| Name | Data Type | Comment |
|---|---|---|
| SOCIAL_SECURITY_NO | VARCHAR(20) | Social security number of the applicant. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| COUNTY_NM | VARCHAR(50) | County name. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.57*** *ENROLLMENT_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| ENROLLMENT_STATUS_CD | VARCHAR(10) | Code for enrollment status. Examples: Regular, night, continuing education. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ENROLLMENT_STATUS_DESC | VARCHAR(255) | Description of the enrollment status. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.58 ETHNICITY Table*

| Name | Data Type | Comment |
|---|---|---|
| ETHNICITY_CD | VARCHAR(10) | A unique code to define employee's ethnic origin. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the ethnicity code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the ethnicity code is valid to. |
| ETHNICITY_DESC | VARCHAR(255) | Description of the ethnicity code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.59 EVIDENCE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| EVIDENCE_TYPE_CD | VARCHAR(10) | A code identifying the type of evidence of the competency. Example: CER for Certificate, DEG for Degree. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| EVIDENCE_TYPE_DESC | VARCHAR(255) | Description of the evidence type. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.60*** *EXEMPT_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| EXEMPT_STATUS_CD | VARCHAR(10) | A unique code representing an employee's exempt status under the Fair Labor Standards Act. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the exempt status code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the exempt status code is valid to. |
| EXEMPT_STATUS_DESC | VARCHAR(255) | Description of the status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.61*** *EXTERNAL_ORG Table*

| Name | Data Type | Comment |
|---|---|---|
| EXTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| EXTERNAL_ORG_ID | VARCHAR(32) | Unique customer-provided ID that identifies the members of the table. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| ORGANIZATION_NM | VARCHAR(50) | Short name used to describe the external organization. |
| ORGANIZATION_LEGAL_NM | VARCHAR(50) | Long name used to describe the external organization. |
| DOING_BUSINESS_AS_NM | VARCHAR(50) | "Doing business as" name. A legally registered alternate name for the entity. |
| ORGANIZATION_DESC | VARCHAR(255) | Description of the external organization. |
| ORGANIZATION_TYPE_CD | VARCHAR(10) | An ORGANIZATION_TYPE_CD value from the ORG_TYPE table. |
| INDUSTRY_CD | VARCHAR(10) | Code that represents the industry of the external organization. |
| MAIN_TELEPHONE_NO | VARCHAR(20) | Main telephone number for the organization. |
| MAIN_FAX_NO | VARCHAR(20) | Main fax number for the organization. |
| FISCAL_CALENDAR_START_DAY_MTH | NUMERIC(4) | Starting day and month of the external organization's fiscal year. |
| CENTRAL_GOVERNMENT_TAX_ID | VARCHAR(20) | Unique government-provided tax ID number. |
| DUNS_NO | VARCHAR(20) | The D&B D-U-N-S Number is D&B's distinctive nine-digit identification sequence; it links products and services originating exclusively from D&B. |

| Name | Data Type | Comment |
|---|---|---|
| EXCHANGE_SYMBOL_CD | VARCHAR(10) | Exchange symbol used. |
| EXTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Identifies if this member's values rolls up into its parent. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.62*** *EXTERNAL_ORG_ADDRESS Table*

| Name | Data Type | Comment |
|---|---|---|
| EXTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| ADDRESS_TYPE_CD | VARCHAR(10) | Code for an address type. Typical address types are business, shipping, mailing, and primary residence. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ADDRESS_LINE_1_TXT | VARCHAR(100) | First line of address. |
| ADDRESS_LINE_2_TXT | VARCHAR(100) | Second line of address. |
| ADDRESS_LINE_3_TXT | VARCHAR(100) | Third line of address. |
| ADDRESS_LINE_4_TXT | VARCHAR(100) | Fourth line of address. |
| CITY_NM | VARCHAR(50) | City name. |
| STATE_REGION_CD | VARCHAR(10) | Unique code associated with states or regions. Examples include "AK" for Alaska, "AL" for Alabama, and "AR" for Arkansas. |
| POSTAL_CD | VARCHAR(20) | Postal code. |

| Name | Data Type | Comment |
|---|---|---|
| COUNTRY_CD | VARCHAR(3) | This column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.63 EXTERNAL_ORG_ASSOC Table*

| Name | Data Type | Comment |
|---|---|---|
| EXTERNAL_ORG_RK | NUMERIC(10) | The EXTERNAL_ORG_RK value from the EXTERNAL_ORG table for the child in a parent-child relationship. |
| PARENT_EXTERNAL_ORG_RK | NUMERIC(10) | The EXTERNAL_ORG_RK value from the EXTERNAL_ORG table for the parent in a parent-child relationship. |
| EXTERNAL_ORG_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |
| EXTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.64** EXTERNAL_ORG_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| EXTERNAL_ORG_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| EXTERNAL_ORG_ASSOC_TYPE_DESC | VARCHAR(255) | A description of the hierarchy. |
| EXTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| DEFAULT_EXTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.65** EXTERNAL_ORG_CONTACT Table*

| Name | Data Type | Comment |
|---|---|---|
| CONTACT_RK | NUMERIC(10) | Unique ID to identify external organization contact records. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| CONTACT_ROLE_CD | VARCHAR(3) | Code for a contact role. |

| Name | Data Type | Comment |
|---|---|---|
| CONTACT_ID | VARCHAR(32) | Unique ID of the contact within the external organization. |
| EXTERNAL_ORG_RK | NUMERIC(10) | Since source data for EXTERNAL_ORG can come from multiple systems, the business-supplied keys might not be unique. A surrogate key is added by the DDS job to ensure a unique identifier for EXTERNAL_ORG. Used with VALID_FROM_DTTM for versioning of rows. |
| FIRST_NM | VARCHAR(50) | Contact first name. |
| MIDDLE_NM | VARCHAR(50) | Contact middle initial. |
| LAST_NM | VARCHAR(50) | Contact last name. |
| JOB_TITLE | VARCHAR(20) | Contact job title. |
| DEPARTMENT_NM | VARCHAR(50) | Name of department that contact person works in or reports to. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.66*** *EXTERNAL_ORG_NLS Table*

| Name | Data Type | Comment |
|---|---|---|
| EXTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORGANIZATION_NM | VARCHAR(50) | Short name used to describe the external organization. |

| Name | Data Type | Comment |
|---|---|---|
| ORGANIZATION_LEGAL_NM | VARCHAR(50) | Long name used to describe the external organization. |
| DOING_BUSINESS_AS_NM | VARCHAR(50) | "Doing business as" name. A legally registered alternate name for the entity. |
| ORGANIZATION_DESC | VARCHAR(255) | Description of the external organization. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.67*** *FICE Table*

| Name | Data Type | Comment |
|---|---|---|
| FICE_CD | VARCHAR(10) | The institution's FICE Code as identified by the Federal Interagency Committee on Education. FICE codes are assigned by the U.S. Department of Education and are used as the primary identifiers for academic institutions. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| FICE_DESC | VARCHAR(255) | Description of the FICE code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.68 FLSA_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| FLSA_STATUS_CD | VARCHAR(10) | Employee FLSA (Fair Labor Standards Act) status code. Can be the same as the exempt status code. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the FLSA status code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the FLSA status code is valid to. |
| FLSA_STATUS_DESC | VARCHAR(255) | Description of the FLSA status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.69 GENDER Table*

| Name | Data Type | Comment |
|---|---|---|
| GENDER_CD | VARCHAR(3) | Code used to specify the gender. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the gender code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the gender code is valid to. |

| Name | Data Type | Comment |
|---|---|---|
| GENDER_DESC | VARCHAR(255) | Description of the gender code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.70*** *GRADUATING_DEGREE Table*

| Name | Data Type | Comment |
|---|---|---|
| GRADUATING_DEGREE_CD | VARCHAR(10) | A code associated with the completion of the education. Examples: G for graduating, Q for qualifying, A for attending. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| GRADUATING_DEGREE_DESC | VARCHAR(255) | Description of the graduating degree. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.71*** *HONORS_PROGRAM Table*

| Name | Data Type | Comment |
|---|---|---|
| HONORS_PROGRAM_CD | VARCHAR(10) | Code for any honors program associated with the degree. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| HONORS_PROGRAM_DESC | VARCHAR(255) | Description of the honors program. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.72*** *INDUSTRY Table*

| Name | Data Type | Comment |
|---|---|---|
| INDUSTRY_CD | VARCHAR(10) | Code for an industry. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| INDUSTRY_DESC | VARCHAR(255) | Description of the industry that is identified in INDUSTRY_CD. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.73 INTERNAL_ORG Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table, based on the INTERNAL_ORG_ID value in the INTERNAL_ORG stage table. This column is used by a look-up in the DDS jobs. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. The date and time that the INTERNAL_ORG_ID is valid from. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the INTERNAL_ORG_ID is valid to. |
| STATE_REGION_CD | VARCHAR(10) | A STATE_REGION_CD value from the STATE_REGION table. |
| COUNTRY_CD | VARCHAR(3) | This column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. |
| INTERNAL_ORG_ID | VARCHAR(32) | Unique ID used to identify an internal organization. This column is used by a look-up in the DDS jobs. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| ORGANIZATION_NM | VARCHAR(50) | Short name used to describe an internal organization. |

| Name | Data Type | Comment |
|---|---|---|
| ORGANIZATION_DESC | VARCHAR(255) | Long name used to describe an internal organization. |
| ORGANIZATION_TYPE_CD | VARCHAR(10) | An ORGANIZATION_TYPE_CD value from the ORG_TYPE table. |
| MANAGING_EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table that identifies the manager of this organization. This column is populated by the DDS job, based on the MANAGING_EMPLOYEE_ID value in the INTERNAL_ORG stage table. |
| ADDRESS_LINE_1_TXT | VARCHAR(100) | Line 1 of an address. |
| ADDRESS_LINE_2_TXT | VARCHAR(100) | Line 2 of an address. |
| ADDRESS_LINE_3_TXT | VARCHAR(100) | Line 3 of an address. |
| ADDRESS_LINE_4_TXT | VARCHAR(100) | Line 4 of an address. |
| CITY_NM | VARCHAR(50) | City name. |
| POSTAL_CD | VARCHAR(20) | Postal code. |
| COST_CENTER_RK | NUMERIC(10) | A COST_CENTER_RK value from the COST_CENTER table. |
| BOOK_OF_RECORD_CURRENCY_CD | VARCHAR(3) | This column is not used. |
| REPORTING_CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. For SAS Financial Management, this is the currency that this organization reports in. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| LEGAL_ENTITY_FLG | CHAR(1) | Flag Y, if this organization is a legal entity. |
| EXTERNAL_TRADER_FLG | CHAR(1) | This column is not used. |
| COUNTY_NM | VARCHAR(50) | County name. |
| INTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Identifies if this member's values rolls up into its parent. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.74 INTERNAL_ORG_ASSOC Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_RK | NUMERIC(10) | The INTERNAL_ORG_RK value from the INTERNAL_ORG table for the child in a parent-child relationship. This value is generated by the DDS job, based on the INTERNAL_ORG_ID value in the stage INTERNAL_ORG_ASSOC table. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| PARENT_INTERNAL_ORG_RK | NUMERIC(10) | The INTERNAL_ORG_RK value from the INTERNAL_ORG table for the parent in a parent-child relationship. This value is generated by the DDS job, based on the PARENT_INTERNAL_ORG_ID value in the stage INTERNAL_ORG_ASSOC table. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| INTERNAL_ORG_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. The date and time which the INTERNAL_ORG_ID relationship to the PARENT_INTERNAL_ORG_ID is valid from. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. The date and time which the INTERNAL_ORG_ID relationship to the PARENT_INTERNAL_ORG_ID is valid to. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |
| INTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.75*** *INTERNAL_ORG_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. This column is required by SAS Human Capital Management. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the INTERNAL_ORG_ASSOC_TYPE_CD is valid from. This column is required by SAS Human Capital Management. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the INTERNAL_ORG_ASSOC_TYPE_CD is valid to. |
| INTERNAL_ORG_ASSOC_TYPE_DESC | VARCHAR(255) | A description of the hierarchy. |

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_ADK | VARCHAR(32) | This column is not used. |
| DEFAULT_INTERNAL_ORG_RK | NUMERIC(10) | The INTERNAL_ORG_RK value from the INTERNAL_ORG table for the default member of an organization hierarchy. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.76*** *INTERNAL_ORG_NLS Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORGANIZATION_NM | VARCHAR(50) | Short name used to describe an internal organization in the associated language/locale. |
| ORGANIZATION_DESC | VARCHAR(255) | Long name used to describe an internal organization in the associated language/locale. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.77* ITEM_CATEGORY Table

| Name | Data Type | Comment |
|---|---|---|
| ITEM_CATEGORY_CD | VARCHAR(3) | Unique code indicating item category. Can be user-defined or industry standard. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ITEM_CATEGORY_DESC | VARCHAR(255) | Text description for the unique item category code. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Identifies if this member's values rolls up into its parent. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.78* ITEM_CATEGORY_ASSOC Table

| Name | Data Type | Comment |
|---|---|---|
| ITEM_CATEGORY_CD | VARCHAR(3) | Unique code indicating item category. Can be user-defined or industry standard. |
| PARENT_ITEM_CATEGORY_CD | VARCHAR(3) | Parent code used lower in the commodity-coding hierarchy. |
| ITEM_CATEGORY_ASSOC_TYPE_CD | VARCHAR(32) | Cross-reference element indicating association between category and type codes. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| ORDER_NO | NUMERIC(8) | Identifies the order in which the members should be listed for a given hierarchy level. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.79*** *ITEM_CATEGORY_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| ITEM_CATEGORY_ASSOC_TYPE_CD | VARCHAR(32) | Cross-reference element indicating association between category and type codes. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ITEM_CATEGORY_ASSOC_TYPE_DESC | VARCHAR(255) | Text description of the ITEM_CATEGORY_ASSOC_TYPE_CD. |
| DEFAULT_ITEM_CATEGORY_CD | VARCHAR(3) | Item Category Code that is identified as the default. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.80*** *JOB Table*

| Name | Data Type | Comment |
|---|---|---|
| JOB_CD | VARCHAR(10) | A code that represents a job within the company. This column is used for data validation in DDS jobs |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the job code is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the job code is valid to. |
| ACTIVE_FLG | CHAR(1) | Flag to indicate whether the job is active or inactive. |
| JOB_TITLE_TXT | VARCHAR(100) | Title for the job. |
| JOB_GROUP_CD | VARCHAR(10) | Job group that this job belongs to. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.81*** *JOB_GROUP Table*

| Name | Data Type | Comment |
|---|---|---|
| JOB_GROUP_CD | VARCHAR(10) | Job group code. This is usually a grouping of jobs. It is usually associated with only one EEO category. This column is used for data validation in DDS jobs. This column is required for the build of the SAS Human Capital Management JOBS table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the job group code is valid from. This column is required for the build of the SAS Human Capital Management JOBS table. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the job group code is valid to. |
| JOB_GROUP_DESC | VARCHAR(255) | Description of the job group. |

| Name | Data Type | Comment |
|---|---|---|
| EEO_CLASS_CD | VARCHAR(10) | Standard EEO classification codes. EEO1 classifications are used for corporations and EEO4 classifications are used for government reporting at the federal, state and local levels. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.82*** *JOB_POSITION Table*

| Name | Data Type | Comment |
|---|---|---|
| POSITION_CD | VARCHAR(10) | A code that represents a position within the company. This column is used for data validation in DDS jobs. This column is required for the build of the SAS Human Capital Management POS table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the position code is valid from. This column is required for the build of the SAS Human Capital Management JOBS and POS tables. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the position code is valid to. |
| POSITION_TITLE_TXT | VARCHAR(100) | Title of the position. |
| OPEN_DT | DATE | Date on which the position was opened. |
| CLOSE_DT | DATE | Closing date for the position. |
| POSITION_STATUS_CD | VARCHAR(10) | Status of the position. For example: open, canceled, filled. |
| FLSA_STATUS_CD | VARCHAR(10) | Employee FLSA (Fair Labor Standards Act) status code. Can be the same as the exempt status code. |
| CANCELLATION_DT | DATE | Date that the position was canceled. |

| Name | Data Type | Comment |
|---|---|---|
| APPROVAL_DT | DATE | Date that the position was approved. |
| INTERNAL_ORG_RK | NUMERIC(10) | An INTERNAL_ORG_RK value from the INTERNAL_ORG table for the organization that has the position. This column should be populated with INTERNAL_ORG_ID in the stage table. The DDS job looks up the RK value. |
| PAY_LEVEL_CD | VARCHAR(20) | The pay level code that is associated with the position. |
| PAY_LEVEL_STRUCTURE_CD | VARCHAR(20) | Code used to join to the PAY_LEVEL table. It is a unique identifier of a PAY_LEVEL since the PAY_LEVEL column itself cannot be guaranteed to be unique. If the pay levels are unique, this column can be loaded with just the PAY_LEVEL. |
| PAY_FREQUENCY_CD | VARCHAR(3) | Unique code for a time frequency or time span. Defines the type of time frequency. For example: week, bi-weekly, month, year. |
| BENEFIT_PERCENT_NO | NUMERIC(9,4) | Benefit percent associated with the position. This column is not currently used. |
| EXEMPT_STATUS_CD | VARCHAR(10) | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." |
| FTE_RT | NUMERIC(9,4) | Full-time equivalent rate associated with the position. |
| ALLOCATED_TO_COST_CENTER_RK | NUMERIC(10) | A COST_CENTER_RK value from the COST_CENTER table. If there is cost center data, the stage table should be populated with the cost center ID. The DDS job looks up the RK value. Requires job modifications when this column needs to be included in SAS Human Capital Management. |

| Name | Data Type | Comment |
|---|---|---|
| PERMANENCE_CD | VARCHAR(10) | A unique code to identify a position permanence status. For example, R=regular and T=temporary. |
| REQUISITION_COST_AMT | NUMERIC(10,2) | The cost of the position requisition. |
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. |
| REQUISITION_NO | VARCHAR(10) | The requisition number for the position. |
| JOB_CD | VARCHAR(10) | Job code associated with this position. This column is used for data validation in DDS jobs. This column is required for the build of the SAS Human Capital Management JOBS table. |
| JOB_POSITION_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.83*** *JOB_X_COMPETENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| JOB_CD | VARCHAR(10) | A code that represents a job within the company. Reference key used to establish the relationship of the JOB table to the COMPETENCY table. Since jobs are assigned a competency this links these together. |
| COMPETENCY_RK | NUMERIC(10) | Reference key of the competency associated with the COMPETENCY_WEIGHT_BOUNDS table row data. |
| WEIGHT_TYPE_CD | VARCHAR(10) | A code for a weight type. The COMPETENCY_WEIGHT_BOUNDS table can have multiple entries based on weight type. For example, there can be multiple rows representing a required bound, level of interest bound, or skill bound. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the JOB_X_COMPETENCY table data is valid to. |
| COMPETENCY_ID | VARCHAR(32) | A competency code that is defined in the COMPETENCY table. |
| COMPETENCY_IMPORTANCE_VAL | VARCHAR(20) | Importance level for the competency associated with the job. Example: For a senior Java developer, Java is of high importance. For a manager, it might be of low importance. |
| NUMERIC_REQ_LEVEL_VAL | NUMERIC(15,2) | The numeric level required for the job. This is used as a numeric basis of comparison between the job and the employee holding the job. This column is populated when the measurements for the competency and employee are numeric. |
| NUMERIC_REQ_LEVEL_VAL_DESC | VARCHAR(255) | Description for the NUMERIC_REQ_LEVEL_VAL column. |
| STRING_REQ_LEVEL_VAL | VARCHAR(20) | The COMPETENCY table level required. This is used as a basis of comparison between the job and the employee holding the job. This column is populated when the measurements for the competency and employee are character. |
| STRING_REQ_LEVEL_VAL_DESC | VARCHAR(255) | Description for the STRING_REQ_LEVEL_VAL column. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.84*** *MARITAL_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| MARITAL_STATUS_CD | VARCHAR(3) | Code used to define an employee's marital status. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the marital status code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the marital status code is valid to. |
| MARITAL_STATUS_DESC | VARCHAR(255) | Description of the marital status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.85*** *MEASURE Table*

| Name | Data Type | Comment |
|---|---|---|
| MEASURE_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| MEASURE_NM | VARCHAR(50) | Name of the measure that is identified in MEASURE_RK. |

| Name | Data Type | Comment |
|---|---|---|
| MEASURE_DESC | VARCHAR(255) | Description of the measure that is identified in MEASURE_RK. |
| RESPONSIBLE_PARTY_NM | VARCHAR(50) | Name of the person who is responsible for the measure that is identified in MEASURE_RK. |
| DIRECTIVE_TXT | VARCHAR(255) | This column is not used. |
| CATEGORY_TXT | VARCHAR(32) | User-entered string to categorize the measure names. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.86*** *MILITARY_EXPERIENCE Table*

| Name | Data Type | Comment |
|---|---|---|
| EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. |
| MILITARY_EXPERIENCE_TYPE_CD | VARCHAR(10) | Military status or experience. Used to show an employee's military or veteran status. |
| MILITARY_EXPERIENCE_DT | DATE | Date that the military experience started. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.87*** *MILITARY_EXPERIENCE_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| MILITARY_EXPERIENCE_TYPE_CD | VARCHAR(10) | Employee's military status or experience. Used to show military or veteran status. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the military experience type code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the military experience type code is valid to. |
| MILITARY_EXPERIENCE_DESC | VARCHAR(255) | Description of the military experience type code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.88*** *ORG_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| ORGANIZATION_TYPE_CD | VARCHAR(10) | Unique code for a type of organization. Examples include division, department, and subsidiary. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORGANIZATION_TYPE_DESC | VARCHAR(255) | Names describing the types of organization. Examples include division, department, and subsidiary. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.89 OTHER_HONORS Table*

| Name | Data Type | Comment |
|---|---|---|
| OTHER_HONORS_CD | VARCHAR(10) | Code for an honor that a student can receive. Example: Phi Beta Kappa. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| OTHER_HONORS_DESC | VARCHAR(255) | Description of other honors. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.90*** *PAY_LEVEL Table*

| Name | Data Type | Comment |
|---|---|---|
| PAY_LEVEL_CD | VARCHAR(20) | A code that represents a pay level or a set of related pay levels. This column is used for data validation in DDS jobs. |
| PAY_LEVEL_STRUCTURE_CD | VARCHAR(20) | A unique code for a pay level. If PAY_LEVEL_CD is unique for each record, then PAY_LEVEL_STRUCTURE_CD and PAY_LEVEL_CD can be the same in every record. This column is required for the build of the SAS Human Capital Management GRADE table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the pay level code is valid from. This column is required for the build of the SAS Human Capital Management GRADE table. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the pay level code is valid to. |
| MINIMUM_AMT | NUMERIC(10,2) | Minimum amount for this pay range. |
| MIDPOINT_AMT | NUMERIC(10,2) | Midpoint amount of the pay range. This column might not apply to all pay levels. |
| MAXIMUM_AMT | NUMERIC(10,2) | Maximum amount for this pay range. |
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. |

| Name | Data Type | Comment |
|---|---|---|
| PAY_FREQUENCY_CD | VARCHAR(3) | Indicates the pay frequency for the minimum, middle, and maximum amount columns. It is a unique code for a time frequency or time span. It defines the type of time frequency. For example: weekly, bi-weekly, monthly, and yearly. |
| PAY_LEVEL_DESC | VARCHAR(255) | Description of the pay level. |
| ACTIVE_FLG | CHAR(1) | Indicates whether the pay level is active or not. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.91 PAY_LEVEL_STRUCTURE Table*

| Name | Data Type | Comment |
|---|---|---|
| PAY_LEVEL_STRUCTURE_CD | VARCHAR(20) | Code used to join to the PAY_LEVEL table. It is a unique identifier of a pay level since the PAY_LEVEL column cannot be guaranteed to be unique. If the PAY_LEVEL values are unique, then this column can be loaded with the PAY_LEVEL values. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. Requires job modifications when this column needs to be included in SAS Human Capital Management. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the pay level structure code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the pay level structure code is valid to. |
| PAY_LEVEL_STRUCTURE_DESC | VARCHAR(255) | Description of the pay level structure code. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.92*** *POSITION_PERMANENCE Table*

| Name | Data Type | Comment |
|---|---|---|
| PERMANENCE_CD | VARCHAR(10) | A unique code to identify a position permanence status. For example R=regular, T=temporary. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the permanence code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the permanence code is valid to. |
| PERMANENCE_DESC | VARCHAR(255) | Description of the permanence code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.93*** *POSITION_STATUS Table*

| Name | Data Type | Comment |
|---|---|---|
| POSITION_STATUS_CD | VARCHAR(10) | Status of the position. For example: open, canceled, filled. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the position status code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the position status code is valid to. |
| POSITION_STATUS_DESC | VARCHAR(255) | The description of the position status code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.94 PROFIT_CENTER Table*

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| PROFIT_CENTER_ID | VARCHAR(32) | Unique ID for a functional area within an organization, to which revenues and costs are assigned. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| RESPONSIBLE_EMPLOYEE_RK | NUMERIC(10) | An EMPLOYEE_RK value from the EMPLOYEE table. |

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_NM | VARCHAR(50) | Short name for describing categories that identify the division of functional areas within an organization. These categories are typically nonphysical entities to which revenues and costs are assigned. |
| PROFIT_CENTER_DESC | VARCHAR(255) | Long name for describing categories that identify the division of functional areas within an organization. These categories are typically nonphysical entities to which revenues and costs are assigned. |
| PROFIT_CENTER_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Indicates whether this member's values roll up to its parent. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.95*** *PROFIT_CENTER_ASSOC Table*

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_RK | NUMERIC(10) | The PROFIT_CENTER_RK value from the PROFIT_CENTER table for the child in a parent-child relationship. |
| PARENT_PROFIT_CENTER_RK | NUMERIC(10) | The PROFIT_CENTER_RK value from the PROFIT_CENTER table for the parent in a parent-child relationship. |
| PROFIT_CENTER_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.96*** *PROFIT_CENTER_ASSOC_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the hierarchies represented in the association table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| PROFIT_CENTER_ASSOC_TYPE_DESC | VARCHAR(255) | A description of the hierarchy. |
| PROFIT_CENTER_ADK | VARCHAR(32) | This column is not used. |
| DEFAULT_PROFIT_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.97*** *PROFIT_CENTER_NLS Table*

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| PROFIT_CENTER_NM | VARCHAR(50) | Short name for describing categories that identify the division of functional areas within an organization of the associated language/locale. These categories are typically nonphysical entities to which revenues and costs are assigned. |
| PROFIT_CENTER_DESC | VARCHAR(255) | Long name for describing categories that identify the division of functional areas within an organization in the associated language/locale. These categories are typically nonphysical entities to which revenues and costs are assigned. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.98*** *RECRUITMENT_SOURCE Table*

| Name | Data Type | Comment |
|---|---|---|
| RECRUITMENT_SOURCE_CD | VARCHAR(10) | A code that represents various recruitment sources. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the recruitment source code is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the recruitment source code is valid to. |
| RECRUITMENT_SOURCE_DESC | VARCHAR(255) | Description of the recruitment source. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.99*** *REJECTION_REASON Table*

| Name | Data Type | Comment |
|---|---|---|
| REJECTION_REASON_CD | VARCHAR(6) | A code that represents an employment application rejection reason. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the rejection reason code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the rejection reason code is valid to. |
| REJECTION_REASON_DESC | VARCHAR(255) | Description of the rejection reason code. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.100*** *RETAINED_EARN_ROLL_FWD_METHOD Table*

| Name | Data Type | Comment |
|---|---|---|
| RETAINED_EARN_ROLL_FWD_CD | VARCHAR(32) | A predefined code from the SAS_RETAINED_EARN_ROLL_FWD_CD staging table. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| RETAINED_EARN_ROLL_FWD_DESC | VARCHAR(255) | Description of the roll-forward method that is identified in RETAINED_EARN_ROLL_FWD_CD. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.101*** *SASOP_DETAIL Table*

| Name | Data Type | Comment |
|---|---|---|
| INTERNAL_ORG_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table, based on the INTERNAL_ORG_ID value in the INTERNAL_ORG stage table. This column is used by a look-up in the DDS jobs. This column is required by SAS Human Capital Management for the build of the WRKGRP table. |
| GL_ACCOUNT_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| ANALYSIS_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| CURRENCY_CD | VARCHAR(3) | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar. This column is used for data validation in DDS ETL jobs. |
| COST_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |

| Name | Data Type | Comment |
|---|---|---|
| PROFIT_CENTER_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| ITEM_CATEGORY_CD | VARCHAR(3) | Unique code indicating item category. Can be user-defined or industry standard. |
| TIME_PERIOD_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| TRANSACTION_AMT | NUMERIC(14,2) | Amount or value of transaction. Use traditional debit or credit signs for values. Debit values are positive; credit values are negative. |
| TRANSACTION_AMT_YTD_FLG | CHAR(1) | Valid values are "Y" or "N". "Y" means to compute the period activity value by subtracting TRANSACTION_AMT for the previous period from TRANSACTION_AMT for the current period. "N" means to take TRANSACTION_AMT for the current period as the period activity value. |
| TRANSACTION_DT | DATE | Date of transaction record. |
| PROCESSED_DTTM | DATE | The timestamp for the last time a record was processed, typically by ETL load processing. The timestamp could also be updated when inter-ETL cycle modifications are made to a record. |

***Table 36.102*** *SCHOOL_DEPT Table*

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_DEPT_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the SCHOOL_DEPT table data is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the SCHOOL_DEPT table data is valid to. |
| SCHOOL_DEPT_ID | VARCHAR(32) | Business key for SCHOOL_DEPT table. |
| INTERNAL_ORG_RK | NUMERIC(10) | Reference key of the internal organization associated with the school department. |
| EXTERNAL_ORG_RK | NUMERIC(10) | Reference key of the external organization associated with the school department. |
| SCHOOL_OR_INSTITUTION_RK | NUMERIC(10) | Reference key of the school or institution associated with the school department. |
| SCHOOL_DEPT_NM | VARCHAR(40) | Name of the school department. |
| SCHOOL_DEPT_TYPE_CD | VARCHAR(10) | A department type code that is defined in the SCHOOL_DEPT_TYPE table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.103*** *SCHOOL_DEPT_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_DEPT_TYPE_CD | VARCHAR(10) | A type code for a school department. Examples: "CS" for Computer Science, "PT" for Physical Therapy. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| SCHOOL_DEPT_TYPE_DESC | VARCHAR(255) | Description of the school department. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.104*** *SCHOOL_NAME_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_NAME_TYPE_CD | VARCHAR(10) | A code that is used to distinguish different types of educational institution. Example: "PU" for a public institution. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| SCHOOL_NAME_TYPE_DESC | VARCHAR(255) | Description of the school name type. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.105*** *SCHOOL_OR_INSTITUTION Table*

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_OR_INSTITUTION_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the SCHOOL_OR_INSTITUTION table data is valid from. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the SCHOOL_OR_INSTITUTION table data is valid to. |
| SCHOOL_OR_INSTITUTION_ID | VARCHAR(32) | Business key for school or institution. |
| INTERNAL_ORG_RK | NUMERIC(10) | Reference key of the internal organization associated with the school or institution. |
| EXTERNAL_ORG_RK | NUMERIC(10) | Reference key of the External Organization associated with the school or institution. |
| SCHOOL_TYPE_CD | VARCHAR(10) | Type code used to distinguish different types of educational institutions. Examples: community college, trade school, university. |
| SCHOOL_NM | VARCHAR(40) | Name of the educational institution. |
| SCHOOL_NAME_TYPE_CD | VARCHAR(10) | Type code to qualify the school name. Can be used as a broader level classification of the school or institution. |
| FICE_CD | VARCHAR(10) | A FICE code that is defined in the FICE table. |
| INTERNET_DOMAIN_NM | VARCHAR(40) | A domain name intended as a practical identifier for the organization, typically used for Web and e-mail. Not intended as a Web address (URL). Examples: microsoft.com, bund.de, google.com. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.106*** *SCHOOL_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| SCHOOL_TYPE_CD | VARCHAR(10) | A code used to distinguish different types of educational institutions. Examples: community college, trade school, university. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| SCHOOL_TYPE_DESC | VARCHAR(255) | Description of the school type. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.107 SOURCE_SYSTEM Table*

| Name | Data Type | Comment |
|---|---|---|
| SOURCE_SYSTEM_CD | VARCHAR(3) | A valid source system code from the SAS_SOURCE_SYSTEM table. These codes are used in the SOURCE_SYSTEM_CD columns of other tables to indicate how the records in those tables originated. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the source system code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the source system code is valid to. |

| Name | Data Type | Comment |
|---|---|---|
| SOURCE_SYSTEM_DESC | VARCHAR(255) | Description of the source system. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.108*** *SPECIAL_REF_COMP Table*

| Name | Data Type | Comment |
|---|---|---|
| SPECIAL_REF_COMP_CD | VARCHAR(10) | Special reference for a competency code. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| SPECIAL_REF_COMP_DESC | VARCHAR(255) | Description of the special reference competency. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.109*** *STATE_REGION Table*

| Name | Data Type | Comment |
|---|---|---|
| STATE_REGION_CD | VARCHAR(10) | Unique code for a state or region. Examples include "AK" for Alaska, "AL" for Alabama, and "AR" for Arkansas. This column is used for data validation in DDS jobs. |

| Name | Data Type | Comment |
|---|---|---|
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the state region code is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the state region code is valid to. |
| STATE_REGION_NM | VARCHAR(50) | Name describing states or regions. Examples are Alaska, Alabama, and Midwest. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.110*** *TAXONOMY Table*

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_CD | VARCHAR(10) | A code assigned to a competency that is used to associate it for comparison with other competencies internally or externally. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| TAXONOMY_DESC | VARCHAR(255) | Description of the taxonomy. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.111*** *TAXONOMY_CATALOG Table*

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_CATALOG_CD | VARCHAR(10) | Code for the taxonomy catalog. If the taxonomy is identified within a catalog, this identifies the catalog. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| TAXONOMY_CATALOG_DESC | VARCHAR(255) | Description of the taxonomy catalog. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.112*** *TAXONOMY_COMP_CLASS Table*

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_COMP_CLASS_CD | VARCHAR(10) | Code for the taxonomy competency class used to classify competencies within a higher level classification. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| TAXONOMY_COMP_CLASS_DESC | VARCHAR(255) | Description of the taxonomy competency class. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.113 TAXONOMY_SOURCE Table*

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_SOURCE_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. This is the date and time that the TAXONOMY_SOURCE table data is valid from. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. This is the date and time that the TAXONOMY_SOURCE table data is valid to. |
| TAXONOMY_SOURCE_ID | VARCHAR(32) | Business key for taxonomy source. |
| TAXONOMY_CD | VARCHAR(10) | A taxonomy code that is defined in the TAXONOMY table. |
| TAXONOMY_OWNER_NM | VARCHAR(40) | Owner name for the taxonomy source. |
| TAXONOMY_DESC | VARCHAR(255) | Description of the taxonomy source. |
| TAXONOMY_COMP_CLASS_CD | VARCHAR(10) | A broad class code used to identify the taxonomy source. |

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_CATALOG_CD | VARCHAR(10) | Catalog code for the taxonomy source. |
| TAXONOMY_IMPORTANCE_VAL | VARCHAR(20) | Importance value for the taxonomy source. Multiple taxonomies can be used for comparing a competency. This column contains a rating of importance for this source. |
| TAXONOMY_LEVEL_VAL | VARCHAR(20) | Level value for the taxonomy source. |
| TAXONOMY_SCALE_VAL | VARCHAR(20) | Scale value for the taxonomy source. |
| TAXONOMY_SPECIAL_REF_COMP_CD | VARCHAR(10) | Special reference code for the taxonomy source. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.114*** *TAXONOMY_SPECIAL_REF_COMP Table*

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_SPECIAL_REF_COMP_CD | VARCHAR(10) | Special reference code for the taxonomy source. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |

| Name | Data Type | Comment |
|---|---|---|
| TAXONOMY_SPECIAL_REF_COMP_DESC | VARCHAR(255) | Description of the special reference competency for the taxonomy. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.115*** *TIME_FREQUENCY Table*

| Name | Data Type | Comment |
|---|---|---|
| TIME_FREQUENCY_CD | VARCHAR(3) | Unique code for a time frequency or time span. Defines the type of time frequency. For example: week, bi-weekly, month, year. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| TIME_FREQUENCY_DESC | VARCHAR(255) | A description of the time frequency or time span. |
| HOURS_PER_PERIOD_QTY | NUMERIC(9,4) | Number of hours associated with the time frequency code. For example, a time frequency code of WEEK represents 168 hours. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.116*** *TIME_PERIOD Table*

| Name | Data Type | Comment |
|---|---|---|
| TIME_PERIOD_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| SOURCE_SYSTEM_CD | VARCHAR(3) | A code that indicates how each record originated. The valid codes are defined in the SAS_SOURCE_SYSTEM table. If no code is specified in the staging table, then "ETL" is supplied automatically by the job that loads the DDS table. |
| TIME_PERIOD_ID | VARCHAR(32) | Unique ID for a time period. Examples include AllYears, YR2002, 1stQtr2002, and Jan2002. |
| TIME_PERIOD_NM | VARCHAR(50) | Short name describing a time period. |
| TIME_PERIOD_DESC | VARCHAR(255) | Long name describing a time period. |
| PERIOD_TYPE_CD | VARCHAR(3) | A PERIOD_TYPE_CD value from the PERIOD_TYPE table. |
| START_DTTM | DATE | Actual start date represented by the time period ID. |
| END_DTTM | DATE | Actual end date represented by the time period ID. |
| TIME_PERIOD_ADK | VARCHAR(32) | This column is not used. |
| ROLL_UP_TO_PARENT_FLG | CHAR(1) | Identifies if this member's values rolls up into its parent. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.117* TIME_PERIOD_ASSOC Table

| Name | Data Type | Comment |
|---|---|---|
| TIME_PERIOD_RK | NUMERIC(10) | The TIME_PERIOD_RK value from the TIME_PERIOD table for the child in a parent-child relationship. |
| PARENT_TIME_PERIOD_RK | NUMERIC(10) | The TIME_PERIOD_RK value from the TIME_PERIOD table for the parent in a parent-child relationship. |
| TIME_PERIOD_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the types of hierarchies represented in the association table. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| ORDER_NO | NUMERIC(8) | Identifies the order in which the nodes should be listed for a given hierarchy level. |
| TIME_PERIOD_ADK | VARCHAR(32) | This column is not used. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.118* TIME_PERIOD_ASSOC_TYPE Table

| Name | Data Type | Comment |
|---|---|---|
| TIME_PERIOD_ASSOC_TYPE_CD | VARCHAR(32) | Code used to identify the types of hierarchies represented in the association table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| TIME_PERIOD_ASSOC_TYPE_DESC | VARCHAR(255) | Time Period Association Type name used to describe the code used to identify the types of hierarchies represented in the association table. |
| TIME_PERIOD_ADK | VARCHAR(32) | This column is not used. |
| DEFAULT_TIME_PERIOD_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

***Table 36.119*** *TIME_PERIOD_NLS Table*

| Name | Data Type | Comment |
|---|---|---|
| TIME_PERIOD_RK | NUMERIC(10) | Surrogate integer-valued key that is generated by the job that loads the table. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| TIME_PERIOD_NM | VARCHAR(50) | Short name describing a time period in the associated language/locale. |
| TIME_PERIOD_DESC | VARCHAR(255) | Long name describing a time period in the associated language/locale. |

| Name | Data Type | Comment |
|---|---|---|
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.120 TIME_UNIT_OF_MEASURE Table*

| Name | Data Type | Comment |
|---|---|---|
| TIME_UOM_CD | VARCHAR(3) | Unique code for a type of unit of measure. This column is used for data validation in DDS jobs. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |
| VALID_TO_DTTM | DATE | The moment that ends the period of time during which a row of data is valid. |
| TIME_UOM_DESC | VARCHAR(255) | Name describing units of measure. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Table 36.121 WEIGHT_TYPE Table*

| Name | Data Type | Comment |
|---|---|---|
| WEIGHT_TYPE_CD | VARCHAR(10) | A code for a type of weighting that is used for measurement. Examples: "LOI" for level of interest, "REQ" for required, "SKL" for skill. |
| LANGUAGE_CD | VARCHAR(3) | A code that identifies the language/ locale for names and descriptions. The code must be defined in the CODE_LANGUAGE table. An example is "en" for English. |
| VALID_FROM_DTTM | DATE | The moment that begins the period of time during which a row of data is valid. |

| Name | Data Type | Comment |
|---|---|---|
| VALID_TO_DTTM | DATE | Standard dates used for versioning. The row content is valid within the time range specified by from and to dates. For a given identifier, versions of its rows are distinguished by different nonoverlapping from and to date ranges. |
| WEIGHT_TYPE_DESC | VARCHAR(255) | Description of the weight type. |
| PROCESSED_DTTM | DATE | The moment at which a row of data was created or most recently modified by the job that loads the table. This time stamp is supplied by the job. |

*Chapter 37*

# More Information about DDS Tables

**More Information about Detail Data Store Tables** . . . . . . . . . . . . . . . . . . . . . . . . . 509

## More Information about Detail Data Store Tables

The following tables contain a variety of information about the DDS tables used by SAS Human Capital Management for which there are table-loading jobs. The information provided for each table includes the key columns and any look-up or validation dependencies on other tables. The information is presented according to the sequence number of the associated job.

***Table 37.1*** *COUNTRY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COUNTRY | 100100 | N/A | | COUNTRY_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COUNTRY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | yes | SAS_COUNTRY_ISO3166 | This table contains a list of unique two-character country codes as defined by ISO 3166 standard. Validation for the LANGUAGE_CD column goes to the SAS-supplied language table and the ISO code. This table is used to build the HCM formats. | HCM formats catalog Required |

*Table 37.2 CODE_LANGUAGE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| CODE_LANGUAGE | 100200 | COUNTRY | | LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| LANGUAGE_CD<br>VALID_FROM_DTTM | yes | SAS_LANGUAGE_ISO0639 | | Required |

*Table 37.3 TIME_FREQUENCY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TIME_FREQUENCY | 100700 | CODE_LANGUAGE | | LANGUAGE_CD<br>TIME_FREQUENCY_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TIME_FREQUENCY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.4 ABSENCE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ABSENCE_TYPE | 100800 | CODE_LANGUAGE | | ABSENCE_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ABSENCE_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | The types of absence that an employee can take are defined in this table. Examples are sickness, vacation, disability, family leave, and bereavement. This table is used to build the HCM formats. | HCM formats catalog |

***Table 37.5*** *ACTION_REASON*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ACTION_REASON | 100900 | CODE_LANGUAGE | | EMPLOYEE_ACTION_REASON_ CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ACTION_REASON_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table describes the reason for a specific action associated with an employee. This table is used to build the HCM formats. | HCM formats catalog<br>HCM Acthist Table |

***Table 37.6*** *ACTION_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ACTION_TYPE | 101000 | CODE_LANGUAGE | | ACTION_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ACTION_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines the type of employee action, such as pay increase, probation, or suspension. This table is used to build the HCM formats. | HCM formats catalog |

***Table 37.7*** *ADDRESS_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ADDRESS_TYPE | 101100 | CODE_LANGUAGE |  | ADDRESS_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ADDRESS_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM |  |  |  |  |

***Table 37.8*** *APP_SCHEMA*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| APP_SCHEMA | 101800 | N/A |  | SCHEMA_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHEMA_ID<br>VALID_FROM_DTTM (Index)<br>SCHEMA_RK<br>VALID_FROM_DTTM (Prim_Key) |  |  |  |  |

***Table 37.9*** *APP_SCHEMA_NLS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| APP_SCHEMA_NLS | 101830 | CODE_LANGUAGE | SCHEMA_RK (APP_SCHEMA) |  |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHEMA_ID<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>SCHEMA_RK<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.10 APPLICATION_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| APPLICATION_STATUS | 101900 | CODE_LANGUAGE | | APPLICATION_STATUS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| APPLICATION_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>APPLICATION_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | This table contains status codes for employment applications. This table is used to build the HCM formats. | HCM formats catalog |

*Table 37.11 COMPENSATION_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPENSATION_TYPE | 102100 | CODE_LANGUAGE | | LANGUAGE_CD<br>COMPENSATION_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPENSATION_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>COMPENSATION_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | This table defines types of additional compensation, such as commissions or bonuses. This table is used to build the HCM formats. | |

*Table 37.12* CONTACT_ROLE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| CONTACT_ROLE | 102200 | CODE_LANGUAGE | | CONTACT_ROLE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| CONTACT_ROLE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>CONTACT_ROLE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.13* COUNTY

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COUNTY | 102300 | CODE_LANGUAGE | | STATE_REGION_CD<br>COUNTY_NM<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| STATE_REGION_CD<br>COUNTY_NM<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table contains codes that identify counties in a state or region. | HCM formats catalog |

*Table 37.14* CURRENCY

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| CURRENCY | 102400 | CODE_LANGUAGE | | CURRENCY_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| CURRENCY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | yes | SAS_CURRENCY | | |

*Table 37.15* EDUCATION_LEVEL

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EDUCATION_LEVEL | 102800 | CODE_LANGUAGE | | EDUCATION_LEVEL_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_LEVEL_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines the employee's level of education. | |

*Table 37.16* EEO_CLASS

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EEO_CLASS | 102900 | CODE_LANGUAGE | | EEO_CLASS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EEO_CLASS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table contains the standard EEO classification codes. EEO1 classifications are used for corporations and EEO4 classifications are used for government reporting at the federal, state, and local level. This table is used to build the HCM formats. | |

*Table 37.17* EMPLOYEE_INVOLVEMENT_TYPE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_INVOLVEMENT_TYPE | 103000 | CODE_LANGUAGE | | EMPLOYEE_INVOLVEMENT_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_INVOLVEMENT_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | Optional HCM table. | |

***Table 37.18*** *EMPLOYEE_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_STATUS | 103100 | CODE_LANGUAGE | | LANGUAGE_CD<br>EMLOYEE_STATUS_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines the employment status of an employee (for example, active, terminated, or inactive). This table is used to build the HCM formats. | |

***Table 37.19*** *EMPLOYEE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_TYPE | 103200 | CODE_LANGUAGE | | LANGUAGE_CD<br>EMPLOYEE_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table contains unique codes to indicate an employee's employment type. This table is used to build the HCM formats. | HCM formats catalog |

***Table 37.20*** *EMPLOYEE_UNION*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_UNION | 103300 | CODE_LANGUAGE | | UNION_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| UNION_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table contains codes for labor unions. This table is used to build the HCM formats. | |

*Table 37.21 ETHNICITY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ETHNICITY | 103400 | CODE_LANGUAGE | | LANGUAGE_CD<br>ETHNICITY_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ETHNICITY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table is used to define an individual's ethnic origin. This table is used to build the HCM formats. | HCM formats catalog |

*Table 37.22 EXEMPT_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXEMPT_STATUS | 103500 | CODE_LANGUAGE | | LANGUAGE_CD<br>EXEMPT_STATUS_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXEMPT_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines a position as "exempt" or "non-exempt." An exempt position is a management, administrative, or professional position in which an employee works without close supervision. Such employees are exempt from coverage under the Fair Labor Standards Act. This table is used to build the HCM formats. | HCM formats catalog |

***Table 37.23*** *FLSA_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| FLSA_STATUS | 103600 | CODE_LANGUAGE | | FLSA_STATUS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| FLSA_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines the Employee FLSA (Fair Labor Standards Act) status. The FLSA status might be the same as the exempt status. This table is used to build the HCM formats. | HCM formats catalog |

***Table 37.24*** *GENDER*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| GENDER | 103700 | CODE_LANGUAGE | | LANGUAGE_CD<br>GENDER_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| GENDER_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This table defines the codes for gender. This table is used to build the HCM formats. | HCM formats catalog |

*Table 37.25 INDUSTRY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| INDUSTRY | 104000 | CODE_LANGUAGE | | LANGUAGE_CD<br>INDUSTRY_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| INDUSTRY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.26 JOB_GROUP*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| JOB_GROUP | 104100 | EEO_CLASS | | JOB_GROUP_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| JOB_GROUP_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.27 JOB*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| JOB | 104200 | JOB_GROUP | | JOB_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| JOB_CD<br>VALID_FROM_DTTM | | | This table contains the various jobs that the company has. HCM uses this table for loading the JOBS table. | |

***Table 37.28*** *MARITAL_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| MARITAL_STATUS | 104300 | CODE_LANGUAGE | | LANGUAGE_CD<br>MARITAL_STATUS_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| MARITAL_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

***Table 37.29*** *MILITARY_EXPERIENCE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| MILITARY_EXPERIENCE_TYPE | 104400 | CODE_LANGUAGE | | LANGUAGE_CD<br>MILITARY_EXPERIENCE_TYPE_ CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| MILITARY_EXPERIENCE_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.30 MEASURE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| MEASURE | 104500 | CODE_LANGUAGE<br>SOURCE_SYSTEM (SOURCE_SYSTEM_CD) | | MEASURE_NM<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| MEASURE_NM<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>MEASURE_RK<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | yes | SAS_MEASURE | | |

*Table 37.31 ORG_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ORG_TYPE | 104600 | CODE_LANGUAGE | | LANGUAGE_CD<br>ORGANIZATION_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ORGANIZATION_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.32 PAY_LEVEL_STRUCTURE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PAY_LEVEL_STRUCTURE | 104700 | CODE_LANGUAGE | | LANGUAGE_CD<br>PAY_LEVEL_STRUCTURE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PAY_LEVEL_STRUCTURE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.33* PAY_LEVEL

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PAY_LEVEL | 104800 | CODE_LANGUAGE<br>TIME_FREQUENCY (TIME_FREQUENCY_CD)<br>PAY_LEVEL_STRUCTURE (PAY_LEVEL _STRUCTURE_CD)<br>CURRENCY (CURRENCY_ CD) | | PAY_LEVEL_STRUCTURE_CD<br>PAY_LEVEL_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PAY_LEVEL_CD<br>PAY_LEVEL_STRUCTURE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.34* POSITION_PERMANENCE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| POSITION_PERMANENCE | 104900 | CODE_LANGUAGE | | LANGUAGE_CD<br>PERMANENCE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PERMANENCE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.35 POSITION_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| POSITION_STATUS | 105000 | CODE_LANGUAGE | | LANGUAGE_CD<br>POSITION_STATUS_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| POSITION_STATUS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.36 RECRUITMENT_SOURCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| RECRUITMENT_SOURCE | 105100 | CODE_LANGUAGE | | RECRUITMENT_SOURCE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| RECRUITMENT_SOURCE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

***Table 37.37*** *REJECTION_REASON*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| REJECTION_REASON | 105200 | CODE_LANGUAGE | | LANGUAGE_CD<br>REJECTION_REASON_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| REJECTION_REASON_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

***Table 37.38*** *STATE_REGION*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| STATE_REGION | 105400 | CODE_LANGUAGE | | LANGUAGE_CD<br>STATE_REGION_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| STATE_REGION_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

***Table 37.39*** *TIME_UNIT_OF_MEASURE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TIME_UNIT_OF_MEASURE | 105600 | CODE_LANGUAGE | | LANGUAGE_CD<br>TIME_UOM_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TIME_UOM_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.40 ITEM_CATEGORY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ITEM_CATEGORY | 105700 | CODE_LANGUAGE | | LANGUAGE_CD<br>ITEM_CATEGORY_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ITEM_CATEGORY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.41 ITEM_CATEGORY_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ITEM_CATEGORY_ASSOC_TYPE | 105710 | CODE_LANGUAGE | | LANGUAGE_CD<br>ITEM_CATEGORY_ASSOC_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ITEM_CATEGORY_ASSOC_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.42 ITEM_CATEGORY_ ASSOC*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ITEM_CATEGORY _ ASSOC | 105720 | ITEM_CATE GORY_ASSOC_TY PE (ITEM_CATE GORY_ASSOC_TY PE_ CD)<br>ITEM_CATE GORY (ITEM_CATE GORY_CD)<br>ITEM_CATE GORY (PARENT_IT EM_CATEGO RY_CD = ITEM_CATE GORY_CD) | | ITEM_CATEGORY _CD<br>PARENT_ITEM_CA TEGORY_CD<br>ITEM_CATEGORY _ASSOC_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ITEM_CATEGORY _CD<br>PARENT_ITEM_CA TEGORY _CD<br>ITEM_CATEGORY _ASSOC_ TYPE_CD<br>VALID_FROM_DT TM | | | | |

*Table 37.43 TIME_PERIOD*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TIME_PERIOD | 105800 | SOURCE_SYSTEM (SOURCE_SYSTEM _CD)<br>PERIOD_TYPE (PERIOD_TYPE_CD ) | | TIME_PERIOD_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TIME_PERIOD_ID<br>VALID_FROM_DTTM (Index)<br>TIME_PERIOD_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.44 TIME_PERIOD_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TIME_PERIOD_ASSOC_TYPE | 105810 | CODE_LANGUAGE | TIME_PERIOD (DEFAULT_TIME_PERIOD_ID = DEFAULT_TIME_PERIOD_RK) | LANGUAGE_CD<br>TIME_PERIOD_ASSOC_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TIME_PERIOD_ASSOC_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | |

*Table 37.45 TIME_PERIOD_ASSOC*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TIME_PERIOD_ASSOC | 105820 | TIME_PERIOD_ASSOC_TY PE (TIME_PERIOD_ASSOC_TYPE_CD) | TIME_PERIOD (TIME_PERIOD_ID = TIME_PERIOD_RK)<br>TIME_PERIOD (PARENT_TIME_PERIOD_ID = PARENT_TIME_PERIOD_RK) | TIME_PERIOD_RK<br>PARENT_TIME_PERIOD_RK<br>TIME_PERIOD_ASSOC_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TIME_PERIOD_ID<br>PARENT_TIME_PERIOD_ID<br>TIME_PERIOD_ASSOC_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>TIME_PERIOD_RK<br>PARENT_TIME_PERIOD_RK<br>TIME_PERIOD_ASSOC_TYPE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.46 EMPLOYEE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE | 106000 | COUNTY (COUNTY_NM)<br>ETHNICITY (ETHNICITY_ CD)<br>COUNTRY (COUNTRY_CD)<br>GENDER (GENDER_CD)<br>CODE_LANGUAGE (PRIMARY_LANGUAGE_CD = LANGUAGE_CD)<br>STATE_REGION (STATE_REGION_CD)<br>MARITAL_STATUS (MARITAL_STATUS_CD)<br>COUNTRY (CITIZENSHIP_COUNTRY_CD = COUNTRY_CD)<br>POSITION_PERMANENCE (PERMANENCE_CD) | | EMPLOYEE_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>VALID_FROM_DTTM (Index)<br>EMPLOYEE_RK<br>VALID_FROM_DTTM (Prim_Key) | | | This table contains current and historical information about employees. HCM uses this table to load the WRKGRP, EMPGEN, ABSHIST, ACTHIST, APPHIST, COMPHIST, and POS tables. | |

*Table 37.47* *COST_CENTER*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COST_CENTER | 106100 | SOURCE_SY STEM (SOURCE_SYSTEM_CD) | EMPLOYEE (RESPONSIBLE_EMPLOYEE_I D = RESPONSIBLE_EMPLOYEE_RK ) | COST_CENTER_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COST_CENTER_ID<br>VALID_FROM_DTTM (Index)<br>COST_CENTER_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.48* *COST_CENTER_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COST_CENTER_ASSOC_TYPE | 106110 | CODE_LANGUAGE | COST_CENTER (DEFAULT_COST_CENTER_ID = DEFAULT_COST_CENTER_RK) | LANGUAGE_CD<br>COST_CENTER_ASSOC_TYPE_ CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COST_CENTER_ASSOC_TY PE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>COST_CENTER_ASSOC_TY PE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.49 COST_CENTER_ASSOC*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COST_CENTER_ASSOC | 106120 | COST_CENTER_ASSOC_T YPE (COST_CENTER_ASSOC_TYPE_CD ) | COST_CENTER ( COST_CENTER_ID = COST_CENTER_RK)<br>COST_CENTER (PARENT_COST_CENTER_ID = PARENT_COST_CENTER_RK) | COST_CENTER_RK<br>PARENT_COST_CENTER_RK<br>COST_CENTER_ASSOC_TYPE_ CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COST_CENTER_ID<br>PARENT_COST_CENTER_I D<br>COST_CENTER_ASSOC_TY PE_CD<br>VALID_FROM_DTTM (Index)<br>COST_CENTER_RK<br>PARENT_COST_CENTER_R K<br>COST_CENTER_ASSOC_TY PE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.50 COST_CENTER_NLS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COST_CENTER_NLS | 106130 | CODE_LANGUAGE | COST_CENTER (COST_CENTER_ID = COST_CENTER_RK) | COST_CENTER_RK<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COST_CENTER_ID<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Index)<br>COST_CENTER_RK<br>LANGUAGE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.51 INTERNAL_ORG*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| INTERNAL_ORG | 106200 | CURRENCY (BOOK_OF_RECORD_CURRENCY_ CD)<br>COUNTRY (COUNTRY_C D)<br>COUNTY (COUNTY_N M)<br>STATE_REGI ON (STATE_REGION_CD)<br>ORG_TYPE (ORGANIZATION_TYPE_C D)<br>CURRENCY (REPORTING_CURRENCY _CD) | EMPLOYEE (MANAGING_EMPLOYEE_ID = MANAGING_EMPLOYEE_RK)<br>COST_CENTER (COST_CENTER_ID = COST_CENTER_RK) | INTERNAL_ORG_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| INTERNAL_ORG_ID<br>VALID_FROM_DTTM (Index)<br>INTERNAL_ORG_RK<br>VALID_FROM_DTTM (Prim_Key) | | | This table defines organizations that are part of your enterprise. Some examples are departments, divisions, and subsidiaries. Compare EXTERNAL_ORG. HCM uses this table to load the WRKGRP, ACTHIST, and POS tables. | HCM EDUHIST Table |

***Table 37.52*** *INTERNAL_ORG_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| INTERNAL_ORG_ASSOC_TYPE | 106210 | CODE_LANGUAGE | INTERNAL_OR G (DEFAULT_INTERNAL_ORG_I D = DEFAULT_INTERNAL_ORG_R K) | LANGUAGE_CD<br>INTERNAL_ORG_ASSOC_TYPE _CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| INTERNAL_ORG_ASSOC_T YPE_C D<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | This Association Type table is used to list codes that identify the types of hierarchies represented in the association table. | |

*Table 37.53 INTERNAL_ORG_ASSOC*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| INTERNAL_ORG_A SSOC | 106220 | INTERNAL_ ORG_ASSOC_ TYPE (INTERNAL_ ORG_ASSOC_ TYPE_C D) | INTERNAL_OR G (INTERNAL_OR G_ID = INTERNAL_OR G_RK)<br>INTERNAL_OR G (PARENT_INTE RNAL_ORG_ID = PARENT_INTE RNAL_ORG_RK) | INTERNAL_ORG_ RK<br>PARENT_INTERN AL_ORG_RK<br>INTERNAL_ORG_ ASSOC_TYPE _CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| INTERNAL_ORG_I D<br>PARENT_INTERNA L_ORG_ ID<br>INTERNAL_ORG_A SSOC_T YPE_C D<br>VALID_FROM_DT TM (Index)<br>INTERNAL_ORG_R K<br>PARENT_INTERNA L_ORG_ RK<br>INTERNAL_ORG_A SSOC_T YPE_C D<br>VALID_FROM_DT TM (Prim_Key) | | | This Association table is used to list one or more hierarchical parent and child relationships for the Internal Organization member IDs. The INTERNAL_ORG_A SSOC_TYPE _CD is used to distinguish each unique hierarchy type. HCM uses this table to load the WRKGRP table. | |

***Table 37.54*** *JOB_POSITION*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| JOB_POSITION | 106300 | PAY_LEVEL (PAY_LEVEL _CD)<br>POSITION_S TATUS (POSITION_S TATUS_CD)<br>JOB (JOB_CD)<br>FLSA_STATU S (FLSA_STAT US_CD)<br>POSITION_P ERMANENC E (PERMANEN CE_CD)<br>PAY_LEVEL_ STRUCTURE (PAY_LEVEL_STR UCTUR E)<br>EXEMPT_ST ATUS (EXEMPT_ST ATUS_CD)<br>TIME_FREQ UENCY (TIME_FREQ UENCY_CD)<br>CURRENCY (CURRENCY_ CD) | INTERNAL_OR G (INTERNAL_OR G_ID = INTERNAL_OR G_RK)<br>COST_CENTER (ALLOCATED_T O_COST_CENT ER_ID = ALLOCATED_T O_COST_CENT ER_RK) | POSITION_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| POSITION_CD<br>VALID_FROM_DT TM | | | | |

***Table 37.55*** *EXTERNAL_ORG*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG | 106700 | INDUSTRY (INDUSTRY_ CD)<br>ORG_TYPE (ORGANIZAT ION_TYPE_C D)<br>SOURCE_SY STEM (SOURCE_SY STEM_CD) | | EXTERNAL_ORG_ ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXTERNAL_ORG_I D<br>VALID_FROM_DT TM (Index)<br>EXTERNAL_ORG_ RK<br>VALID_FROM_DT TM (Prim_Key) | | | | HCM EDUHIST Table |

***Table 37.56*** *EXTERNAL_ORG_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG_ ASSOC_TYPE | 106710 | CODE_LANG UAGE | EXTERNAL_OR G (DEFAULT_EXT ERNAL_ORG_I D = DEFAULT_EXT ERNAL_ORG_R K) | LANGUAGE_CD<br>EXTERNAL_ORG_ ASSOC_TYPE _CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXTERNAL_ORG_ ASSOC_T YPE_C D<br>LANGUAGE_CD<br>VALID_FROM_DT TM | | | | |

*Table 37.57* EXTERNAL_ORG_ASSOC

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG_ ASSOC | 106720 | EXTERNAL_ ORG_ASSOC_ TYPE (EXTERNAL_ ORG_ASSOC_ TYPE_C D) | EXTERNAL_OR G (EXTERNAL_OR G_ID = EXTERNAL_OR G_RK)<br>EXTERNAL_OR G (PARENT_EXTE RNAL_ORG_ID = PARENT_EXTE RNAL_ORG_RK) | EXTERNAL_ORG_ RK<br>PARENT_EXTERN AL_ORG_RK<br>EXTERNAL_ORG_ ASSOC_TYPE _CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXTERNAL_ORG_I D<br>PARENT_EXTERN AL_ORG_ ID<br>EXTERNAL_ORG_ ASSOC_T YPE_C D<br>VALID_FROM_DT TM (Index)<br>EXTERNAL_ORG_ RK<br>PARENT_EXTERN AL_ORG_ RK<br>EXTERNAL_ORG_ ASSOC_T YPE_C D<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.58* EXTERNAL_ORG_NLS

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG_ N LS | 106730 | CODE_LANG UAGE | EXTERNAL_OR G (EXTERNAL_OR G_ID = EXTERNAL_OR G_RK) | EXTERNAL_ORG_ RK<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXTERNAL_ORG_I D<br>LANGUAGE_CD<br>VALID_FROM_DT TM (Index)<br>EXTERNAL_ORG_ RK<br>LANGUAGE_CD<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.59 EXTERNAL_ORG_ADDRESS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG_ ADDRESS | 106740 | COUNTRY (COUNTRY_C D)<br>ADDRESS_T YPE (ADDRESS_T YPE_CD)<br>STATE_REGI ON (STATE_REG ION_CD) | EXTERNAL_OR G (EXTERNAL_OR G_ID = EXTERNAL_OR G_RK) | EXTERNAL_ORG_ RK<br>ADDRESS_TYPE_ CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EXTERNAL_ORG_I D<br>ADDRESS_TYPE_C D<br>VALID_FROM_DT TM (Index)<br>EXTERNAL_ORG_ RK<br>ADDRESS_TYPE_C D<br>VALID_FROM_DT TM (Prim_Key) | | | | HCM EDUHIST Table |

***Table 37.60*** *EXTERNAL_ORG_CONTACT*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EXTERNAL_ORG_ CONTACT | 106800 | CONTACT_R OLE (CONTACT_R OLE_CD) | EXTERNAL_OR G (EXTERNAL_OR G_ID = EXTERNAL_OR G_RK) | CONTACT_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| CONTACT_ID<br>VALID_FROM_DT TM (Index)<br>CONTACT_RK<br>VALID_FROM_DT TM (Prim_Key) | | | | |

***Table 37.61*** *PROFIT_CENTER*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PROFIT_CENTER | 106900 | SOURCE_SY STEM | EMPLOYEE (RESPONSIBLE _EMPLOYEE_I D = RESPONSIBLE_ EMPLOYEE_RK ) | PROFIT_CENTER_ ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PROFIT_CENTER_I D<br>VALID_FROM_DT TM (Index)<br>PROFIT_CENTER_ RK<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.62* PROFIT_CENTER_ASSOC

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PROFIT_CENTER_ ASSOC | 106910 | PROFIT_CEN TER_ASSOC_ TYPE (PROFIT_CE NTER_ASSO C_TYPE_C D) | PROFIT_CENTE R (PARENT_PROF IT_CENTER_ID = PARENT_PROFI T_CENTER_RK)<br>PROFIT_CENTE R (PROFIT_CENT ER_ID = PROFIT_CENTE R_RK) | PROFIT_CENTER_ RK<br>PARENT_PROFIT_ CENTER_RK<br>PROFIT_CENTER_ ASSOC_TYP E_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PROFIT_CENTER_I D<br>PARENT_PROFIT_ CENTER _ID<br>PROFIT_CENTER_ ASSOC_ TYPE_C D<br>VALID_FROM_DT TM (Index)<br>PROFIT_CENTER_ RK<br>PARENT_PROFIT_ CENTER _RK<br>PROFIT_CENTER_ ASSOC_ TYPE_C D<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.63* PROFIT_CENTER_ASSOC_TYPE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PROFIT_CENTER_ ASSOC_TYPE | 106920 | CODE_LANG UAGE | PROFIT_CENTE R (DEFAULT_PRO FIT_CENTER_I D TO DEFAULT_PRO FIT_CENTER_R K) | LANGUAGE_CD<br>PROFIT_CENTER_ ASSOC_TYP E_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PROFIT_CENTER_ ASSOC_ TYPE_C D<br>LANGUAGE_CD<br>VALID_FROM_DT TM | | | | |

*Table 37.64* PROFIT_CENTER_NLS

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| PROFIT_CENTER_ NLS | 106930 | CODE_LANG UAGE | PROFIT_CENTE R (PROFIT_CENT ER_ID = PROFIT_CENTE R_RK) | PROFIT_CENTER_ RK<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| PROFIT_CENTER_I D<br>LANGUAGE_CD<br>VALID_FROM_DT TM (Index)<br>PROFIT_CENTER_ RK<br>LANGUAGE_CD<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.65* COMPENSATION

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPENSATION | 107200 | CURRENCY<br>COMPENSAT ION_TYPE | EMPLOYEE (EMPLOYEE_ID - EMPLOYEE_RK ) | |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>COMPENSATION_ TYPE_C D<br>PAYMENT_DT (Index)<br>EMPLOYEE_RK<br>COMPENSATION_ TYPE_C D<br>PAYMENT_DT (Prim_Key) | | | This table contains a history of additional compensation (other than base pay) paid to an employee (for example, bonuses or one-time awards). HCM uses this table to load the COMPHIST table. | HCM formats catalog |

*Table 37.66 EMPLOYEE_X_INTERNAL_ORG*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_X_INT ERNAL_ORG | 107300 | INTERNAL_ ORG_ASSOC_ TYPE (INTERNAL_ ORG_ASSOC_ TYPE_C D) | INTERNAL_OR G (INTERNAL_OR G_ID = INTERNAL_OR G_RK)<br>EMPLOYEE (EMPLOYEE_ID = EMPLOYEE_RK ) | EMPLOYEE_RK<br>INTERNAL_ORG_ ASSOC_TYPE _CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| INTERNAL_ORG_I D<br>EMPLOYEE_ID<br>INTERNAL_ORG_A SSOC_T YPE_C D<br>VALID_FROM_DT TM (Index)<br>INTERNAL_ORG_R K<br>EMPLOYEE_RK<br>INTERNAL_ORG_A SSOC_T YPE_C D<br>VALID_FROM_DT TM (Prim_Key) | | | This table identifies the employees associated with an internal organization and vice versa. HCM uses this table to load the ACTHIST table. | |

*Table 37.67* EMPLOYEE_X_JOB

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_X_JOB | 107400 | JOB (JOB_CD)<br>JOB_POSITI ON (POSITION_C D) | EMPLOYEE (EMPLOYEE_ID - EMPLOYEE_RK ) | N/A |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>JOB_CD<br>START_DT (Index)<br>EMPLOYEE_RK<br>JOB_CD<br>START_DT (Prim_Key) | | | This table associates an employee with a position or job. HCM uses this table to load the ACTHIST and POS tables. | |

*Table 37.68* EMPLOYEE_ABSENCE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_ABSE NCE | 107500 | ABSENCE_T YPE (ABSENCE_T YPE_CD)<br>TIME_UNIT_ OF_MEASUR E (DURATION_ TIME_UOM_ CD look-up TIME_UOM_ CD) | EMPLOYEE (EMPLOYEE_ID - EMPLOYEE_RK ) | N/A |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ABSENCE_ID | | | This table defines and tracks the information about an employee absence. HCM uses this table to load the ABSHIST table. | |

*Table 37.69 EMPLOYEE_ACTION*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_ACTION | 107600 | PAY_LEVEL (PAY_LEVEL _CD)<br>CURRENCY (CURRENCY_ CD)<br>ACTION_REA SON (EMPLOYEE_ ACTION_REA SON_CD)<br>EXEMPT_ST ATUS (EXEMPT_ST ATUS_CD)<br>EMPLOYEE_ STATUS (EMPLOYEE_ STATUS_CD)<br>EMPLOYEE_ TYPE (EMPLOYEE_ TYPE_CD)<br>ACTION_TYP E (ACTION_TY PE_CD)<br>PAY_LEVEL_ STRUCTURE (PAY_LEVEL _STRUCTUR E_CD)<br>TIME_FREQ UENCY (PAY_FREQU ENCY_CD)<br>TIME_FREQ UENCY (EMPLOYEE_ PAID_FREQU ENCY_C D) | N/A | N/A |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ACTION_DT<br>EMPLOYEE_ID<br>ACTION_TYPE_CD (Index)<br>ACTION_DT<br>EMPLOYEE_RK<br>ACTION_TYPE_CD (Prim_Key) | | | This table contains particular employee actions, such as pay increases, suspensions, or probations. This table represents valid combinations of Action Type and Reason Type. HCM uses this table for loading the ACTHIST table. | HCM Acthist table |

***Table 37.70*** *EMPLOYMENT_APPLICATION*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYMENT_AP PLICATION | 107700 | ETHNICITY<br>APPLICATIO N_STATUS<br>SOURCE_SY STEM<br>RECRUITME NT_SOURCE<br>JOB_POSITI ON<br>COUNTY<br>GENDER<br>STATE_REGI ON<br>EMPLOYEE<br>REJECTION_ REASON<br>SAS_COUNT RY_ISO3166 (COUNTRY _CD = COUNT RY_CD_3DIG IT) | EMPLOYEE (EMPLOYEE_ID - EMPLOYEE_RK ) - Null values accepted | N/A |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYMENT_AP PLICATION_ID | | | This table contains employment application data. HCM uses this table for loading the APPHIST table. | HCM Apphist table |

*Table 37.71 MILITARY_EXPERIENCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| MILITARY_EXPERIENCE | 107800 | MILITARY_EXPERIENCE_ TYPE | EMPLOYEE (EMPLOYEE_ID - EMPLOYEE_RK ) | N/A |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>MILITARY_EXPERIENCE_T YPE_CD (Index)<br>EMPLOYEE_RK<br>MILITARY_EXPERIENCE_T YPE_CD (Prim_Key) | | | | |

*Table 37.72 ACADEMIC_CREDIT*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ACADEMIC_CREDIT | 201010 | CODE_LANGUAGE | | ACADEMIC_CREDIT_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ACADEMIC_CREDIT_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.73 ACADEMIC_CREDIT*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ACADEMIC_CREDIT | 201010 | CODE_LANGUAGE | | ACADEMIC_CREDIT_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ACADEMIC_CREDIT_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.74 ACADEMIC_HONORS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ACADEMIC_HONORS | 201020 | CODE_LANGUAGE | | ACADEMIC_HONORS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ACADEMIC_HONORS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.75 ATTENDANCE_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ATTENDANCE_STATUS | 201030 | CODE_LANGUAGE | | ATTENDANCE_STATUS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ATTENDANCE_STATUS_C D<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.76* COURSE_LEVEL

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COURSE_LEVEL | 201040 | CODE_LANGUAGE | | COURSE_LEVEL_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COURSE_LEVEL_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.77* DEGREE_CONCENTRATION

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| DEGREE_CONCENTRATION | 201050 | CODE_LANGUAGE | | DEGREE_CONCENTRATION_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| DEGREE_CONCENTRATIO N_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.78* DEGREE_OPTION

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| DEGREE_OPTION | 201060 | CODE_LANGUAGE | | DEGREE_OPTION_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| DEGREE_OPTION_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.79 DEGREE_PROGRAM*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| DEGREE_PROGRAM | 201070 | CODE_LANGUAGE | | DEGREE_PROGRAM_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| DEGREE_PROGRAM_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.80 DEGREE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| DEGREE_TYPE | 201080 | CODE_LANGUAGE | | DEGREE_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| DEGREE_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.81*** *EDUCATION_VALUE_SYSTEM*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EDUCATION_VALUE_SYSTEM | 201090 | CODE_LANGUAGE | | EDUCATION_VALUE_SYSTEM_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_VALUE_SYST EM_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.82*** *EDUCATION_VALUE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EDUCATION_VALUE_TYPE | 201100 | CODE_LANGUAGE | | EDUCATION_VALUE_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_VALUE_TYPE _CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.83*** *ENROLLMENT_STATUS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| ENROLLMENT_STATUS | 201110 | CODE_LANGUAGE | | ENROLLMENT_STATUS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| ENROLLMENT_STATUS_C D<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.84 FICE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| FICE | 201120 | CODE_LANGUAGE | | FICE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| FICE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.85 GRADUATING_DEGREE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| GRADUATING_DEGREE | 201130 | CODE_LANGUAGE | | GRADUATING_DEGREE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| GRADUATING_DEGREE_C D<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.86*** *HONORS_PROGRAM*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| HONORS_PROGRAM | 201140 | CODE_LANGUAGE | | HONORS_PROGRAM_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| HONORS_PROGRAM_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.87*** *OTHER_HONORS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| OTHER_HONORS | 201150 | CODE_LANGUAGE | | OTHER_HONORS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| OTHER_HONORS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.88*** *SCHOOL_DEPT_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SCHOOL_DEPT_TYPE | 201160 | CODE_LANGUAGE | | SCHOOL_DEPT_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHOOL_DEPT_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.89* SCHOOL_TYPE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SCHOOL_TYPE | 201170 | CODE_LANGUAGE | | SCHOOL_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHOOL_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.90* SCHOOL_NAME_TYPE

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SCHOOL_NAME_TYPE | 201180 | CODE_LANGUAGE | | SCHOOL_NAME_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHOOL_NAME_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.91* SCHOOL_OR_INSTITUTION

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SCHOOL_OR_INSTITUTION | 201190 | SCHOOL_TY PE (SCHOOL_TYPE_CD)<br>SCHOOL_NAME_TYPE (SCHOOL_NAME_TYPE_C D)<br>FICE (FICE_CD) | INTERNAL_OR G (INTERNAL_ORG_ID=INTERNAL_ORG_RK)<br>EXTERNAL_OR G (EXTERNAL_ORG_ID=EXTERNAL_ORG_RK) | SCHOOL_OR_INSTITUTION_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHOOL_OR_INSTITUTIO N_ID<br>VALID_FROM_DTTM (Index)<br>SCHOOL_OR_INSTITUTIO N_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | HCM EDUHIST Table<br>HCM EDUVAL Table |

*Table 37.92* SCHOOL_DEPT

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SCHOOL_DEPT | 201200 | SCHOOL_DEPT_TYPE (SCHOOL_DEPT_TYPE_CD ) | EXTERNAL_OR G (EXTERNAL_ORG_ID=EXTERNAL_ORG_RK)<br>INTERNAL_OR G (INTERNAL_ORG_ID=INTERNAL_ORG_RK)<br>SCHOOL_OR_INSTITUTION (SCHOOL_OR_INSTITUTION_ID =SCHOOL_OR_INSTITUTION_R K) | SCHOOL_DEPT_I D |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SCHOOL_DEPT_ID<br>VALID_FROM_DTTM (Index)<br>SCHOOL_DEPT_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | HCM EDUHIST Table<br>HCM EDUVAL Table |

*Table 37.93* EDUCATION_HISTORY

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EDUCATION_HISTORY | 201210 | ACADEMIC_CREDIT (ACADEMIC_CREDIT_CD)<br>COURSE_LE VEL (COURSE_LEVEL_CD )<br>ACADEMIC_HONORS (ACADEMIC_HONORS_CD)<br>ATTENDANCE_STATUS (ATTENDANCE_STATUS_CD )<br>DEGREE_CONCENTRATI ON (DEGREE_CONCENTRATION_CD)<br>DEGREE_OP TION (DEGREE_OPTION_C D)<br>DEGREE_PROGRAM (DEGREE_PROGRAM_CD)<br>DEGREE_TY PE (DEGREE_TYPE_CD)<br>EDUCATION_VALUE_SYS TEM (EDUCATION_VALUE_SYSTEM_C D)<br>GRADUATING_DEGREE ( GRADUATING_DEGREE_ CD)<br>HONORS_PROGRAM (HONORS_PROGRAM_CD)<br>OTHER_HON ORS (OTHER_HONORS_CD ) | EMPLOYEE (EMPLOYEE_ID=EMPLOYEE_RK)<br>SCHOOL_DEPT ( SCHOOL_DEPT_ID=SCHOOL_DEPT_RK) | EDUCATION_HISTORY_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_HISTORY_ID<br>VALID_FROM_DTTM (Index)<br>EDUCATION_HISTORY_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | HCM EDUHIST Table<br>HCM EDUVAL Table |

***Table 37.94*** *EDUCATION_VALUE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EDUCATION_VALUE | 201220 | EDUCATION _VALUE_TYP E (EDUCATION_VALUE_TYPE_CD) | EDUCATION_HISTORY (EDUCATION_HISTORY _ID=EDUCATION_HISTORY_RK ) | EDUCATION_HISTORY_RK<br>EDUCATION_VALUE_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_HISTORY_ID<br>EDUCATION_VALUE_TYPE _CD<br>VALID_FROM_DTTM (Index)<br>EDUCATION_HISTORY_RK<br>EDUCATION_VALUE_TYPE _CD<br>VALID_FROM_DTTM (Prim_Key) | | | | HCM EDUVAL Table |

***Table 37.95*** *DATES_OF_ATTENDANCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| DATES_OF_ATTENDANCE | 201230 | ENROLLMENT_STATUS ( ENROLLMENT_STATUS_ CD) | EDUCATION_HISTORY (EDUCATION_HISTORY _ID=EDUCATION_HISTORY_RK ) | |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EDUCATION_HISTORY_ID<br>ATTENDANCE_START_DT (Index)<br>EDUCATION_HISTORY_RK<br>ATTENDANCE_START_DT (Prim_Key) | | | | HCM EDUHIST Table |

*Table 37.96* COMPETENCY_CATALOG

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_CATALOG | 211010 | CODE_LANGUAGE | | COMPETENCY_CATALOG_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_CATALOG_ CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.97* COMPETENCY_CLASS

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_CLASS | 211020 | CODE_LANGUAGE | | COMPETENCY_CLASS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_CLASS_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.98*** *COMPETENCY_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_TYPE | 211030 | CODE_LANGUAGE | | COMPETENCY_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.99*** *COMPETENCY_CATEGORY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_CATEGORY | 211040 | CODE_LANGUAGE | | COMPETENCY_CATEGORY_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_CATEGORY _CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.100*** *EVIDENCE_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EVIDENCE_TYPE | 211050 | CODE_LANGUAGE | | EVIDENCE_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EVIDENCE_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.101 SPECIAL_REF_COMP*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| SPECIAL_REF_COMP | 211060 | CODE_LANGUAGE | | SPECIAL_REF_COMP_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| SPECIAL_REF_COMP_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.102 TAXONOMY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TAXONOMY | 211070 | CODE_LANGUAGE | | TAXONOMY_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TAXONOMY_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.103*** *TAXONOMY_CATALOG*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TAXONOMY_CATALOG | 211080 | CODE_LANGUAGE | | TAXONOMY_CATALOG_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TAXONOMY_CATALOG_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.104*** *TAXONOMY_COMP_CLASS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TAXONOMY_COMP_CLASS | 211090 | CODE_LANGUAGE | | TAXONOMY_COMP_CLASS_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TAXONOMY_COMP_CLASS _CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.105*** *TAXONOMY_SPECIAL_REF*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TAXONOMY_SPECIAL_REF | 211100 | CODE_LANGUAGE | | TAXONOMY_SPECIAL_REF_COMP_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TAXONOMY_SPECIAL_REF_COMP_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.106 WEIGHT_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| WEIGHT_TYPE | 211110 | CODE_LANGUAGE | | WEIGHT_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| WEIGHT_TYPE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

*Table 37.107 COMPETENCY_CATEGORY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_CATEGORY | 211120 | COMPETENCY_CATEGOR Y (COMPETENCY_CATEGORY_CD)<br>COMPETENCY_CATALOG ( COMPETENCY_CATALOG_ CD)<br>COMPETENCY_CLASS (COMPETENCY_CLASS_CD)<br>JOB_GROUP ( COMPETENCY_JOB_GROUP_CD)<br>COMPETENCY_TYPE (COMPETENCY_TYPE_CD)<br>SPECIAL_REF_COMP (SPECIAL_REF_COMP_CD) | | COMPETENCY_CATEGORY_CLASS_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_CATEGORY_CLASS_CD<br>VALID_FROM_DTTM | | | | |

***Table 37.108*** *TAXONOMY_SOURCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| TAXONOMY_SOURCE | 211130 | TAXONOMY ( TAXONOMY_CD)<br>TAXONOMY_CATALOG (TAXONOMY_CATALOG_CD)<br>TAXONOMY_COMP_CLAS S (TAXONOMY_COMP_CLASS_CD) | | TAXONOMY_SOURCE_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| TAXONOMY_SOURCE_ID<br>VALID_FROM_DTTM (Index)<br>TAXONOMY_SOURCE_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | |

***Table 37.109*** *COMPETENCY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY | 211140 | COMPETENCY_CATEGORY_CLASS (COMPETENCY_CATEGORY_CLASS_CD) | | COMPETENCY_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ID<br>VALID_FROM_DTTM (Index)<br>COMPETENCY_RK<br>VALID_FROM_DTTM (Prim_ Key) | | | | |

***Table 37.110** COMPETENCY_ASSOC_TYPE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_ASSOC_TYPE | 211150 | CODE_LANGUAGE<br>COMPETENC Y (DEFAULT_ COMPETENCY_ID) | COMPETENCY ( DEFAULT_COMPETENCY_ID=DEFAULT_COMPETENCY_RK) | COMPETENCY_ASSOC_TYPE_CD<br>LANGUAGE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ASSOC_TY PE_CD<br>LANGUAGE_CD<br>VALID_FROM_DTTM | | | | HCM formats catalog |

***Table 37.111** COMPETENCY_ASSOC*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_ASSOC | 211160 | COMPETENCY_ASSOC_TY PE (COMPETENCY_ASSOC_TYPE_CD) | COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK)<br>COMPETENCY ( PARENT_COMPETENCY_ID=PARENT_COMPETENCY_RK) | COMPETENCY_RK<br>PARENT_COMPETENCY_RK<br>COMPETENCY_ASSOC_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ID<br>PARENT_COMPETENCY_I D<br>COMPETENCY_ASSOC_TY PE_CD<br>VALID_FROM_DTTM (Index)<br>COMPETENCY_RK<br>PARENT_COMPETENCY_R K<br>COMPETENCY_ASSOC_TY PE_CD<br>VALID_FROM_DTTM (Prim_ Key) | | | | |

*Table 37.112* COMPETENCY_X_TAXONOMY

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_X_TAXONOMY | 211170 | N/A | COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK)<br>TAXONOMY_SOURCE (TAXONOMY_SOURCE_ID=TAXONOMY_SOURCE_RK) | COMPETENCY_ID<br>TAXONOMY_SOURCE_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ID<br>TAXONOMY_SOURCE_ID<br>VALID_FROM_DTTM (Index)<br>COMPETENCY_RK<br>TAXONOMY_SOURCE_RK<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.113 COMPETENCY_WEIGHT_BOUNDS*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_WEIGHT_BOUNDS | 211180 | WEIGHT_TY PE (WEIGHT_TYPE_CD) | COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK) | COMPETENCY_ID<br>WEIGHT_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ID<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>COMPETENCY_RK<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.114 COMPETENCY_EVIDENCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| COMPETENCY_EVIDENCE | 211190 | EVIDENCE_T YPE (EVIDENCE_TYPE_CD ) | COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK) | COMPETENCY_ID<br>EVIDENCE_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| COMPETENCY_ID<br>EVIDENCE_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>COMPETENCY_RK<br>EVIDENCE_TYPE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.115 EMPLOYEE_X_COMPETENCY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_X_CO MPETENCY | 211200 | EMPLOYEE ( EMPLOYEE_ ID)<br>COMPETENC Y (COMPETE NCY_ID) | COMPETENCY ( COMPETENCY_ ID=COMPETEN CY_RK)<br>EMPLOYEE (EM PLOYEE_ID=E MPLOYEE_RK) | EMPLOYEE_ID<br>COMPETENCY_ID |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>COMPETENCY_ID<br>VALID_FROM_DT TM (Index)<br>EMPLOYEE_RK<br>COMPETENCY_RK<br>VALID_FROM_DT TM (Prim_Key) | | | | |

*Table 37.116 EMPLOYEE_X_EVIDENCE*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_X_EVI DENCE | 211210 | EVIDENCE_T YPE (EVIDEN CE_TYPE_CD ) | EMPLOYEE (EM PLOYEE_ID=E MPLOYEE_RK)<br>COMPETENCY ( COMPETENCY_ ID=COMPETEN CY_RK) | EMPLOYEE_ID<br>COMPETENCY_ID<br>EVIDENCE_TYPE_ CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>COMPETENCY_ID<br>EVIDENCE_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>EMPLOYEE_RK<br>COMPETENCY_RK<br>EVIDENCE_TYPE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

*Table 37.117* *JOB_X_COMPETENCY*

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| JOB_X_COMPETENCY | 211220 | JOB (JOB_CD) | COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK) | JOB_CD<br>COMPETENCY_ID<br>WEIGHT_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| JOB_CD<br>COMPETENCY_ID<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>JOB_CD<br>COMPETENCY_RK<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Prim_ Key) | | | | |

*Table 37.118* EMPLOYEE_X_JOB_X_COMPETENCY

| Table Name | Job Reference Number | Table Validation Dependencies | Lookups | SCD2 Business Key |
|---|---|---|---|---|
| EMPLOYEE_X_JOB_X_COMPETENCY | 211230 | JOB (JOB_CD)<br>JOB_POSITION (POSITIO N_CD) | EMPLOYEE (EMPLOYEE_ID=EMPLOYEE_RK)<br>COMPETENCY ( COMPETENCY_ID=COMPETENCY_RK) | EMPLOYEE_ID<br>JOB_CD<br>START_DT<br>COMPETENCY_ID<br>WEIGHT_TYPE_CD |

| Business Index & Primary Key | SAS Reference Table? | SAS Reference Table Name | Data Content Comments | HCM Table Relationships |
|---|---|---|---|---|
| EMPLOYEE_ID<br>JOB_CD<br>START_DT<br>COMPETENCY_ID<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Index)<br>EMPLOYEE_RK<br>JOB_CD<br>START_DT<br>COMPETENCY_RK<br>WEIGHT_TYPE_CD<br>VALID_FROM_DTTM (Prim_Key) | | | | |

# Chapter 38
# Solutions Data Mart Jobs

Solutions Data Mart Jobs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 571

## Solutions Data Mart Jobs

The following table contains descriptions of the jobs that load data into the Solutions Data Mart.

These jobs are discussed in the Data Administration part of this book.

***Table 38.1*** *Solutions Data Mart Jobs*

| Site Checkoff | Job Name | Description |
|---|---|---|
| | solnsvc_1100_load_source_system | Loads source system codes from the SOURCE_SYSTEM table of the DDS into the SAS_SOURCE_SYSTEM and SAS_SOURCE_SYSTEM_NLS tables of the SDM. Under normal circumstances, you do not need to run this job. All predefined source system codes are placed in the SDM by the installation procedure. |
| | solnsvc_1200_import_locales | Loads language and country combinations from the CODE_LANGUAGE table of the DDS into the SAS_DATA_LOCALE table of the SDM. |
| | solnsvc_1300_load_users | Loads users from the metadata server into the SAS_USER table of the SDM. |
| | solnsvc_1400_load_groups | Loads groups from the metadata server into the SAS_GROUP table of the SDM. |
| | solnsvc_1500_load_user_x_group | Loads user and group relationships from the metadata server into the SAS_USER_X_GROUP table of the SDM. |
| | solnsvc_2000_load_dimension_types | Loads dimension types. |

| Site Checkoff | Job Name | Description |
|---|---|---|
| | solnsvc_2100_create_application_dimension | Loads dimension records from the APP_DIMENSION table of the DDS into the SAS_DIMENSION table of the SDM. |
| | solnsvc_2200_create_dimension | Creates a dimension record in the SAS_DIMENSION table of the SDM, based on option values that you set in the job. |
| | solnsvc_3200_load_dimension | Loads member and hierarchy records for a specified dimension from the DDS into the appropriate tables of the SDM.<br>*Note:* You must load currencies for a currency dimension before you load organizations for an intorg dimension, due to the dependency of organizations on currencies. |
| | solnsvc_3300_load_measure_table | Loads measures from the SAS_MEASURE source table and possibly other sources into the SAS_MEASURE table of the SDM. |
| | solnsvc_3400_load_metric_table | Creates or updates a metric table in the SDM, based on option values that you set in the job. |
| | solnsvc_4100_export_dimension | Exports member and hierarchy records for a specified dimension from the SDM to the StageDDS library or another specified library. |

# *Chapter 39*
# HCM Data Mart Jobs

**HCM Data Mart Jobs** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 573

## HCM Data Mart Jobs

The following tables contain descriptions of the jobs that load data into the HCM Data Mart. The jobs are listed in the order in which the hcm_100000_load_datamart_all_tables umbrella job runs them. This is also the order in which you should run the jobs if you run them individually. Jobs identified as optional are not included in the umbrella job.

Many of the jobs call Prebuild.sas (Precode) and Updatedd.sas (Postcode). Run jobs only if data is available to load their target tables or cubes. The individual cube jobs can be run in any order, after the source tables are built or refreshed.

For more information about the following jobs, see the Data Administration part of this book.

***Table 39.1*** *hcm_110050_load_formats_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS_HCM_FORMATS | hcm_110050_load_formats_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| DDS Reference Tables<br>cind_dds.Country<br>sas_sup.Sas_country_iso3166 | Formats.sas | | Loads and updates HCM Formats |

***Table 39.2*** *hcm_110100_load_formats_table_from_catalog (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS_HCM_FORMATS | hcm_110100_load_formats_table_from_catalog |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| | Fmttupd.sas | | This is an optional job to load new formats into SAS Human Capital Management from a formats catalog. Edit the job to specify the path to a formats catalog before running this job. |

***Table 39.3*** *hcm_110150_load_wrkgrp_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Work Group | hcm_110150_load_wrkgrp_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Internal_org_assoc<br>cind_dds.Internal_org<br>cind_dds.Internal_org_nls<br>cind_dds.Employee<br>Formats | Wrkgrp.sas | | |

***Table 39.4*** *hcm_110200_load_abshist_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Absence History | hcm_110200_load_abshist_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Employee_Absence<br>cind_dds.Employee | Mapped | Employee<br>(Employee_id) | |

***Table 39.5*** *hcm_110250_load_acthist_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Job Action History | hcm_110250_load_acthist_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Employee_Action<br>cind_dds.Employee<br>cind_dds.Employee_x_job<br>cind_dds.Job_Position<br>cind_dds.employee_x_internal_o rg<br>cind_dds.Internal_org_x<br>cind_dds.Employee_x<br>Formats | Acthist.sas | | |

***Table 39.6*** *hcm_110300_load_apphist_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Applicant History | hcm_110300_load_apphist_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Employment_Application<br>cind_dds.Employee | Mapped | Employee<br>(Employee_id) | |

***Table 39.7*** *hcm_110350_load_comphist_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Compensation History | hcm_110350_load_comphist_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Compensation<br>cind_dds.Compensation_type<br>cind_dds.Employee | Mapped | Employee<br>(Employee_id) | |

***Table 39.8*** *hcm_110400_load_empgen_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Employee General | hcm_110400_load_empgen_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Employee<br>cind_dds.Military_Experience | Mapped | | |

*Table 39.9 hcm_110450_load_useremployee_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS_USER_EMPLOYEE | hcm_110450_load_useremployee_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| HCMData.Empgen<br>HCMData.Acthist<br>HCM Group in OMR | Useremp.sas | | Loads Employee User Name information for SAS Human Capital Management. Employee User Name information needs to be loaded before a user can log on to SAS Human Capital Management. |

*Table 39.10 hcm_110500_load_grade_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Pay Grade | hcm_110500_load_grade_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Pay_Level | Mapped | | |

*Table 39.11 hcm_110550_load_jobs_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Jobs | hcm_110550_load_jobs_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Job_Group<br>cind_dds.Job | Mapped | | |

***Table 39.12*** *hcm_110600_load_openpos_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Open Position | hcm_110600_load_openpos_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Job_Position | Mapped | | |

***Table 39.13*** *hcm_110650_load_pos_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Position History | hcm_110650_load_pos_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Job_Position<br>cind_dds.Employee_x_Job<br>cind_dds.Employee<br>cind_dds.Internal_org | Mapped | Employee (Employee_id)<br>Internal_org (Internal_org_id) | |

***Table 39.14*** *hcm_110700_load_cgrade_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Current Pay Grade | hcm_110700_load_cgrade_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| HCMData.Grade | Mapped | | |

***Table 39.15*** *hcm_110750_load_cjobs_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Current Jobs | hcm_110750_load_cjobs_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| HCMData.Jobs | Mapped | | |

***Table 39.16*** *hcm_110800_load_cwrkgrp_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Current Work Group | hcm_110800_load_cwrkgrp_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| HCMData.Wrkgrp | Mapped | | |

***Table 39.17*** *hcm_118050_load_eduhist_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Education History | hcm_118050_load_eduhist_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.School_dept<br>cind_dds.School_or_institution<br>cind_dds.External_org<br>cind_dds.External_org_address<br>cind_dds.Education_history<br>cind_dds.Dates_of_attendance<br>cind_dds.Internal_org | Mapped | | |

***Table 39.18*** *hcm_118100_load_eduval_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Education Assessment | hcm_118100_load_eduval_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.school_dept<br>cind_dds.School_or_institution<br>cind_dds.Education_history<br>cind_dds.Education_value | Mapped | | |

***Table 39.19*** *hcm_120050_load_abshmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Absence History Master | hcm_120050_load_abshmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Absence History<br>Job Action History<br>Position History<br>Work Group<br>Jobs<br>Pay Grade<br>Current Employee General | Abshmast.sas | | |

***Table 39.20*** *hcm_120100_load_acthmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Job Action History Master | hcm_120100_load_acthmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Work Group<br>Jobs<br>Pay Grade<br>Current Employee General | Acthmast.sas | | |

***Table 39.21*** *hcm_120150_load_apphmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Applicant History Master | hcm_120150_load_apphmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Applicant History<br>Position History<br>Work Group<br>Jobs<br>Pay Grade<br>Current Employee General | Apphmast.sas | | |

***Table 39.22*** *hcm_120200_load_churn_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Internal Movement | hcm_120200_load_churn_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History Master | Churn.sas | | |

***Table 39.23*** *hcm_120250_load_empmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Employee Master | hcm_120250_load_empmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Current Work Group<br>Current Jobs<br>Current Pay Grade<br>Current Employee General | Empmast.sas | | |

***Table 39.24*** *hcm_120300_load_oposmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Open Position Master | hcm_120300_load_oposmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Open Position History<br>Position History<br>Work Group<br>Jobs | Oposmast.sas | | |

***Table 39.25*** *hcm_120350_load_termmast_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Terminations Master | hcm_120350_load_termmast_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History Master | Terminat.sas | | |

*Table 39.26 hcm_125050_load_headsum_summary_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Headcount Summary | hcm_125050_load_headsum_summary_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grade<br>Current Employee General<br>Job Action History Master | Headsum.sas | | |

*Table 39.27 hcm_125100_load_opossum_ summary_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Open Position Summary | hcm_125100_load_opossum_ summary_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Open Position History<br>Position History<br>Work Group<br>Jobs | Opossum.sas | | |

*Table 39.28 hcm_125150_load_salhist_summary_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Salary History | hcm_125150_load_salhist_summary_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History Master | Salhist.sas | | |

***Table 39.29*** *hcm_125200_load_salhsum_summary_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Salary History Summary | hcm_125200_load_salhsum_summary_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Compensation<br>Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grades<br>Current Employee General | Salhsum.sas | | |

***Table 39.30*** *hcm_125250_load_tip_summary_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Time In Position | hcm_125250_load_tip_summary_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History Master | Tip.sas | | |

***Table 39.31*** *hcm_126050_run_month_forecast (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Headcount Forecast - Monthly | hcm_126050_run_month_forecast |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grade<br>Current Employee General<br>Job Action History Master | Headfcst.sas | | |

***Table 39.32*** *hcm_126100_run_quarter_forecast (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Headcount Forecast - Quarterly | hcm_126100_run_quarter_forecast |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grade<br>Current Employee General<br>Job Action History Master | Headfcst.sas | | |

***Table 39.33*** *hcm_126100_run_year_forecast (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Headcount Forecast - Yearly | hcm_126100_run_year_forecast |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grade<br>Current Employee General<br>Job Action History Master | Headfcst.sas | | |

***Table 39.34*** *hcm_128050_load_sas_measures_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS Measures | hcm_128050_load_sas_measures_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Headcount Summary<br>Job Action History Master<br>Salary History Summary | Hcmmval.sas | | |

***Table 39.35** hcm_128100_load_sas_measures_table_with_org (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS Measures with Org | hcm_128100_load_sas_measures_table_with_org |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Headcount Summary<br>Job Action History Master<br>Salary History Summary | Hcmmval.sas | | |

***Table 39.36** hcm_128900_load_sdm_metric_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SDM Metrics Table | hcm_128900_load_sdm_metric_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| SAS Measures | Code | | An optional job that you can run if you have SAS Strategic Management. If necessary, the following default Options parameters need to be modified: TIME_PERIOD_ID\|++\|TIME\|++\|TIME\|++\|TIME_MR |

***Table 39.37** hcm_128901_load_sdm_metric_table_with_org (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SDM Metrics Table with Org | hcm_128901_load_sdm_metric_table_with_org |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| SAS Measures | Code | | You can either run this job or hcm_128900_load_sdm_metric_table, but not both. If necessary, the following default Options parameters need to be modified: TIME_PERIOD_ID\|++\|T IME\|++\|TIME\|++\|TIME_MR\|++\|INTORG_HR_ID\|++\|ORG\|++\|INTORG\|++\|INTORG_HR |

***Table 39.38** hcm_140050_load_hrvanly2_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | hrvanly2 | hcm_140050_load_hrvanly2_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Compensation<br>Job Action History<br>Position History<br>Jobs<br>Work Group<br>Pay Grades<br>Current Employee General<br>Job Action History Master<br>Time in Position | Vtadata.sas | | An optional job that is run for the Retention Analysis Module. |

***Table 39.39*** *hcm_140100_load_modelscores_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Employee Scoring Detail | hcm_140100_load_modelscores_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| hrvanly2 | Vtamodel.sas | | An optional job that is run for the Retention Analysis Module. |

***Table 39.40*** *hcm_140150_load_empscores_table (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Employee Scoring Master | hcm_140150_load_empscores_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Employee Scoring Detail<br>Employee Master | Vtscores.sas | | An optional job that is run for the Retention Analysis Module. |

***Table 39.41*** *hcm_129990_load_sas_hierarchy_mapping_table*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | SAS_HIERARCHY_MAPPING<br>SAS_WRKGRP HIER | hcm_129990_load_sas_hierarchy_mapping_table |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| cind_dds.Internal_org_assoc_type | Updthier.sas | | Use this job to load and update the required Org Hierarchy information for SAS Human Capital Management. |

***Table 39.42*** *hcm_210050_create_abshcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Absence History Cube | hcm_210050_create_abshcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Absence History Master | Abshcube.sas | | |

***Table 39.43*** *hcm_210100_create_acthcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Job Action History Cube | hcm_210100_create_acthcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Job Action History Master | Acthcube.sas | | |

***Table 39.44*** *hcm_210150_create_apphcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Applicant History Cube | hcm_210150_create_apphcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Applicant History Master | Apphcube.sas | | |

***Table 39.45*** *hcm_210200_create_empcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Employee Master Cube | hcm_210200_create_empcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Employee Master | Empcube.sas | | |

*Table 39.46 hcm_210250_create_hdsmcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Headcount History Cube | hcm_210250_create_hdsmcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Headcount Summary | Hdsmcube.sas | | |

*Table 39.47 hcm_210300_create_oposcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Open Position Cube | hcm_210300_create_oposcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Open Position Master | Oposcube.sas | | |

*Table 39.48 hcm_210350_create_salhcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Salary History Cube | hcm_210350_create_salhcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Salary History | Salhcube.sas | | |

*Table 39.49 hcm_210400_create_termcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Terminations Cube | hcm_210400_create_termcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Terminations Master | Termcube.sas | | |

***Table 39.50*** *hcm_210450_create_tipcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Time in Position Cube | hcm_210450_create_tipcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Time In Position | Tipcube.sas | | |

***Table 39.51*** *hcm_210500_create_opsmcube_cube*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | Open Position Summary Cube | hcm_210500_create_opsmcube_cube |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Open Position Summary | Opsmcube.sas | | |

***Table 39.52*** *hcm_300000_create_information_maps*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | | hcm_300000_create_information_maps |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Cubes and Tables exist. | Hcmimap.sas | | This job builds and rebuilds all Information Maps that are registered in SAS Human Capital Management. |

***Table 39.53*** *hcm_900000_refresh_cache*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | | hcm_900000_refresh_cache |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| The mid-tier Web applications need to be running. | Refcache.sas | | Refreshes the SAS Human Capital Management cache. This job should be run anytime that one or more of the previously specified jobs have been run. |

***Table 39.54*** *hcm_100000_load_datamart_all_tables (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | | hcm_100000_load_datamart_all_tables |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Deployed Jobs | Runhcm.sas | | With the exception of the information_maps job, this umbrella job runs all of the previously specified non-optional jobs. This umbrella job executes the Runhcm.sas macro. Check the Runhcm.sas macro for tables to be built, and modify the Runhcm.sas macro as necessary. This umbrella job can be executed from within SAS Data Integration Studio. After the jobs are deployed, this umbrella job might need to be modified to edit the default path for the deployed jobs. |

***Table 39.55*** *hcm_200000_create_all_cubes (Optional)*

| Site Checkoff | Table Name | Job Name |
|---|---|---|
| | | hcm_200000_create_all_cubes |

| Dependencies | Mapped or Code | Lookups | Description |
|---|---|---|---|
| Source Tables | Code | | An umbrella job that executes the code to build and refresh the default cubes in SAS Human Capital Management that were built by previous cube jobs. Edit the transformation in the job to change the specification of the cubes that are built and refreshed. |

# *Chapter 40*
# HCM Data Mart Tables

**HCM Data Mart Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 591

## HCM Data Mart Tables

The following tables contain descriptions of the tables in the HCM Data Mart. There is a table for each table category.

***Table 40.1*** *Cube Table*

| Table Name | Table Description | Comment |
|---|---|---|
| ABSHCUBE | HCM Absence History Cube | OLAP cube containing employee absence information. |
| ACTHCUBE | HCM Action History Cube | OLAP cube containing employee job action information. |
| APPHCUBE | HCM Applicant History Cube | OLAP cube containing employment application information. |
| EMPCUBE | HCM Employee Master Cube | OLAP cube containing employee personnel information, such as ethnicity, gender, and marital status. |
| HDSMCUBE | HCM Headcount Summary Cube | OLAP cube containing headcount information over a period of time. |
| OPOSCUBE | HCM Open Position Cube | OLAP cube containing open job position information. |
| OPSMCUBE | HCM Open Position Summary Cube | OLAP cube containing open job position summary information. |
| SALHCUBE | HCM Salary History Cube | OLAP cube containing historical compensation information for the organization, summarized at different dimensions. |

| Table Name | Table Description | Comment |
|---|---|---|
| TERMCUBE | HCM Terminations Cube | OLAP cube containing employee termination or separation information. |
| TIPCUBE | HCM Time in Position Cube | OLAP cube containing information about employees' time in job positions. |

***Table 40.2*** *DETAIL Table*

| Table Name | Table Description | Comment |
|---|---|---|
| ABSHIST | HCM Absence Detail | Absence history of employees. |
| ACTHIST | HCM Job Actions History | Job action data for all employees who were active on the job within the time period covered by the HCM Data Mart. Contains the employee hire, separation records, and other job actions within that time span. When the data mart covers a shorter time span than the life of the organization, a baseline record for the start date of the data mart is required. |
| APPHIST | HCM Applicant History | History of tracked individuals applying for positions within the organization. |
| CGRADE | HCM Current Pay Grades | Currently active information about pay grades, compa-ratio, mid, min, and max, or pay bands. |
| CJOBS | HCM Current Jobs | Current job codes to track job groups and Equal Employment Opportunity (EEO) classifications. |
| COMPHIST | HCM Additional Compensation History | Additions to annualized salaries to generate total compensation including information such as bonuses or car allowances. |
| CWRKGRP | HCM Current Work Group | Tracks the current organizational reporting group and its parent group. |
| EDUHIST | HCM Education History | Employee historical education information. |
| EDUVAL | HCM Education Assessment | Measures used for tracking and assessing employees' education. The measures include data such as GPA and class ranking. |

| Table Name | Table Description | Comment |
|---|---|---|
| EMPGEN | HCM Employee General | Basic employee information including name, diversity, and historically tracked employee information. |
| GRADE | HCM Pay Grades History | Historical information about pay grades, compa-ratio, mid, min, and max, or pay bands. |
| HCBNCHMRK | HCM Human Capital Benchmarks | Saratoga benchmark data, if available. |
| JOBS | HCM Jobs | Job codes to track job groups and Equal Employment Opportunity (EEO) classifications. |
| MODELSCORES | HCM Employee Scoring | Employee scoring for predicted termination, with Retention Modeling information. |
| OPENPOS | HCM Open Positions | History detail of open positions. |
| POS | HCM Positions | History detail of positions, including the dates that each position was opened, closed, filled, and inactive. |
| WRKGRP | HCM Work Group | Historical organizational changes listing the reporting group and its parent group. |

***Table 40.3*** *Master Table*

| Table Name | Table Description | Comment |
|---|---|---|
| ABSHMAST | HCM Absence History Master | Denormalized table that joins absence history with employee general, jobs, and organizational hierarchy data. |
| ACTHMAST | HCM Job Actions History Master | Denormalized table that contains employee actions as well as pertinent job groupings, organizational, management, and geographical hierarchies. |
| APPHMAST | HCM Applicant History Master | Denormalized table that joins an applicant’s history with organizational hierarchy information and position information. |
| CHURN | HCM Churn (Internal Movement) | Internal movement actions within the organization. |
| EMPMAST | HCM Employee Master | Identifies all employees by employee number and contains current personnel information for each employee. |

| Table Name | Table Description | Comment |
|---|---|---|
| EMPSCORES | HCM Employee Scoring Master | Employee scoring for predicted termination with current employee information. |
| OPOSMAST | HCM Open Position Master | History detail of open positions joined with jobs, grade, and employee data. |
| TERMAST | HCM Terminations Master | Employee termination actions from the Job Actions History Master table. |

***Table 40.4*** *Summary Table*

| Table Name | Table Description | Comment |
|---|---|---|
| FACTORSITE | HCM Factor Site | Site-defined factors used for HCM metrics. |
| HCMMONTHFORECAST | HCM Headcount Forecast - Monthly | Headcount, termination count, hire count, and other action counts forecasted at a monthly interval using different classification dimensions. |
| HCMQTRFORECAST | HCM Headcount Forecast - Quarterly | Headcount, termination count, hire count, and other action counts forecasted at a quarterly interval using different classification dimensions. |
| HCMYEARFORECAST | HCM Headcount Forecast - Yearly | Headcount, termination count, hire count, and other action counts forecasted at a yearly interval using different classification dimensions. |
| HEADSUM | HCM Headcount Summary | Headcounts and terminations by month, quarter, or year, as specified in the prebuild.sas file. |
| OPOSSUM | HCM Open Positions Summary | Open position count summaries by specified classification dimensions. |
| SALHIST | HCM Salary History | History of employee salary changes. |
| SALHSUM | HCM Salary Summary | Employee compensation by month, quarter, or year, as specified in the prebuild.sas file. |
| TIP | HCM Time in Position | Employee time in a position. |

*Table 40.5 Utility Table*

| Table Name | Table Description | Comment |
|---|---|---|
| HIERMAX | HCM Hiermax | Lists the highest hierarchy level for each hierarchy that is identified within HCM. |
| HRV_GBL | HCM Global Map Data | Global map data used for the HCM Geographic Analysis. |
| HRV_STATE | HCM US State Map Data | U.S. state map data used for the HCM Geographic Analysis. |
| HRV_US | HCM US Map Data | U.S. map data used for the HCM Geographic Analysis. |
| HRVANLY2 | HCM Hrvanly2 | Table used by the Retention Analysis jobs. This table is created when the HCM job hcm_140050_load_hrvanly2_table runs the first time. |
| SAS_HCM_FORMATS | HCM Formats | HCM SAS formats used for data display and in the HCM ETL build. |
| SAS_HCMMETACOLUMN | HCM SAS_hcmmetacolumn | HCM data dictionary table that contains metadata for registered HCM table columns. |
| SAS_HCMMETATABLE | HCM SAS_hcmmetatable | HCM data dictionary table that contains metadata for registered HCM tables. |
| SAS_MEASURES | HCM Measures | Available metric calculations and benchmarks for the organization. |
| SAS_MEASURES<OrgHier> | HCM Measures by Org | Available metric calculations and benchmarks for the organization identified within the Org hierarchy. |
| SAS_USER_EMPLOYEE | HCM Employee Users | Employees and their associated user names for HCM. An HCM User cannot log in to HCM until his or her information is populated in this table. |

# Chapter 41
# Descriptions of HCM Data Mart Table Columns

Descriptions of HCM Data Mart Table Columns . . . . . . . . . . . . . . . . . . . . . . . . . 597

## Descriptions of HCM Data Mart Table Columns

Each table in this chapter contains descriptions of all the columns in a particular HCM Data Mart table. The columns contained in each table can vary based on your specific needs. The tables are listed in alphabetical order.

***Table 41.1*** *ABSHIST Table*

| Column Name | Comment |
|---|---|
| ABSENCE_END_DT | Ending date of the employee absence. |
| ABSENCE_ID | A unique ID for an absence. |
| ABSENCE_START_DT | Effective or beginning date of the employee absence. |
| ABSENCE_TYPE_CD | Code for a type of absence. Typical absence types are sickness, vacation, disability, family leave, and bereavement. |
| DURATION_QTY | Duration or length of the employee absence. |
| DURATION_TIME_UOM_CD | Time unit of measure for the absence duration. Typically, this has a value of DAY or HOUR. |
| EMPLOYEE_ID | Business key identifying the employee. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |

*Table 41.2* *ABSHMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record for the employee. |
| ABSENCE_END_DT | Ending date of the employee absence. |
| ABSENCE_ID | A unique ID for an absence. |
| ABSENCE_MONTH | A number representing the month of the absence. 1 = January, 2 = February, 3 = March, and so on. |
| ABSENCE_START_DT | Effective or beginning date of the employee absence. |
| ABSENCE_TYPE_CD | Code for a type of absence. Typical absence types are sickness, vacation, disability, family leave, and bereavement. |
| ABSENCE_YEAR | Year of absence. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| BDAY | The day of the month on which the employee was born. Possible values are 1 through 31. |
| BIRTH_DT | Employee's birth date. |
| CITY_NM | Employee's city. |
| COUNTRY_CD | This column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DURATION_QTY | Duration or length of the employee absence. |
| DURATION_TIME_UOM_CD | Time unit of measure for the absence duration. Typically, this has a value of DAY or HOUR. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |

| Column Name | Comment |
|---|---|
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the absence. |
| SERVICE_START_DT | Date that an employee began to work with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |

| Column Name | Comment |
|---|---|
| WKDAY | A number representing the day of the week of the absence. 1 = Sunday, 2 = Monday, 3 = Tuesday, and so on. |

*Table 41.3 ACTHIST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record for the employee. |
| ACTION_DT | The date of the employee action. |
| ACTION_SEQ_NO | Determines the correct sequence of actions for employee actions that occur on the same day. |
| ACTION_TYPE_CD | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EMPLOYEE_ACTION_REASON_CD | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, or year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |

| Column Name | Comment |
|---|---|
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| JOB_CD | A code that represents a job within the company. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.4*** *ACTHMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record for the employee. |

| Column Name | Comment |
|---|---|
| ACTION_DT | The date of the employee action. |
| ACTION_MONTH | A number representing the month of the employee action. 1 = January, 2 = February, 3 = March, and so on. |
| ACTION_SEQ_NO | Determines the correct sequence of actions for employee actions that occur on the same day. |
| ACTION_TYPE_CD | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| ACTION_YEAR | The year of the employee action. |
| ACTIONS | Given a count of 1 for each action. This is used in the summarization for OLAP cubes. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| AMT_OVER_MAXIMUM | The amount that the employee's pay is over his or her pay grade maximum (where applicable). |
| AMT_UNDER_MINIMUM | The amount that the employee's pay is under his or her pay grade minimum (where applicable). |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CITY_NM | Employee's city. |
| COMRATIO | A ratio of the employee's salary to the employee’s pay grade mid-point. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DATE_IN_GRADE | The date the employee moved into his or her pay grade. |

| Column Name | Comment |
|---|---|
| DATE_IN_JOB | The date the employee moved into his or her job. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ACTION_REASON_CD | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, or year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, or RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| FTEQUIV | The calculated full-time Annual Pay Equivalent salary. |
| GENDER_CD | Code used to specify the gender. |

| Column Name | Comment |
|---|---|
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PCT_OVER_MAXIMUM | The percent the employee's pay is over his or her pay grade maximum (where applicable). |
| PCT_UNDER_MINIMUM | The percent the employee's pay is under his or her pay grade minimum (where applicable). |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the employee action. |
| RNG_PENE | Range penetration of the employee's salary in his or her pay grade range. |

| Column Name | Comment |
|---|---|
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| UNION_CD | Unique code indicating labor union. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.5*** *APPHIST Table*

| Column Name | Comment |
|---|---|
| APPLICANT_BIRTH_DT | Birth date of the applicant. |
| APPLICANT_CITY_NM | City of the applicant. |
| APPLICANT_COUNTRY_CD | Country of the applicant. A three-character ISO 3166 standard country code. For example, AFG = Afghanistan and ALB = Albania. |
| APPLICANT_COUNTY_NM | County of the applicant. |
| APPLICANT_ETHNICITY_CD | Ethnicity of the applicant. |
| APPLICANT_GENDER_CD | Gender of the applicant. |
| APPLICANT_INTERVIEWED_FLG | Indicates whether the applicant was interviewed or not. |
| APPLICANT_NAME | Name of applicant. |
| APPLICANT_POSTAL_CD | Address postal code. |
| APPLICANT_SOCIAL_SECURITY_NO | Applicant's Social Security number. Specific to the United States. |
| APPLICANT_STATE_REGION_CD | Unique code for applicant's state or region. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| APPLICATION_DT | Date of the application. |

| Column Name | Comment |
|---|---|
| APPLICATION_STATUS_CD | The status of the employment application. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYMENT_APPLICATION_ID | A unique ID for the employment application. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| POSITION_CD | A code that represents a position within the company. |
| RECRUITMENT_SOURCE_CD | Code for the recruitment source that the applicant used. |
| REJECTION_REASON_CD | A code that represents an employment application rejection reason. |

***Table 41.6*** *APPHMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y. |
| APPAGE | Age of the applicant. |
| APPLICANT_BIRTH_DT | Birth date of the applicant. |
| APPLICANT_CITY_NM | City of the applicant. |
| APPLICANT_COUNTRY_CD | Country of the applicant. A three-character ISO 3166 standard country code. For example, AFG = Afghanistan and ALB = Albania. |
| APPLICANT_COUNTY_NM | County of the applicant. |
| APPLICANT_ETHNICITY_CD | Ethnicity of the applicant. |
| APPLICANT_GENDER_CD | Gender of the applicant. |
| APPLICANT_INTERVIEWED_FLG | Indicates whether the applicant was interviewed or not. |
| APPLICANT_NAME | Name of applicant. |
| APPLICANT_POSTAL_CD | Address postal code. |
| APPLICANT_SOCIAL_SECURITY_NO | Applicant's Social Security number. Specific to the United States. |
| APPLICANT_STATE_REGION_CD | Unique code for applicant's state or region. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |

| Column Name | Comment |
|---|---|
| APPLICATION_DT | Date of the application. |
| APPLICATION_MONTH | A number representing the month of the application. 1 = January, 2 = February, 3 = March, and so on. |
| APPLICATION_STATUS_CD | The status of the employment application. |
| APPLICATION_YEAR | Year of the application. |
| APPOV40 | A flag indicating whether the applicant is 40 or older. Y = Yes, N = No. |
| CNT | Given a count of 1 for each application. This is used in the summarization for OLAP cubes. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYMENT_APPLICATION_ID | A unique ID for the employment application. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| INTORG_HR | The department that the applicant position is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table, HIERARCHY_CD column. If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_HR5 | Level 5 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the employment application. |

| Column Name | Comment |
|---|---|
| RECRUITMENT_SOURCE_CD | Code for the recruitment source that the applicant used. |
| REJECTION_REASON_CD | A code that represents an employment application rejection reason. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |

*Table 41.7 CGRADE Table*

| Column Name | Comment |
|---|---|
| ACTIVE_FLG | Indicates whether the pay grade is active or not. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| MAXIMUM_AMT | Maximum amount for the pay grade. |
| MIDPOINT_AMT | Midpoint amount for the pay grade. This column might not apply to all pay grades. |
| MIDPOINT_DIFF | The percentage difference between midpoints of successive grades. |
| MINIMUM_AMT | Minimum amount for the pay grade. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | A code that represents a pay grade. |
| PAY_LEVEL_DESC | Description of the pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. If PAY_LEVEL_CD is unique for each record, then PAY_LEVEL_STRUCTURE_CD and PAY_LEVEL_CD can be the same in every record. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| RANGE_SPREAD | Percentage difference from the minimum to the maximum of the grade. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

*Table 41.8 CHURN Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record from ACTHIST for the employee. |
| ACTION_DT | The date of the employee action. |
| ACTION_MONTH | A number representing the month of the employee action. 1 = January, 2 = February, 3 = March, and so on. |
| ACTION_SEQ_NO | Determines the correct sequence of actions for employee actions that occur on the same day. |
| ACTION_TYPE_CD | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| ACTION_YEAR | The year of the employee action. |
| ACTIONS | Given a count of 1 for each action. This is used in the summarization for OLAP cubes. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| AMT_OVER_MAXIMUM | The amount the employee's pay is over his or her pay grade maximum (where applicable). |
| AMT_UNDER_MINIMUM | The amount the employee's pay is under his or her pay grade minimum (where applicable). |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CITY_NM | Employee's city. |
| COMRATIO | A ratio of the employee's salary to the employee’s pay grade mid-point. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |

| Column Name | Comment |
|---|---|
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DATE_IN_GRADE | The date the employee moved into his or her pay grade. |
| DATE_IN_JOB | The date the employee moved into his or her job. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ACTION_REASON_CD | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, or year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, or RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |

| Column Name | Comment |
|---|---|
| FTEQUIV | The calculated full-time Annual Pay Equivalent salary. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PCT_OVER_MAXIMUM | The percent the employee's pay is over his or her pay grade maximum (where applicable). |
| PCT_UNDER_MINIMUM | The percent the employee's pay is under his or her pay grade minimum (where applicable). |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| PINTORG_HR | Employee's previous department prior to the Internal Transfer action. |
| POSITION_CD | A code that represents a position within the company. |

| Column Name | Comment |
|---|---|
| POSTAL_CD | Address postal code. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the internal transfer action. |
| RNG_PENE | Range penetration of the employee's salary in his or her pay grade range. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.9*** *CJOBS Table*

| Column Name | Comment |
|---|---|
| ACTIVE_FLG | Indicates whether the job is active or not. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.10*** *COMPHIST Table*

| Column Name | Comment |
|---|---|
| COMPENSATION_AMT | Amount of compensation paid to the employee. |
| COMPENSATION_TYPE_CD | Type of compensation. |
| EMPLOYEE_ID | Business key identifying the employee. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| PAYMENT_DT | Date that the compensation was paid. |

***Table 41.11*** *CWRKGRP Table*

| Column Name | Comment |
|---|---|
| GRPDESC | Description of the Organization Report Group. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIERARCHY_CD | Code designating the organizational hierarchy defined in this table. Multiple organizational hierarchies can be defined in this table. |
| INTORG | Code for a department or manager. This is the child of PARENT_INTORG. |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| PARENT_INTORG | Code for a department or manager. This is the parent of INTORG. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.12*** *EDUHIST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | A flag that designates whether or not this is the last record for an employee's educational enrollment. Value of Y or N. |
| ACADEMIC_CREDIT_CD | Code that indicates the type of credit awarded by the institution. Examples: A = adult credits, C = continuing education units, N = no credit, Q = quarter-hour credit. |

| Column Name | Comment |
|---|---|
| ACADEMIC_CREDIT_HRS_ATTPD_NO | Total number of credits or credit hours attempted and earned and included in this summary. Examples: Credit hours to be included, if available, are credits for which non-punitive grades such as "I" or "W" or "Audit" were awarded. |
| ACADEMIC_CREDIT_HRS_ERND_NO | Total number of credits or credit hours included on the record for this particular summary. This is normally all credits for which the student paid, whether the credits were used to calculate the grade point average. |
| ACADEMIC_CREDIT_HRS_INCLD_NO | Total number of credits or credit hours included in the grade point average for this particular summary. Inclusion or exclusion of certain credits depends on the policy of the sending institution. |
| ACADEMIC_HONORS_CD | Code for any academic honors associated with the degree. Example: magna cum laude. |
| ADDRESS_LINE_1_SCHOOL | First line of school address. |
| ADDRESS_LINE_2_SCHOOL | Second line of school address. |
| ANTICIPATED_GRADUATION_DT | Date on which the student is scheduled to graduate. |
| ATTENDANCE_END_DT | Ending date of attendance at an educational institution. |
| ATTENDANCE_START_DT | Beginning date of attendance at an educational institution. |
| ATTENDANCE_STATUS_CD | Code for the status of the student. Examples: A student transfers from the School of Business to the School of Journalism. The first record shows the School of Business with an attendance status code of TRNSFR and the second record shows the School of Journalism with an attendance status code of ATTENDING. |
| CITY_NM_SCHOOL | City of the school. |
| CLASS_RANK_NO | A student's numerical class rank with the highest student in the class having a rank or position of 1. |
| COMMENTS_TXT | Comments relating to education. |
| COUNTRY_CD_SCHOOL | This column contains the two-character ISO 3166 standard code for a country. For example, AF = Afghanistan and AL = Albania. |
| COURSE_LEVEL_CD | Code established by the National Center for Education Statistics to indicate the level and/or the type of work, which is reflected in the grade average and the credit hours. Examples: 1 = Remedial; 2 = Basic; 3 = Teacher's Aide; 4 = General; A = Summary of all courses taken at all institutions; AR = Academic Renewal. |

| Column Name | Comment |
|---|---|
| CUMULATIVE_SUMMARY_FLG | Flag that indicates whether this is a summary of all work included in the record. |
| CURRENTLY_ENROLLED_FLG | Indicates whether the person is currently enrolled as a student. |
| DEGREE_CONCENTRATION_CD | Code for the emphasis associated with a major or field of study. Example: A masters degree in mathematics with a concentration in Bioinformatics. |
| DEGREE_DESCRIPTION | Description of the degree that was attained. |
| DEGREE_OPTION_CD | Code for the official name of a major option associated with the student's degree title. Examples: A particular institution might require its Civil Engineering majors to select either a mechanical engineering option or a structural engineering option. |
| DEGREE_PROGRAM_CD | Code reflecting the program associated with the course of study. |
| DEGREE_TYPE_CD | Code for the type of degree. Examples: doctorate, masters, bachelors, associate certificate. |
| EDUCATION_ACHIEVEMENT_NM | Name of the educational achievement attained. |
| EDUCATION_HISTORY_ID | Unique ID for the employee's education history record. |
| EDUCATION_VALUE_SYSTEM_CD | Code for the scale against which the educational measure can be compared. Business rules: For a grade point average (GPA), this might contain the highest GPA that can be attained at this institution. For a class rank, this might contain the size of the class. |
| EDUHIST_SEQ_NO | Determines the correct sequence of employee education history records that occur on the same day. The table should be sorted by EDUCATION_HISTORY_ID, VALID_FROM_DT, and then EDUHIST_SEQ_NO. |
| EMPLOYEE_ID | Business key identifying the employee. |
| ENROLLMENT_STATUS_CD | Code for enrollment status. Examples: regular, night, continuing education. |
| EXAM_PASSED_FLG | Flag indicates the individual passed an exam related to or required for the education. |
| EXAM_REQUIRED_FLG | Flag indicating if an exam is required to complete the education. |

| Column Name | Comment |
|---|---|
| EXCESSIVE_VALUE_FLG | Flag that indicates whether an excessive value is possible on the relevant scale or rating system. Examples: A "False" value indicates that it is not possible to have a higher grade point average (GPA) than indicated in range maximum. A "True" value indicates it is possible to have a higher GPA than indicated. |
| FICE_CD | The institution's FICE code as identified by the Federal Interagency Committee on Education. FICE codes are assigned by the U.S. Department of Education and are used as the primary identifiers for academic institutions. |
| GOOD_STUDENT_FLG | Flag that indicates whether the person is a good student under the criteria established by the school. Good students might be eligible for insurance discounts. |
| GPA_NO | A student's numerical grade point average. |
| GRADUATING_DEGREE_CD | Code that indicates whether a diploma or certification is issued at the completion of the education. Examples: Graduating, Qualifying, Certification. |
| GRADUATION_DT | Date that the student graduated from the academic program. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HONORS_PROGRAM_CD | Code for any honors program associated with the degree. |
| INTERNET_DOMAIN_NM | A domain name intended as a practical identifier for the organization, typically used for Web and e-mail. Not intended as a Web address (URL). Examples: microsoft.com, bund.de, google.com. |
| MAJOR1_NM | Name of the first major degree. |
| MAJOR2_NM | Name of the second major degree. |
| MINOR1_NM | Name of the first minor degree. |
| MINOR2_NM | Name of the second minor degree. |
| OTHER_HONORS_CD | Code for any other honors that the student received. Example: Phi Beta Kappa. |
| POSTAL_CD_SCHOOL | Address postal code of the school. |
| SCHOOL_DEPT_NM | Name of the school department. |
| SCHOOL_DEPT_TYPE_CD | A type code for a school department. Examples: CS for Computer Science, PT for Physical Therapy. |

| Column Name | Comment |
|---|---|
| SCHOOL_NAME_TYPE_CD | Type code to qualify the school name. Can be used as a broader level classification of the school or institution. |
| SCHOOL_NM | Name of the educational institution. |
| SCHOOL_TYPE_CD | Type code used to distinguish different types of educational institutions. Examples: community college, trade school, university. |
| STATE_REGION_CD_SCHOOL | State or region code for the school. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STUDENT_IN_GOOD_STANDING_FLG | Indicates whether the person in question is a student in good standing. This is frequently a value returned in response to checks to verify a dependent student's eligibility for insurance under a parent's insurance policy. |
| TERMINAL_DEGREE_FLG | A flag to indicate the highest academic degree in a field of study. If a doctorate is the highest degree that a field offers, then it is the terminal degree. If a bachelors or masters are the highest degree available in the field, then that is the terminal degree. |
| TOTAL_STUDENTS_NO | Total number of students in the class, which is used to help position the student's rank. |
| TUITION_ASSISTANCE_FLG | Flag indicating if the student received tuition assistance to attain the education. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.13*** *EDUVAL Table*

| Column Name | Comment |
|---|---|
| _LASTREC | A flag that designates whether this is the last record for an employee's educational enrollment assessment values. Value of Y or N. |
| EDUCATION_HISTORY_ID | Unique ID for the employee's education history record. |
| EDUCATION_NUM_MAX_VAL | Contains the highest possible numeric value on the scale or rating system. Used in conjunction with the EDUCATION_NUM_MIN_VAL column to define the measurement bounds. |
| EDUCATION_NUM_MIN_VAL | Contains the lowest possible numeric value on the scale or rating system. Used in conjunction with the EDUCATION_NUM_MAX_VAL column to define the measurement bounds. |

| Column Name | Comment |
|---|---|
| EDUCATION_NUM_VAL | Numeric value of the educational performance of the person, within the bounds set in the EDUCATION_NUM_MAX_VAL column and EDUCATION_NUM_MIN_VAL column. Examples: If Education Value Type is grade point average (GPA), then this column might contain a value of 3.5. If Education Value Type is rank in class, then this column might contain a value of 130, ranked 130 out of 5000. |
| EDUCATION_STR_MAX_VAL | Used in conjunction with the EDUCATION_STR_MIN_VAL column to identify the low and high measurement bounds. Represents the highest possible string value on the scale or rating system. Example: If Education Value Type is Pass/Fail and the employee passed, then this column contains PASS. |
| EDUCATION_STR_MIN_VAL | Used in conjunction with the EDUCATION_STR_MAX_VAL column to identify the low and high measurement bounds. Contains the lowest possible string value on the scale or rating system. Example: If Education Value Type is Pass/Fail, then this column contains FAIL. |
| EDUCATION_STR_VAL | Represents the value of the educational performance of the person. Example: If Education Value Type is Pass/Fail, and the employee passed, this column contains PASS. |
| EDUCATION_VALUE_TYPE_CD | Code for type of education value. Examples: grade point average, rank in class. Used as part of the measurement of the academic performance of the person who has the education. |
| EDUVAL_SEQ_NO | Determines the correct sequence of employee education assessment records that occur on the same day. The table should be sorted by EDUCATION_HISTORY_ID, EDUCATION_VALUE_TYPE_CD, VALID_FROM_DT, and then EDUVAL_SEQ_NO. |
| EMPLOYEE_ID | Business key identifying the employee. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| SCHOOL_DEPT_NM | Name of the school department. |
| SCHOOL_NM | Name of the educational institution. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

*Table 41.14 EMPGEN Table*

| Column Name | Comment |
|---|---|
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| BIRTH_DT | Employee's birth date. |
| CITIZENSHIP_COUNTRY_CD | Country of citizenship. The three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| CITY_NM | Employee's city. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| MARITAL_STATUS_CD | Code indicating marital status. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MILITARY_EXPERIENCE_TYPE_CD | Military status or experience. Used to show an employee's military or veteran status. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| POSTAL_CD | Address postal code. |

| Column Name | Comment |
|---|---|
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SOCIAL_SECURITY_NO | Employee's Social Security number. Specific to the United States. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| UNION_CD | Unique code indicating labor union. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.15*** *EMPMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record from ACTHIST for the employee. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| AMT_OVER_MAXIMUM | The amount the employee's pay is over his or her pay grade maximum (where applicable). |
| AMT_UNDER_MINIMUM | The amount the employee's pay is under his or her pay grade minimum (where applicable). |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CITIZENSHIP_COUNTRY_CD | Country of citizenship. The three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| CITY_NM | Employee's city. |

| Column Name | Comment |
|---|---|
| COMRATIO | A ratio of the employee's salary to the employee’s pay grade mid-point. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, or year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, or RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |

| Column Name | Comment |
|---|---|
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| FTEQUIV | The calculated full-time Annual Pay Equivalent salary. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MARITAL_STATUS_CD | Code indicating marital status. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MILITARY_EXPERIENCE_TYPE_CD | Military status or experience. Used to show an employee's military or veteran status. |
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PCT_OVER_MAXIMUM | The percent the employee's pay is over his or her pay grade maximum (where applicable). |
| PCT_UNDER_MINIMUM | The percent the employee's pay is under his or her pay grade minimum (where applicable). |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |

| Column Name | Comment |
|---|---|
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| RNG_PENE | Range penetration of the employee's salary in his or her pay grade range. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| SOCIAL_SECURITY_NO | Employee's Social Security number. Specific to the United States. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| UNION_CD | Unique code indicating labor union. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.16*** *EMPSCORES Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record from ACTHIST for the employee. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| AMT_OVER_MAXIMUM | The amount the employee's pay is over his or her pay grade maximum (where applicable). |
| AMT_UNDER_MINIMUM | The amount the employee's pay is under his or her pay grade minimum (where applicable). |

| Column Name | Comment |
|---|---|
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CITIZENSHIP_COUNTRY_CD | Country of citizenship. The three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| CITY_NM | Employee's city. |
| COMRATIO | A ratio of the employee's salary to the employee’s pay grade mid-point. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |

| Column Name | Comment |
|---|---|
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| FTEQUIV | The calculated full-time Annual Pay Equivalent salary. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MARITAL_STATUS_CD | Code indicating marital status. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MILITARY_EXPERIENCE_TYPE_CD | Military status or experience. Used to show an employee's military or veteran status. |

| Column Name | Comment |
|---|---|
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| P_VTERM_F1 | Probability that an employee will terminate, expressed as a decimal less than or equal to 1. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |

| Column Name | Comment |
|---|---|
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PCT_OVER_MAXIMUM | The percent the employee's pay is over his or her pay grade maximum (where applicable). |
| PCT_UNDER_MINIMUM | The percent the employee's pay is under his or her pay grade minimum (where applicable). |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| RNG_PENE | Range penetration of the employee's salary in his or her pay grade range. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| SOCIAL_SECURITY_NO | Employee's Social Security number. Specific to the United States. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| UNION_CD | Unique code indicating labor union. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |
| VTGROUP | A grouping of employees by risk of termination. 0 = Low risk, 1 = Moderate risk, 2 = High risk. |
| VTSCORE | Risk of termination numerical score. |

*Table 41.17 GRADE Table*

| Column Name | Comment |
|---|---|
| ACTIVE_FLG | Indicates whether the pay grade is active or not. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| MAXIMUM_AMT | Maximum amount for the pay grade. |
| MIDPOINT_AMT | Midpoint amount for the pay grade. This column might not apply to all pay grades. |
| MINIMUM_AMT | Minimum amount for the pay grade. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | A code that represents a pay grade. |
| PAY_LEVEL_DESC | Description of the pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. If PAY_LEVEL_CD is unique for each record, then PAY_LEVEL_STRUCTURE_CD and PAY_LEVEL_CD can be the same in every record. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

*Table 41.18 HCMMONTHFORECAST Table*

| Column Name | Comment |
|---|---|
| ABSOLUTELOWER_CHURN | The difference between the predicted value and lower confidence limit value for internal movement. |
| ABSOLUTELOWER_HEADCNTCHG | The difference between the predicted value and lower confidence limit value for change in head count. |
| ABSOLUTELOWER_NHIRES | The difference between the predicted value and lower confidence limit value for new hires. |
| ABSOLUTEUPPER_CHURN | The difference between the predicted value and upper confidence limit value for internal movement. |
| ABSOLUTEUPPER_HEADCNTCHG | The difference between the predicted value and upper confidence limit value for head count change. |

| Column Name | Comment |
|---|---|
| ABSOLUTEUPPER_NHIRES | The difference between the predicted value and upper confidence limit value for new hires. |
| CHURN | Historical data values for internal movement. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| FORECAST_CHURN | Contains the adjusted historical values and the forecast values for internal movement. |
| FORECAST_HEADCNTCHG | Contains the adjusted historical values and the forecast values for change in headcount. |
| FORECAST_NHIRES | Contains the adjusted historical values and the forecast values for new hires. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HEADCNTCHG | Change in employee headcount. |
| HLEVEL | The level in the hierarchy variable used. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| LOWER_CHURN | Lower confidence limit for forecast values of internal movement. |
| LOWER_HEADCNTCHG | Lower confidence limit for forecast values of head count change. |
| LOWER_NHIRES | Lower confidence limit for forecast values of new hires. |
| NHIRES | Historical data values for new hires. |
| PERCENTAGELOWER_CHURN | Percentage difference between the forecast value and lower confidence limit for forecast value of internal movement. |
| PERCENTAGELOWER_HEADCNTCHG | Percentage difference between the forecast value and lower confidence limit for forecast value of change in head count change. |
| PERCENTAGELOWER_NHIRES | Percentage difference between the forecast value and lower confidence limit for forecast value of new hires. |
| PERCENTAGEUPPER_CHURN | Percentage difference between the forecast value and upper confidence limit for forecast value of internal movement. |
| PERCENTAGEUPPER_HEADCNTCHG | Percentage difference between the forecast value and upper confidence limit for forecast value of change in head count change. |
| PERCENTAGEUPPER_NHIRES | Percentage difference between the forecast value and upper confidence limit for forecast value of new hires. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| PREDICTERR_CHURN | Prediction errors for forecast values of internal movement. |
| PREDICTERR_HEADCNTCHG | Prediction errors for forecast values of head count change. |
| PREDICTERR_NHIRES | Prediction errors for forecast values of new hires. |
| PREDICTSTDERR_CHURN | Prediction standard errors for forecast values of internal movement. |

| Column Name | Comment |
|---|---|
| PREDICTSTDERR_HEADCNTCHG | Prediction standard errors for forecast values of head count change. |
| PREDICTSTDERR_NHIRES | Prediction standard errors for forecast values of new hires. |
| RANGE_CHURN | Range between the upper and lower confidence limit for forecast values of internal movement. |
| RANGE_HEADCNTCHG | Range between the upper and lower confidence limit for forecast values of head count change. |
| RANGE_NHIRES | Range between the upper and lower confidence limit for forecast values of new hires. |
| UNION_CD | Unique code indicating labor union. |
| UPPER_CHURN | Upper confidence limit for forecast values of internal movements. |
| UPPER_HEADCNTCHG | Upper confidence limit for forecast values of head count change. |
| UPPER_NHIRES | Upper confidence limit for forecast values of new hires. |
| VALID_DT | The time value of the historical and forecast values. |

*Table 41.19 HCMQTRFORECAST Table*

| Column Name | Comment |
|---|---|
| ABSOLUTELOWER_HEADCNT | The difference between the predicted value and lower confidence limit value for head count. |
| ABSOLUTELOWER_VTERM | The difference between the predicted value and lower confidence limit value for voluntary terminations. |
| ABSOLUTEUPPER_HEADCNT | The difference between the predicted value and upper confidence limit value for head count. |
| ABSOLUTEUPPER_VTERM | The difference between the predicted value and upper confidence limit value for voluntary terminations. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| FORECAST_HEADCNT | Contains the adjusted historical values and the forecast values for headcount. |

| Column Name | Comment |
|---|---|
| FORECAST_VTERM | Contains the adjusted historical values and the forecast values for voluntary terminations. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HEADCNT | Historical data values for headcount. |
| HLEVEL | The level in the hierarchy variable used. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| LOWER_HEADCNT | Lower confidence limit for forecast values of head count. |
| LOWER_VTERM | Lower confidence limit for forecast values of voluntary terminations. |
| PERCENTAGELOWER_HEADCNT | Percentage difference between the forecast value and lower confidence limit for forecast value of head count. |

| Column Name | Comment |
|---|---|
| PERCENTAGELOWER_VTERM | Percentage difference between the forecast value and lower confidence limit for forecast value of voluntary terminations. |
| PERCENTAGEUPPER_HEADCNT | Percentage difference between the forecast value and upper confidence limit for forecast value of head count. |
| PERCENTAGEUPPER_VTERM | Percentage difference between the forecast value and upper confidence limit for forecast value of voluntary terminations. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| PREDICTERR_HEADCNT | Prediction errors for forecast values of head count. |
| PREDICTERR_VTERM | Prediction errors for forecast values of voluntary terminations. |
| PREDICTSTDERR_HEADCNT | Prediction standard errors for forecast values of head count. |
| PREDICTSTDERR_VTERM | Prediction standard errors for forecast values of voluntary terminations. |
| RANGE_HEADCNT | Range between the upper and lower confidence limit for forecast values of head count change. |
| RANGE_VTERM | Range between the upper and lower confidence limit for forecast values of voluntary terminations. |
| UNION_CD | Unique code indicating labor union. |
| UPPER_HEADCNT | Upper confidence limit for forecast values of head count |
| UPPER_VTERM | Upper confidence limit for forecast values of voluntary terminations. |
| VALID_DT | The time value of the historical and forecast values. |
| VTERM | Historical data values for voluntary terminations. |

***Table 41.20*** *HCMYEARFORECAST Table*

| Column Name | Comment |
|---|---|
| ABSOLUTELOWER_ITERM | The difference between the predicted value and lower confidence limit value for involuntary terminations. |
| ABSOLUTEUPPER_ITERM | The difference between the predicted value and upper confidence limit value for involuntary terminations. |

| Column Name | Comment |
|---|---|
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| FORECAST_ITERM | Contains the adjusted historical values and the forecast values for involuntary terminations. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HLEVEL | The level in the hierarchy variable used. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| ITERM | Historical data values for voluntary terminations. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| LOWER_ITERM | Lower confidence limit for forecast values of involuntary terminations. |

| Column Name | Comment |
|---|---|
| PERCENTAGELOWER_ITERM | Percentage difference between the forecast value and lower confidence limit for forecast value of involuntary terminations. |
| PERCENTAGEUPPER_ITERM | Percentage difference between the forecast value and upper confidence limit for forecast value of involuntary terminations. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| PREDICTERR_ITERM | Prediction error for forecast values of involuntary terminations. |
| PREDICTSTDERR_ITERM | Prediction standard errors for forecast values of involuntary terminations. |
| RANGE_ITERM | Range between the upper and lower confidence limit for forecast values of involuntary terminations. |
| UNION_CD | Unique code indicating labor union. |
| UPPER_ITERM | Upper confidence limit for forecast values of involuntary terminations. |
| VALID_DT | The time value of the historical and forecast values. |

***Table 41.21*** *HEADSUM Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last headcount snapshot for the employee. Otherwise, given a value of N. |
| AVGTURN | This is a place holder for system use only. It does not contain any data. |
| CHURN | Employee Internal Transfer action count. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |

| Column Name | Comment |
|---|---|
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FTECNT | This is a place holder for system use only. It does not contain any data. |
| FTEDIF | This is a place holder for system use only. It does not contain any data. |
| FTEGROW | This is a place holder for system use only. It does not contain any data. |
| GENDER_CD | Code used to specify the gender. |
| GROWTH | This is a place holder for system use only. It does not contain any data. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HDDIF | This is a place holder for system use only. It does not contain any data. |
| HEADCNT | Count of employees. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| ITERM | Employee Involuntary Termination action count. |
| ITERMDIF | This is a place holder for system use only. It does not contain any data. |
| ITERMSEP | This is a place holder for system use only. It does not contain any data. |
| NHIREDIF | This is a place holder for system use only. It does not contain any data. |
| NHIRES | Employee New Hire action count. |
| PAY | Employee Change in Pay action count. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the headcount snapshot date. |
| TERMDIF | This is a place holder for system use only. It does not contain any data. |
| TERMS | Employee Termination action count. Includes both voluntary and involuntary terminations. |

| Column Name | Comment |
|---|---|
| TURNOVER | This is a place holder for system use only. It contains no data. |
| VALID_DT | Date of the employee headcount snapshot. |
| VALID_MONTH | A number representing the month of the snapshot date. 1 = January, 2 = February, 3 = March, and so on. |
| VALID_YEAR | The year of the snapshot date. |
| VTERM | Employee Voluntary Termination action count. |
| VTERMDIF | This is a place holder for system use only. It does not contain any data. |
| VTERMSEP | This is a place holder for system use only. It does not contain any data. |

***Table 41.22*** *JOBS Table*

| Column Name | Comment |
|---|---|
| ACTIVE_FLG | Indicates whether the job is active or not. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

*Table 41.23 OPENPOS Table*

| Column Name | Comment |
|---|---|
| CANCELLATION_DT | Date that the position was canceled. |
| CLOSE_DT | Closing date for the position. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| OPEN_DT | Date on which the position was opened. |
| POSITION_CD | A code that represents a position within the company. |
| REQUISITION_COST_AMT | The cost of the position requisition. |
| REQUISITION_NO | The requisition number for the position. |

*Table 41.24 OPOSMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record where the position is open. Otherwise, given a value of N. |
| CANCELLATION_DT | Date that the position was canceled. |
| CLOSE_DT | Closing date for the position. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| INTORG_HR | The department that the position is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table, HIERARCHY_CD column. If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_HR2 | Level 2 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| OPEN_DT | Date on which the position was opened. |
| OPEN_MONTH | A number representing the month of the year the position became open. 1 = January, 2 = February, 3 = March, and so on. |
| OPEN_YEAR | The year the position became open. |
| OPENTIME | Number of days the position was open. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year the position became open. |
| REQUISITION_COST_AMT | The cost of the position requisition. |
| REQUISITION_NO | The requisition number for the position. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |

***Table 41.25*** *OPOSSUM Table*

| Column Name | Comment |
|---|---|
| CNTPCT | Open position count as a percentage of the open position count for the entire organization. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |

| Column Name | Comment |
|---|---|
| INTORG_HR | The department that the position is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table, HIERARCHY_CD column. If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy to which the position is assigned. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| OPCHGPCT | Change in open position count as a percentage. |
| OPDIF | Change in open position count. |
| OPOSCNT | Count of open positions. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| TOTCNT | Number of open positions in the entire organization as of the open position date. |
| VALID_DT | Date of the open position snapshot. |

***Table 41.26*** *POS Table*

| Column Name | Comment |
|---|---|
| EMPLOYEE_ID | Business key identifying the employee. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |

| Column Name | Comment |
|---|---|
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HRPAYPER | Number of hours per pay period for the position. |
| INTERNAL_ORG_ID | Department to which the position is assigned. |
| JOB_CD | A code that represents a job within the company. |
| PAY_FREQUENCY_CD | Frequency of pay for the position. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POS_FTE_RT | Full-time equivalence ratio for the position, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| POSITION_CD | A code that represents a position within the company. |
| POSITION_STATUS_CD | Status of the position. For example, open, canceled, or filled. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.27*** *SALHIST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last record from ACTHIST for the employee. |
| ACTION_DT | The date of the employee action. |
| ACTION_SEQ_NO | Determines the correct sequence of actions for employee actions that occur on the same day. |
| ACTION_TYPE_CD | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |

| Column Name | Comment |
|---|---|
| AMT_OVER_MAXIMUM | The amount the employee's pay is over his or her pay grade maximum (where applicable). |
| AMT_UNDER_MINIMUM | The amount the employee's pay is under his or her pay grade minimum (where applicable). |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CHNGEST | Employee's change in annual salary. Value is positive or negative. |
| CHNGMNTH | Employee's change in monthly salary. Value is positive or negative. |
| CHNGPCT | Employee's percent change in salary. Value is positive or negative. |
| CHNGSAL | Employee's change in hourly salary. Value is positive or negative. |
| CITY_NM | Employee's city. |
| COMRATIO | A ratio of the employee's salary to the employee’s pay grade mid-point. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DATE_IN_GRADE | The date the employee moved into his or her pay grade. |
| DATE_IN_JOB | The date the employee moved into his or her job. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ACTION_REASON_CD | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. |

| Column Name | Comment |
|---|---|
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |
| FTEQUIV | The calculated full-time Annual Pay Equivalent salary. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |

| Column Name | Comment |
|---|---|
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |

| Column Name | Comment |
|---|---|
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| OLD_EST | Previous annual salary. |
| OLD_MNTH | Previous monthly salary. |
| OLD_SAL | Previous hourly salary. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PCT_OVER_MAXIMUM | The percent the employee's pay is over his or her pay grade maximum (where applicable). |
| PCT_UNDER_MINIMUM | The percent the employee's pay is under his or her pay grade minimum (where applicable). |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| RNG_PENE | Range penetration of the employee's salary in his or her pay grade range. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |

| Column Name | Comment |
|---|---|
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |

***Table 41.28*** *SALHSUM Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the last salary snapshot for the employee. Otherwise, given a value of N. |
| ADD1 | Additional compensation items that are not included in the employee's base salary. Examples: Bonus, Commission, Cost of Living Adjustment. These items are extracted from the COMPHIST table. |
| ADD2 | Additional compensation items that are not included in the employee's base salary. Examples: Bonus, Commission, Cost of Living Adjustment. These items are extracted from the COMPHIST table. |
| ADD3 | Additional compensation items that are not included in the employee's base salary. Examples: Bonus, Commission, Cost of Living Adjustment. These items are extracted from the COMPHIST table. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| AGE_RANGE | Age-range category of employee. |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| BIRTH_DT | Employee's birth date. |
| CHGAMT | Employee's change in annual salary from the previous salary snapshot. This value could be positive, negative, or zero. |
| CHGPCT | Employee's percent change in annual salary from the previous salary snapshot. This value could be positive, negative, or zero. |
| CITY_NM | Employee's city. |

| Column Name | Comment |
|---|---|
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| CURRENCY_CD | Standard three-character ISO 4217 code used for identifying currency. For example, USD = U.S. dollar, and EUR = euro. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_EMAIL | E-mail address of the employee. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_NO | Employee's employee number. The value in this column does not have to be unique and is not required. |
| EMPLOYEE_PAID_FREQUENCY_CD | Unique code for a time frequency or time span. Defines the type of time frequency. For example, week, bi-weekly, month, year. |
| EMPLOYEE_STATUS_CD | Code that defines the employment status of an employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| FLSA_STATUS_CD | Employee Fair Labor Standards Act (FLSA) status code. Can be the same as the exempt status code. Specific to the United States. |
| FTE_RT | Full-time equivalence ratio, bounded by 0 and 1, where halftime = 0.5 and fulltime = 1. |

| Column Name | Comment |
|---|---|
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| HOURS_PER_WEEK_CNT | Number of hours per week that an employee works. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_GROUP_DESC | Description of the job group. |
| JOB_TITLE_TXT | Title for the job. |
| LENGTH_OF_SERVICE | Category range for years of service. Examples: "1+ to 3 yrs", "3+ to 5 yrs". |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| MAXIMUM_AMT | Maximum amount for the employee's pay grade. |
| MIDPOINT_AMT | Midpoint amount for the employee's pay grade. |
| MILITARY_EXPERIENCE_DT | Date that the military experience started. |
| MINIMUM_AMT | Minimum amount for the employee's pay grade. |
| MINORITY_FLG | Indicates whether the employee is in an ethnic minority classification. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| ONPAYRL | Flags whether an employee is on the payroll. Used as a factor for certain HCM measures. |
| OVER40 | A flag indicating whether the employee is 40 or older. Y = Yes, N = No. |
| PAY_FREQUENCY_CD | Frequency at which the employee is paid. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_GRADE_FREQUENCY_CD | Time frequency for the employee's pay grade. For example, Weekly, Bi-weekly, Monthly, or Yearly. |
| PAY_LEVEL_CD | The employee's pay grade. |
| PAY_LEVEL_STRUCTURE_CD | A unique code for a pay grade. |
| PAY_LEVEL_TYPE | A unique grouping for the pay grade. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSITION_CD | A code that represents a position within the company. |
| POSTAL_CD | Address postal code. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |

| Column Name | Comment |
|---|---|
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| STECLASS | A Saratoga classification into Manager, Executive, and Staff groupings. Used for certain HCM measures. |
| TOTCOMP | Total compensation for the employee on the snapshot date. This includes the additional compensation from the COMPHIST table. |
| TOTOTHER | Total additional compensation for the employee on the snapshot date. This comes from the additional compensation in the COMPHIST table. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_DT | Date of the employee salary snapshot. |

***Table 41.29*** *TERMMAST Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y. |
| ACTION_DT | The date of the employee action. |
| ACTION_MONTH | A number representing the month of the employee action. 1 = January, 2 = February, 3 = March, and so on. |
| ACTION_TYPE_CD | Code for a type of employee action. Typical employee actions are probation, performance improvement plan, and suspension. |
| ACTION_YEAR | The year of the employee action. |
| ADDRESS_LINE_1_TXT | First line of the address. |
| ADDRESS_LINE_2_TXT | Second line of the address. |
| AGE | Age of employee. |
| ANNUAL_SALARY | The employee's annualized salary. This column contains a calculated value for pay frequencies other than Yearly. |
| CITY_NM | Employee's city. |
| COUNTRY_CD | This column contains the three-character ISO 3166 standard code for a country. For example, AFG = Afghanistan and ALB = Albania. |

| Column Name | Comment |
|---|---|
| COUNTY_NM | County name. Used by the Geographic Analysis. |
| DISABILITY_FLG | Flag indicating if an employee has a disability. |
| DISCIPLINARY_ACTION_FLG | Flag indicating if an employee is on disciplinary action. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ACTION_REASON_CD | Code for a reason for an employee action. For example, "promotion" is a reason for a pay raise. |
| EMPLOYEE_ID | Business key identifying the employee. |
| EMPLOYEE_NAME | Name of employee. |
| EMPLOYEE_TYPE_CD | A unique code to indicate an employee's employment type. For example, RFT for regular full-time, RPT for regular part-time. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIRE_DT | Employee's hire date. |
| HOURLY_SALARY | The hourly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| JOB_TITLE_TXT | Title for the job. |
| MNMLEV1 | Level 1 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |

| Column Name | Comment |
|---|---|
| MNMLEV2 | Level 2 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV3 | Level 3 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV4 | Level 4 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MNMLEV5 | Level 5 of the manager name hierarchy. This is a derived field based on the child-parent relationship for the default hierarchy in the WRKGRP table. The default hierarchy is specified by the DEFAULT_HIER option in the prebuild.sas macro. |
| MONTHLY_SALARY | The monthly salary for an employee. This might be a calculated value, depending on the pay frequency for the employee. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| POSTAL_CD | Address postal code. |
| QTR | A number (1, 2, 3, or 4) representing the quarter of the year of the termination action. |
| SERVICE_START_DT | Date an employee began service with the company. This date is used to calculate an employee's years of service. |
| SERVICE_YEARS | Employee's years of service, calculated from his or her start date. |
| STATE_REGION_CD | State or region code. Examples include AK for Alaska, AL for Alabama, and AR for Arkansas. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| YEARMON | Year and month of the termination. Example, 2007M03 is a termination in March of 2007. |

*Table 41.30 TIP Table*

| Column Name | Comment |
|---|---|
| AGE | Age of employee. |
| EEO_CLASS_CD | Standard EEO classification codes. EEO1 classifications are used for corporations, and EEO4 classifications are used for government reporting at the federal, state, and local levels. |
| EMPLOYEE_ID | Business key identifying the employee. |
| ETHNICITY_CD | Code used to define the ethnicity. |
| EVALUATION_RESULT_CD | The result of an employee's evaluation. |
| EXEMPT_STATUS_CD | Indicates whether the employee meets the requirements of the Fair Labor Standards Act (FLSA) classification of "exempt." For example, "exempt" or "non-exempt." Specific to the United States. |
| GENDER_CD | Code used to specify the gender. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| INTORG_HR | The department that an employee is in. The name "INTORG_HR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MAIN, for example), then this column's name will be MAIN rather than INTORG_HR. |
| INTORG_HR1 | Level 1 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR2 | Level 2 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR3 | Level 3 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR4 | Level 4 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_HR5 | Level 5 of the departmental hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |

| Column Name | Comment |
|---|---|
| INTORG_MGR | The employee's manager (manager's employee ID). The name "INTORG_MGR" correlates to a hierarchy in the WRKGRP table (HIERARCHY_CD column). If the hierarchy has a different name (MANAGER, for example), then this column's name will be MANAGER rather than INTORG_MGR. |
| INTORG_MGR1 | Level 1 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR2 | Level 2 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR3 | Level 3 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR4 | Level 4 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| INTORG_MGR5 | Level 5 of the management hierarchy that an employee is in. This is a derived field based on the child-parent relationship for the hierarchy in the WRKGRP table. |
| JOB_CD | A code that represents a job within the company. |
| JOB_GROUP_CD | Job group code, usually representing a grouping of jobs. It is usually associated with only one EEO category. |
| PERMANENCE_CD | Permanence status of a position. For example, R = regular and T = temporary. |
| TIP_DAYS | Number of days in a position. |
| TIP_MONTHS | Number of months in a position. |
| TIP_YEARS | Number of years in a position. |
| USER_NM | Employee's user name. This contains the same value as the employee's user name populated in the User Manager plug-in in SAS Management Console. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

***Table 41.31*** *WRKGRP Table*

| Column Name | Comment |
|---|---|
| _LASTREC | Given a value of Y if this is the current record for a department or manager. Otherwise, given a value of N. |
| GRPDESC | Description of the Organization Report Group. |
| HCM_UNIQ_ID | Generic system key for the table. Unrelated to the data in the table. |
| HIERARCHY_CD | Code designating the organizational hierarchy defined in this table. Multiple organizational hierarchies can be defined in this table. |
| INTORG | Code for a department or manager. This is the child of PARENT_INTORG. |
| MANAGER_ID | Employee ID of the employee's manager. |
| MANAGER_NAME | Name of the employee's manager. |
| PARENT_INTORG | Code for a department or manager. This is the parent of INTORG. |
| VALID_FROM_DT | Valid start date for the record. |
| VALID_TO_DT | Valid end date for the record. |

# Part 5

# Reference Material

*Appendix 1*
**Object Security: List of Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 673

*Appendix 2*
**Metrics in SAS Human Capital Management** . . . . . . . . . . . . . . . . . . . . . 683

*Appendix 3*
**The Public API** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 705

*Appendix 4*
**MySQL Reserved Words** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 735

*Appendix 5*
**Configuring the SAS Environment Files** . . . . . . . . . . . . . . . . . . . . . . . . . 737

*Appendix 6*
**Default Port Usage** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 743

# *Appendix 1*
# Object Security: List of Objects

**About Object Security** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 673
**Employee Profile Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 673
**Geographical Analysis Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 674
**Organization Analysis Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 675
**General Search Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 675
**Administrator Options: Data Tab** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 676
**Administrator Objects: Customize Tab** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 678
**Administrator Objects: Security Tab** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 679
**Administrator Objects: Configuration Tab** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 681
**Home Page Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 681
**Custom Objects** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 682

## About Object Security

Object security refers to actions that the user can perform. If a user has permission for an object (via user, group, or role permissions), the user can perform the action. On the **Security** tab of the Administration application, objects are divided into groups based on where they appear in the user interface.

This appendix lists the objects that are a part of object security: For more information, see “Securing Objects” on page 67.

## Employee Profile Objects

| Object | Description |
|---|---|
| **External Actions** | Access external actions in the Employee Browser. |
| **Email All** | Send an e-mail message to listed employees. |

| Object | Description |
|---|---|
| **Find People** | Perform a quick search for employees. |
| **Hierarchy** | Select a hierarchy (from the **Hierarchy** menu) in the Employee Browser. |
| **Options** | Modify the options for an employee profile, such as the fields that are displayed after a search or the additional tables that are displayed in an employee profile. |
| **Print Profile Listing** | Print an employee listing. |
| **Print Profile** | Print employee details. |
| **ProfileListing SaveAs** | Access the **Save As** menu from an employee list. |
| **ProfileListing SaveAs Excel** | Save an employee list as a Microsoft Excel file. |
| **ProfileListing SaveAs PDF** | Save an employee list as a PDF file. |
| **ProfileView SaveAs** | Access the **Save As** menu from the employee detail view. |
| **ProfileView SaveAs PDF** | Save employee details as a PDF file. |
| **Employee Details Category Selector** | Select from categories of employee details. Without this permission, only the first category is displayed. |
| **View Employee Details** | View details for the selected employee. |
| **Open Workspace Manager** | Open the workspace from the Employee Browser. |

# Geographical Analysis Objects

| Object | Description |
|---|---|
| **Geographic Copy To** | Make a copy of a geographic analysis. |
| **Geographic DrillDown** | In map view, drill down to more specific maps. |
| **Geographic Export** | Export an employee list from a geographic analysis in table view to a Microsoft Excel file. |
| **Map View** | View data against a map of a geographic area. |
| **Geographic Options** | Modify display options for a geographic analysis. |
| **Print** | Print a geographic analysis (map view or table view). |
| **Table View** | View geographic data in a table. |

| Object | Description |
|---|---|
| **Workspace Manager** | Open the workspace from a geographic analysis. |

## Organization Analysis Objects

| Object | Description |
|---|---|
| **Analysis view** | Display an organization analysis as a hierarchical table. |
| **Copy To** | Make a copy of an organization analysis. |
| **Export** | From the **Save As** menu of an organization analysis, export a table to a Microsoft Excel file. (**Save As** must also be enabled.) |
| **Export to PDF** | From the **Save As** menu of an organization analysis, export a table to a PDF file. (**Save As** must also be enabled.) |
| **Find People** | Perform a quick search for an employee. |
| **Measures** | Select the measures that are displayed in an organization analysis. |
| **Modify** | Modify an organization analysis to simulate a reorganization. |
| **New** | Create an organization analysis. |
| **Org Options** | Modify the display options for an organization analysis. |
| **Presentation view** | View an organization analysis in presentation view. |
| **Print** | Print an organization analysis. |
| **Save As** | Access the **Save As** menu in the toolbar of an organization analysis. |
| **Scorecard** | Open an associated scorecard from an organization analysis. |

## General Search Objects

These objects control general search functionality, for performing a search or for working with the search results.

| Object | Description |
|---|---|
| **Advanced Search** | Perform an advanced general search.<br>The Home page's **Advance Search** object must also be enabled. |

| Object | Description |
|---|---|
| **Bar Chart** | Create a bar chart from general search results. |
| **BarLine Chart** | Create a bar-line chart from general search results. |
| **Search Email All** | Compose an e-mail message to the employees that are listed in the search results. |
| **Export Excel** | Export the general search results to a Microsoft Excel file. |
| **PDF File** | Export the general search results to a PDF file. |
| **Geo Map** | Create a geographic map from general search results. |
| **Search Hierarchy** | Access search information by hierarchy. |
| **History Search** | Access the search history on the search results page. |
| **Line Chart** | Create a line chart from general search results. |
| **Search Options** | Modify the columns that are displayed in the general search results. |
| **Pie Chart** | Create a pie chart from general search results. |
| **Search Print** | Print the results of a general search. |
| **Save As** | Access the **Save As** menu on the general search results page. |
| **Scatter Plot** | Create a scatter plot from general search results. |
| **Search** | Perform a general search.<br>**Search** must also be enabled for the Home page, so that the general search text box is displayed. |
| **Table Viewer** | Access the **Table View** button, to display the general search results in a table. |

## Administrator Options: Data Tab

These objects control functionality on the **Data** tab of the Administration application.

| Object | Description |
|---|---|
| **AdminDataCommon** | |
| **Add Table** | Import a table. |
| **Build Formats Catalog** | Builds the format catalog. |

| Object | Description |
|---|---|
| **SAS Log** | View the SAS log for an operation. |
| **Map Hierarchies** | Create a hierarchy mapping. |
| **New Format** | Create a format. |
| **New Metric** | Create a measure. |
| **Refresh Data Cache** | Access the **Refresh Cache** button from the **Data** tab. |
| **Run Job** | Run a job to calculate values for measures. |
| **Measures Menu — Delete** | Delete a measure. |
| **Measures Menu — Edit** | Edit a measure. |
| **Planning — Apply Changes** | Apply changes to planning attributes. |
| **Planning — SAS Code** | View planning SAS code. |
| **AdminData** | |
| **Table Columns Apply Changes** | Apply changes to table column attributes. |
| **Table Apply Changes** | Apply changes to table attributes. |
| **Table Columns Apply and Propagate Changes** | Modify column attributes and apply the changes to all tables with this column name. |
| **AdminDataContextMenu BuildInfoMap** | Create an information map from a table. |
| **TableDetails Context Menu** | View the action menu for table's column attributes. |
| **AdminDataContextMenu CopyTable** | Copy a table. |
| **Delete Column** | Delete a column. |
| **Delete Formats** | Delete a format. |
| **Delete Hierarchy** | Delete a hierarchy. |
| **AdminDataContextMenu Delete Table** | Delete a table. |
| **Format Properties** | Modify format properties. |
| **Admin Data Map Hierarchies Details** | Modify hierarchy details. |
| **AdminDataContextMenu ExportToExcel** | Export a table to Excel. |
| **Formats Context Menu** | Access the Formats action menu. |
| **Generate Symbol** | Generate a symbol for a column. |

| Object | Description |
|---|---|
| **AdminDataContextMenu GenerateSymbol** | Generate a symbol for a table. |
| **TableContextMenu** | View the action menu for each table in the list of tables. |
| **AdminDataContextMenu ViewTable** | View table data. |
| **AdminCubeIMAP** | |
| **New Cube** | Create a cube. |
| **Rebuild Information Map** | Rebuild information maps. |
| **Refresh Cube** | Refresh all cubes on the page. |
| **AdminDataTabCubeContextMenu Build CubeIMAP** | Build an information map from a cube (action menu selection). |
| **Cube Context Menu** | Open cube context menu. |
| **AdminDataTabCubeContextMenu Delete Cube** | Delete a cube (action menu selection). |
| **IMAP Menu - Delete** | Delete an information map (action menu selection). |
| **AdminDataTabCubeContextMenu Rebuild Cube** | Rebuild a cube (action menu selection). |
| **IMAP Menu - Rebuild** | Rebuild an information map (action menu selection). |
| **AdminDataTabCubeContextMenu ViewCube** | Open a cube in SAS Web Report Studio. |
| **IMAP Menu - View** | Open an information map in SAS Web Report Studio. |
| **Add Cube Dimension ButtonBar** | Create a new dimension in the New Cube wizard. |
| **Rebuild Cube ButtonBar** | Rebuild or refresh one or more cubes. |
| **AdminDataTabCubeContextMenu Cube Refresh** | Refresh a cube (action menu selection) |

## Administrator Objects: Customize Tab

| Object | Description |
|---|---|
| **ApplyChangesCategory** | Modify the fixed categories for an employee profile. |

| Object | Description |
|---|---|
| **ApplyChangesCategoryDetails** | Modify the columns for an employee profile fixed category. |
| **ApplyChangesEEP** | Modify an employee profile. |
| **ApplyChangesEditEEP** | Modify employee profile details. |
| **ApplyChangesHeaderEEP** | Modify the employee profile header settings. |
| **ApplyChangesListEEP** | Modify the employee profile list settings. |
| **ApplyChangesSearch** | Modify the employee profile search settings. |
| **ApplyChangesSearchEEP** | Modify the employee profile search settings. |
| **CopyEEP** | Copy an employee profile. |
| **DeleteEEP** | Delete an employee profile. |
| **Delete template** | Delete a template (for the Home page or the employee profile details page). |
| **Edit Geo ApplyChanges** | Modify geographic analysis defaults. |
| **Edit Org ApplyChanges** | Modify organization analysis defaults. |
| **Edit template** | Modify a template. |
| **ProfileContextMenu** | View the action menu for employee profiles. |
| **NewEmployeeProfile** | Create an employee profile. |
| **New Template** | Create a new template. |
| **RefreshCacheCustomize** | Access the **Refresh Cache** button from the **Customize** tab. |

## Administrator Objects: Security Tab

| Object | Description |
|---|---|
| **Add Filter** | Add a row-level security filter. |
| **Edit Object Permissions** | Modify object permissions. |
| **Apply Changes to Column Permissions** | Modify column permissions. |
| **Apply Changes to Custom Objects** | Modify custom object permissions. |
| **Edit Hierarchy** | Modify the hierarchical filter. |

| Object | Description |
|---|---|
| **Apply Changes to Enable Row Level Security** | Enable or disable row-level security. |
| **Column Permissions Apply and Propagate Changes** | Modify column permissions and apply those changes to all columns with that name. |
| **Column Permissions Context Menu** | View the action menu for column security. |
| **Table Filters Context Menu** | View the action menu for row-level security. |
| **Object Permissions Context Menu** | View the action menu for object security. |
| **Search User Filters Context Menu** | On the Search Users results page for row-level filters, view the action menu. |
| **Show User Filters Context Menu** | On the Show Filters page for row-level filters, view the action menu. |
| **Delete Custom Objects** | Delete custom objects that have been defined. |
| **Delete Filter** | Delete row-level security filters. |
| **Import Users** | Load the SAS_USER_EMPLOYEE table with all users who are members of the HCM Solution Users group. |
| **View Filter** | View or edit an existing row-level security filter. |
| **Refresh Security Cache** | Access the **Refresh Cache** button from the **Security** tab. |
| **Remove Column Permissions** | Delete direct column permissions for an identity. |
| **Remove Object Permissions** | Delete direct object permissions for an identity. |
| **Column Permissions Search Users** | Search for identities for adding column permissions. |
| **Object Permissions Search Users** | Search for identities for adding object permissions. |
| **Filters Search Users** | Search for identities for adding row-level filters. |
| **Show Filters** | Display all row-level filters for an identity. |
| **View Object Permissions** | View all object permissions for an identity. |
| **View Report** | In the permissions for an object, select the types of identities to view (without this permission, all identities are displayed). |

## Administrator Objects: Configuration Tab

| Object | Description |
|---|---|
| **Run Diagnostics** | Run diagnostics from the **Configuration** tab of the Administration application. |
| **Apply Changes Configuration** | Apply changes to items on the **Configuration** tab. |
| **Refresh Cache** | Access the **Refresh Cache** button from the **Configuration** tab. |

## Home Page Objects

The Home page objects control the user interface on the Home page. Many of these objects require additional object permissions, as noted.

If the link opens another Web application, that application might require specific role membership. For more information, see the documentation for those applications.

***CAUTION:***

> Be aware that even if you disable an object for a Web application, users who know the URL might still be able to open the application. The Home page objects control only the display of links to the applications.

| Object | Description |
|---|---|
| **Administration** | Open the Administration application from the Home page. |
| **Advance Search** | Access the **Advanced Search** box on the Home page.<br>The **Advanced Search** object of the General search options must be enabled.<br>See **Search** for additional functionality needed. |
| **Manage Dashboard** | Access the **Manage Dashboards** link on the Home page. |
| **Your Employee Profile** | Open the Employee Browser from the Home page. |
| **History Search** | Access the search history on the Home page.<br>See **Search** for additional functionality needed. |
| **New Geographic Analysis** | Open a new geographic analysis from a link on the Home page. |
| **New Organization Analysis** | Open a new organization analysis from a link on the Home page.<br>The Organization Analysis's **New** object must also be enabled. |
| **My Portal** | Open the main portal page from a link on the Home page. |

| Object | Description |
|---|---|
| **My Report** | From a link on the Home page, open a new report in SAS Web Report Studio (if installed).<br>This object controls only the link to SAS Web Report Studio. It does not affect a user's access to that application. |
| **New HR Scorecard** | From a link on the Home page, open a new scorecard in SAS Strategy Management. If SAS Strategy Management is not installed, the link opens a KPI project.<br>This object controls only the link. It does not affect a user's access to that application. |
| **Search** | Perform a search from the Home page.<br>The General Search's **Search** object must also be enabled. (The Home page object controls whether the general search mechanism is accessible. The General Search object controls the search functionality.) |
| **Search Assist** | Use the search assist (F12) functionality in a simple search on the Home page.<br>See **Search** for additional functionality needed. |
| **Workspace** | Open the workspace from a link on the Home page. |

## Custom Objects

If you have defined any custom objects, you can set their permissions in this section.

# *Appendix 2*
# Metrics in SAS Human Capital Management

**Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 683
**Organizational Effectiveness** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 684
**Human Resources Structure** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 686
**Compensation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 687
**Benefits** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 689
**Separations** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 691
**Staffing** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 695
**Training and Development** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 701

## Introduction

SAS Human Capital Management provides an extensive collection of predefined measures that are designed to be used with data provided by the Saratoga Institute.

*Note:* An administrator must have set up the data to use with these measures by running a job that loads the metric tables. Having the measures does not imply that you have the data.

The measures are available in the following categories:

- Organizational Effectiveness
- Human Resources Structure
- Compensation
- Benefits
- Separations
- Staffing
- Training and Development

## Organizational Effectiveness

***Table A2.1*** *Organizational Effectiveness Measures*

| Name | Description | Formula |
|---|---|---|
| Revenue Factor - Total | Dollars of unit revenue generated per total FTE | Revenue / Total FTE |
| Revenue Factor - Workforce Employees | Dollars of unit revenue generated per workforce on payroll FTE | Revenue / Workforce on Payroll FTE |
| Revenue Factor - Regular Employees | Dollars of unit revenue generated per all regular FTE employees | Revenue / Regular FTE |
| Expense Factor - Total | Dollars of unit operating expense incurred per total FTE | Operating Expense / Total FTE |
| Expense Factor - Workforce Employees | Dollars of unit operating expense incurred per workforce on payroll FTE | Operating Expense / Workforce on Payroll FTE |
| Expense Factor - Regular Employees | Dollars of unit operating expense incurred per all regular FTE employees | Operating Expense / Regular FTE |
| Income Factor - Total | Dollars of unit profit generated per total FTE | (Revenue - Operating Expense) / Total FTE |
| Income Factor - Workforce Employees | Dollars of unit profit generated per workforce on payroll FTE | (Revenue - Operating Expense) / Workforce on Payroll FTE |
| Income Factor - Regular Employees | Dollars of unit profit generated per all regular FTE employees | (Revenue - Operating Expense) / Regular FTE |
| Human Capital Value Added | Dollars of adjusted profit added per total FTE | (Revenue - (Operating Expense - (Compensation Cost (Workforce on Payroll) + Benefit Cost EPTNW))) / Total FTE |
| Human Capital ROI | Dollars of adjusted profit per each dollar spent on employee compensation and benefits | (Revenue - (Operating Expense - (Compensation Cost (Workforce on Payroll) + Benefit Cost EPTNW))) / (Compensation Cost (Workforce on Payroll) + Benefit Cost EPTNW) |
| Human Economic Value Added | Dollars of true profit (after expenses, taxes and capital costs) generated per total FTE | Net Operating Profit After Tax - (Shareholder's Equity * 0.12) / Total FTE |
| Outsourcing Expense Percent | Outsourcing costs as a percentage of total operating costs | Outsourcing Expense / Operating Expense |
| Management Ratio - Total | Average number of employees to each manager | Total Headcount / Management Headcount |

| Name | Description | Formula |
|---|---|---|
| Management Ratio - Regular Employees | Average number of regular employees to each manager | Regular Headcount / Management Headcount |
| Management Investment Factor - Total | Average dollars spent on managers per total FTE | Management Compensation Cost / Total FTE |
| Management Investment Factor - Regular Employees | Average dollars spent on managers per all regular FTE employees | Management Compensation Cost / Regular FTE |
| Average Tenure - Total | Average length of service of all active regular employees | Total Employee Tenure / Regular Employee Headcount |
| Average Tenure - Exempt | Average length of service of all active, exempt, regular employees | Exempt Employee Tenure / Exempt Regular Headcount |
| Average Tenure - Nonexempt | Average length of service of all active, nonexempt, regular employees | Nonexempt Employee Tenure / Nonexempt Regular Headcount |
| Headcount Percent - Contingent - Total | Contingent employees as a percentage of total headcount | Total Contingent Headcount / Total Headcount |
| Headcount Percent - Contingent - Regular Employees | Contingent employees as a percentage of regular employee headcount | Total Contingent Headcount / Regular Employee Headcount |
| Headcount Percent - Contingent - On Payroll | Contingent on-payroll employees as a percentage of total headcount | Contingent On Payroll Headcount / Total Headcount |
| Headcount Percent - Contingent - Off Payroll | Contingent off-payroll employees as a percentage of total headcount | Contingent Off Payroll Headcount / Total Headcount |
| FTE Percent - Contingent - On Payroll | Contingent on-payroll FTE as a percentage total FTE | Contingent On Payroll FTE / Total FTE |
| FTE Percent - Contingent - Off Payroll | Contingent off-payroll FTE as a percentage total FTE | Contingent Off Payroll FTE / Total FTE |
| FTE Percent - Management | Management FTE as a percentage total FTE | Management FTE / Total FTE |
| FTE Percent - Professionals | Professional FTE as a percentage total FTE | Professionals FTE / Total FTE |
| FTE Percent - Sales | Sales FTE as a percentage total FTE | Sales FTE / Total FTE |
| FTE Percent - Office & Clerical | Office & Clerical FTE as a percentage total FTE | Office & Clerical FTE / Total FTE |
| FTE Percent - Operatives | Operatives FTE as a percentage total FTE | Operatives FTE / Total FTE |

## Human Resources Structure

***Table A2.2*** *Human Resources Structure Measures*

| Name | Description | Formula |
|---|---|---|
| HR Expense Percent | Internal and external HR expenses as a percentage of operating expense | HR Expense / Operating Expense |
| HR FTE Ratio - Total | Total number of FTE employees that each Human Resource FTE supports | Total FTE / Total HR FTE |
| HR FTE Ratio - Regular Employees | All regular employees that each Human Resource FTE supports | Regular FTE / Total HR FTE |
| HR Exempt Percent - Total | Exempt HR FTE as a percentage of total HR FTE | HR Exempt FTE / Total HR FTE |
| HR Exempt Percent - Regular Employees | Exempt HR FTE as a percentage of regular HR FTE | HR Exempt FTE / Regular HR FTE |
| HR FTE Investment Factor - Total | Dollars spent on Human Resource functions per total FTE | HR Expense / Total FTE |
| HR FTE Investment Factor - Regular Employees | Dollars spent on Human Resource functions per regular FTE | HR Expense / Regular FTE |
| HR Headcount Investment Factor - Total | Dollars spent on Human Resource functions per headcount employee | HR Expense / Total Headcount |
| HR Headcount Investment Factor- Regular Employees | Dollars spent on Human Resource functions per regular employee headcount | HR Expense / Regular Employee Headcount |
| HR Outsourcing Percent | HR outsourcing expenses as a percentage of total HR expense | HR Outsourcing Cost / HR Expense |
| HR Consulting Percent | HR consulting services expenses as a percentage of HR expense | HR Consulting Cost / HR Expense |
| HR Compensation Expense Percent | HR compensation costs as a percentage of total operating expenses | HR Compensation Cost / Operating Expense |
| HR Employee Cost Factor | Average compensation paid to regular HR FTE | HR Compensation Cost / Regular HR FTE |
| HR Total Employee Cost Factor | Average compensation paid to regular HR employees including benefit costs | HR Compensation Cost * (1 + Benefits Comp Percent / Compensation Cost (Workforce of Payroll)) / Regular HR FTE |
| HR Separation Rate - Total | HR employees who terminated as a percentage of total HR headcount | Total HR Separations / Total HR Headcount |

| Name | Description | Formula |
|---|---|---|
| HR Separation Rate - Exempt | HR exempt employees who terminated as a percentage of total HR headcount | Exempt HR Separations / Exempt HR Headcount |
| HR Separation Rate - Nonexempt | HR nonexempt employees who terminated as a percentage of total HR headcount | Nonexempt HR Separations / Nonexempt HR Headcount |
| HR Structure Breakdown - Administrative | HR FTE in Administrative functions as a percentage of total HR FTE | HR Administrative FTE / Total HR FTE |
| HR Structure Breakdown - Benefits | HR FTE in Benefit functions as a percentage of total HR FTE | HR Benefits FTE / Total HR FTE |
| HR Structure Breakdown - Compensation | HR FTE in Compensation functions as a percentage of total HR FTE | HR Compensation FTE / Total HR FTE |
| HR Structure Breakdown - Employee Relations | HR FTE in Employee Relations functions as a percentage of total HR FTE | HR Employee Relations FTE / Total HR FTE |
| HR Structure Breakdown - HRIS | HR FTE in HRIS functions as a percentage of total HR FTE | HR HRIS FTE / Total HR FTE |
| HR Structure Breakdown - HR Management | HR FTE in HR Management functions as a percentage of total HR FTE | HR Management FTE / Total HR FTE |
| HR Structure Breakdown - Legal | HR FTE in Legal functions as a percentage of total HR FTE | HR Legal FTE / Total HR FTE |
| HR Structure Breakdown - Staffing | HR FTE in Staffing functions as a percentage of total HR FTE | HR Staffing FTE / Total HR FTE |

## Compensation

*Table A2.3 Compensation Measures*

| Name | Description | Formula |
|---|---|---|
| Compensation Revenue Percent - Workforce Employees | Workforce on-payroll compensation cost as a percentage of revenue | Compensation Cost (Workforce on Payroll) / Revenue |
| Compensation Revenue Percent - Regular Employees | Regular employee compensation cost as a percentage of revenue | Compensation Cost (Regular Employees) / Revenue |
| Total Compensation Revenue Percent | Workforce on-payroll compensation and benefit cost, excluding payments for time not worked (EPTNW), as a percentage of revenue | (Compensation Cost (Workforce on Payroll) + Benefit Cost EPTNW) / Revenue |
| Total Labor Cost Revenue Percent | Total compensation and benefit costs, excluding payments for time not worked (EPTNW), as a percentage of revenue | (Compensation Cost (Total) + Benefit Cost EPTNW) / Revenue |

| Name | Description | Formula |
|---|---|---|
| Compensation Expense Percent - Workforce Employees | Workforce on-payroll compensation cost as a percentage of operating expense | Compensation Cost (Workforce on Payroll) / Operating Expense |
| Compensation Expense Percent - Regular Employees | Regular employee compensation cost as a percentage of operating expense | Compensation Cost (Regular Employees) / Operating Expense |
| Total Compensation Expense Percent | Workforce on-payroll compensation and benefit cost, excluding payments for time not worked (EPTNW), as a percentage of operating expense | (Compensation Cost (Workforce on Payroll) + Benefit Cost EPTNW) / Operating Expense |
| Total Labor Cost Expense Percent | Total compensation and benefit costs, excluding payments for time not worked (EPTNW) as a percentage of operating expense | (Compensation Cost (Total) + Benefit Cost EPTNW) / Operating Expense |
| Compensation Percent - Executive | Executive compensation cost as a percentage of workforce on payroll compensation cost | Compensation Cost (Executive) / Compensation Cost (Workforce on Payroll) |
| Compensation Percent - Staff | Staff compensation cost as a percentage of workforce on payroll compensation cost | Compensation Cost (Staff) / Compensation Cost (Workforce on Payroll) |
| Compensation Percent - Variable | Variable compensation cost as a percentage of workforce on-payroll compensation cost | Variable Compensation Cost / Compensation Cost (Workforce on Payroll) |
| Compensation Percent - Contingent - Total | Total contingent compensation cost as a percentage of workforce on-payroll compensation cost | Contingent Cost / Compensation Cost (Workforce on Payroll) |
| Compensation Percent - Contingent - On Payroll | Contingent on-payroll compensation cost as a percentage of workforce on-payroll compensation cost | Contingent On Payroll Cost / Compensation Cost (Workforce on Payroll) |
| Compensation Percent - Contingent - Off Payroll | Contingent off-payroll compensation cost as a percentage of workforce on-payroll compensation cost | Contingent Off Payroll Cost / Compensation Cost (Workforce on Payroll) |
| Contingent Cost Revenue Percent - Total | Costs of contingent workers as a percentage of revenue generated | Contingent Cost / Revenue |
| Contingent Cost Revenue Percent - On Payroll | Costs of contingent on-payroll workers as a percentage of revenue generated | Contingent On Payroll Cost / Revenue |
| Contingent Cost Revenue Percent - Off Payroll | Costs of contingent off-payroll workers as a percentage of revenue generated | Contingent Off Payroll Cost / Revenue |
| Contingent Cost Expense Percent - Total | Costs of contingent workers as a percentage of operating expenses | Contingent Cost / Operating Expense |
| Contingent Cost Expense Percent - On Payroll | Costs of on-payroll contingent on-payroll workers as a percentage of operating expenses | Contingent On Payroll Cost / Operating Expense |

| Name | Description | Formula |
|---|---|---|
| Contingent Cost Expense Percent - Off Payroll | Costs of contingent off-payroll workers as a percentage of operating expenses | Contingent Off Payroll Cost / Operating Expense |
| Employee Cost Factor - Regular Employees | Average compensation paid to each regular FTE employee | Compensation Cost (Regular Employee) / Regular Employee FTE |
| Employee Cost Factor - Workforce | Average compensation paid to each workforce on payroll FTE employee | Compensation Cost (Workforce on Payroll) / Workforce on Payroll FTE |
| Employee Cost Factor - Executive | Average compensation paid to executive level staff (vice-president level and above) | Compensation Cost (Executive) / Executive FTE |
| Employee Cost Factor - Manager | Average compensation paid to managers | Compensation Cost (Manager) / Manager FTE |
| Employee Cost Factor - Staff | Average compensation paid to individual contributor employees, not including executives, managers and contingents | Compensation Cost (Staff) / Staff FTE |
| Total Employee Cost Factor | Average compensation and benefit costs, excluding payments for time not worked (EPTNW), per regular employee FTE | (Compensation Cost (Regular Employee) + Benefits Cost EPTNW) / Regular Employee FTE |
| Total Labor Cost Factor | Average compensation and benefit costs, excluding payments for time not worked (EPTNW), per all FTE, including contingent off-payroll employees | (Compensation Cost (Total) + Benefits EPTNW) / Total FTE |
| Contingent Cost Factor - Total | Average dollars paid to each contingent worker | Contingent Cost / Total Contingent FTE |
| Contingent Cost Factor - On Payroll | Average dollars paid to each contingent on-payroll worker | Contingent On Payroll Cost / Contingent On Payroll FTE |
| Contingent Cost Factor - Off Payroll | Average dollars paid to each contingent off-payroll worker | Contingent Off Payroll Cost / Contingent Off Payroll FTE |

## Benefits

***Table A2.4*** *Benefits Measures*

| Name | Description | Formula |
|---|---|---|
| Benefit Revenue Percent | Employee benefit cost as a percentage of revenue | Benefit Cost / Revenue |
| Benefit Expense Percent | Employee benefit cost as a percentage of operating expense | Benefit Cost / Operating Expense |

| Name | Description | Formula |
|---|---|---|
| Benefit Compensation Percent - Total | Employee benefit cost as a percentage of workforce on-payroll compensation cost | Benefit Cost EPTNW / Compensation Cost (Workforce on Payroll) |
| Benefit Compensation Percent - Regular Employees | Employee benefit cost as a percentage of regular employee compensation cost | Benefit Cost EPTNW / Comp. Cost (Regular Employees) |
| Benefit Factor | Average cost of benefits per workforce on-payroll employee | Benefit Cost / Workforce on Payroll Headcount |
| Benefit Factor - Regular Employees | Average cost of benefits per regular employee headcount | Benefit Cost / Regular Employee Headcount |
| Healthcare Factor | Medical and healthcare benefit cost per covered employee | Medical & Medically Related Benefit Payments / Employees and Retirees participating in Health Program |
| Workers' Compensation Factor | Workers' compensation cost per workforce on-payroll employee | Workers' Compensation Cost / Workforce on Payroll Headcount |
| Workers' Compensation Factor - Regular Employees | Workers' compensation cost per covered regular employee | Workers' Compensation Cost / Regular Employee Headcount |
| Benefit Cost Breakdown - Legally Required Payments | Legally required payments as a percentage of total benefit cost | Legally Required Payments / Benefit Cost |
| Benefit Cost Breakdown - Retirement & Savings Plan Payments | Retirement and savings plan payments as a percentage of total benefit cost | Retirement & Savings Plan Payments / Benefit Cost |
| Benefit Cost Breakdown - Life Insurance & Death Benefit Payments | Life insurance and death benefit payments as a percentage of total benefit cost | Life Insurance & Death Benefit Payments / Benefit Cost |
| Benefit Cost Breakdown - Medical & Medically Related Benefit Payments | Medical and medically related benefit cost as a percentage of total benefit cost | Medical & Medically Related Benefit Payments / Benefit Cost |
| Benefit Cost Breakdown - Payments for Time Not Worked | Payments for time not worked as a percentage of total benefit cost | Payments for Time Not Worked / Benefit Cost |
| Benefit Cost Breakdown - Miscellaneous Benefit Payments | Miscellaneous benefit payments as a percentage of total benefit cost | Miscellaneous Benefit Payments / Benefit Cost |

# Separations

***Table A2.5*** *Separations Measures*

| Name | Description | Formula |
|---|---|---|
| Separation Rate - Total | Total voluntary and involuntary terminations as a percentage of employee headcount | Total Separations / Regular Employee Headcount |
| Separation Rate - Exempt | Total voluntary and involuntary exempt terminations as a percentage of employee headcount | Total Exempt Separations / Exempt Regular Headcount |
| Separation Rate - Nonexempt | Total voluntary and involuntary nonexempt terminations as a percentage of employee headcount | Total Nonexempt Separations / Nonexempt Regular Headcount |
| Voluntary Separation Rate - Total | Voluntary terminations as a percentage of employee headcount | Total Voluntary Separations / Regular Employee Headcount |
| Voluntary Separation Rate - Exempt | Voluntary exempt terminations as a percentage of employee headcount | Exempt Voluntary Separations / Exempt Regular Headcount |
| Voluntary Separation Rate - Nonexempt | Voluntary nonexempt terminations as a percentage of employee headcount | Nonexempt Voluntary Separations / Nonexempt Regular Headcount |
| Involuntary Separation Rate - Total | Involuntary terminations as a percentage of employee headcount | Total Involuntary Separations / Regular Employee Headcount |
| Involuntary Separation Rate - Exempt | Involuntary exempt terminations as a percentage of employee headcount | Exempt Involuntary Separations / Exempt Regular Headcount |
| Involuntary Separation Rate - Nonexempt | Involuntary nonexempt terminations as a percentage of employee headcount | Nonexempt Involuntary Separations / Nonexempt Regular Headcount |
| Voluntary Separations by LOS - 0 to 1 Year | Voluntary separations with less than one year of service as a percentage of total voluntary separations | Voluntary Separations - 0 to 1 Yr / Total Voluntary Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Exempt | Voluntary exempt separations with less than one year of service as a percentage of exempt voluntary separations | Exempt Voluntary Separations - 0 to 1 Yr / Exempt Voluntary Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Nonexempt | Voluntary nonexempt separations with less than one year of service as a percentage of nonexempt voluntary separations | Nonexempt Voluntary Separations - 0 to 1 Yr / Nonexempt Vol Separations |
| Voluntary Separations by LOS - 1+ to 3 Years | Voluntary separations with one to three years of service as a percentage of total voluntary separations | Voluntary Separations - 1+ to 3 Yrs / Total Voluntary Separations |

| Name | Description | Formula |
|---|---|---|
| Voluntary Separations by LOS - 1+ to 3 Years - Exempt | Voluntary exempt separations with one to three years of service as a percentage of exempt voluntary separations | Exempt Voluntary Separations - 1+ to 3 Yrs / Exempt Voluntary Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Nonexempt | Voluntary nonexempt separations with one to three years of service as a percentage of nonexempt voluntary separations | Nonexempt Voluntary Separations - 1+ to 3 Yrs / Nonexempt Vol Separations |
| Voluntary Separations by LOS - 3+ to 5 Years | Voluntary separations with three to five years of service as a percentage of total voluntary separations | Voluntary Separations - 3+ to 5 Yrs / Total Voluntary Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Exempt | Voluntary exempt separations with three to five years of service as a percentage of exempt voluntary separations | Exempt Voluntary Separations - 3+ to 5 Yrs / Exempt Voluntary Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Nonexempt | Voluntary nonexempt separations with three to five years of service as a percentage of nonexempt voluntary separations | Nonexempt Voluntary Separations - 3+ to 5 Yrs / Nonexempt Vol Separations |
| Voluntary Separations by LOS - 5+ to 10 Years | Voluntary separations with five to ten years of service as a percentage of total voluntary separations | Voluntary Separations - 5+ to 10 Yrs / Total Voluntary Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Exempt | Voluntary exempt separations with five to ten years of service as a percentage of exempt voluntary separations | Exempt Voluntary Separations - 5+ to 10 Yrs / Exempt Voluntary Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Nonexempt | Voluntary nonexempt separations with five to ten years of service as a percentage of nonexempt voluntary separations | Nonexempt Voluntary Separations - 5+ to 10 Yrs / Nonexempt Vol Separations |
| Voluntary Separations by LOS - 10+ Years | Voluntary separations with more than ten years of service as a percentage of total voluntary separations | Voluntary Separations - 10+ Yrs / Total Voluntary Separations |
| Voluntary Separations by LOS - 10+ Years - Exempt | Voluntary exempt separations with more than ten years of service as a percentage of exempt voluntary separations | Exempt Voluntary Separations - 10+ Yrs / Exempt Voluntary Separations |
| Voluntary Separations by LOS - 10+ Years - Nonexempt | Voluntary nonexempt separations with more than ten years of service as a percentage of nonexempt voluntary separations | Nonexempt Voluntary Separations - 10+ Yrs / Nonexempt Vol Separations |
| Separation Rate - Management | Total voluntary and involuntary management terminations as a percentage of management employee headcount | Total Management Separations / Management Headcount |
| Separation Rate - Professionals | Total voluntary and involuntary professional terminations as a percentage of professional employee headcount | Total Professionals Separations / Professionals Headcount |
| Separation Rate - Sales | Total voluntary and involuntary sales terminations as a percentage of sales employee headcount | Total Sales Separations / Sales Headcount |

| **Name** | **Description** | **Formula** |
|---|---|---|
| Separation Rate - Office & Clerical | Total voluntary and involuntary office & clerical terminations as a percentage of office & clerical employee headcount | Total Office & Clerical Separations / Office & Clerical Headcount |
| Separation Rate - Operatives | Total voluntary and involuntary operative terminations as a percentage of operative employee headcount | Total Operatives Separations / Operatives Headcount |
| Voluntary Separation Rate - Management | Voluntary management terminations as a percentage of management employee headcount | Management Voluntary Separations / Management Headcount |
| Voluntary Separation Rate - Professionals | Voluntary professional terminations as a percentage of professional employee headcount | Professionals Voluntary Separations / Professionals Headcount |
| Voluntary Separation Rate - Sales | Voluntary sales terminations as a percentage of sales employee headcount | Sales Voluntary Separations / Sales Headcount |
| Voluntary Separation Rate - Office & Clerical | Voluntary office & clerical terminations as a percentage of office & clerical employee headcount | Office & Clerical Voluntary Separations / Office & Clerical Headcount |
| Voluntary Separation Rate - Operatives | Voluntary operative terminations as a percentage of operative employee headcount | Operatives Voluntary Separations / Operatives Headcount |
| Involuntary Separation Rate - Management | Involuntary management terminations as a percentage of management employee headcount | Management Involuntary Separations / Management Headcount |
| Involuntary Separation Rate - Professionals | Involuntary professional terminations as a percentage of professional employee headcount | Professionals Involuntary Separations / Professionals Headcount |
| Involuntary Separation Rate - Sales | Involuntary sales terminations as a percentage of sales employee headcount | Sales Involuntary Separations / Sales Headcount |
| Involuntary Separation Rate - Office & Clerical | Involuntary office & clerical terminations as a percentage of office & clerical employee headcount | Office & Clerical Involuntary Separations / Office & Clerical Headcount |
| Involuntary Separation Rate - Operatives | Involuntary operative terminations as a percentage of operative employee headcount | Operatives Involuntary Separations / Operatives Headcount |
| Voluntary Separations by LOS - 0 to 1 Year - Management | Voluntary management terminations with less than one year of service as a percentage of management voluntary separations | Management Vol Separations - 0 to 1 Yr / Management Vol Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Management | Voluntary management terminations with one to three years of service as a percentage of management voluntary separations | Management Vol Separations - 1+ to 3 Yrs / Management Vol Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Management | Voluntary management terminations with three to five years of service as a percentage of management voluntary separations | Management Vol Separations - 3+ to 5 Yrs / Management Vol Separations |

| Name | Description | Formula |
|---|---|---|
| Voluntary Separations by LOS - 5+ to 10 Years - Management | Voluntary management terminations with five to ten years of service as a percentage of management voluntary separations | Management Vol Separations - 5+ to 10 Yrs / Management Vol Separations |
| Voluntary Separations by LOS - 10+ Years - Management | Voluntary management terminations with more than ten years of service as a percentage of management voluntary separations | Management Vol Separations - 10+ Yrs / Management Vol Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Professionals | Voluntary professionals terminations with less than one year of service as a percentage of professional voluntary separations | Professionals Vol Separations - 0 to 1 Yr / Professionals Vol Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Professionals | Voluntary professionals terminations with one to three years of service as a percentage of professional voluntary separations | Professionals Vol Separations - 1+ to 3 Yrs / Professionals Vol Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Professionals | Voluntary professionals terminations with three to five years of service as a percentage of professional voluntary separations | Professionals Vol Separations - 3+ to 5 Yrs / Professionals Vol Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Professionals | Voluntary professionals terminations with five to ten years of service as a percentage of professional voluntary separations | Professionals Vol Separations - 5+ to 10 Yrs / Professionals Vol Separations |
| Voluntary Separations by LOS - 10+ Years - Professionals | Voluntary professionals terminations with more than years of service as a percentage of professional voluntary separations | Professionals Vol Separations - 10+ Yrs / Professionals Vol Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Sales | Voluntary sales terminations with less than one year of service as a percentage of sales voluntary separations | Sales Voluntary Separations - 0 to 1 Yr / Sales Voluntary Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Sales | Voluntary sales terminations with one to three years of service as a percentage of sales voluntary separations | Sales Voluntary Separations - 1+ to 3 Yrs / Sales Voluntary Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Sales | Voluntary sales terminations with three to five years of service as a percentage of sales voluntary separations | Sales Voluntary Separations - 3+ to 5 Yrs / Sales Voluntary Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Sales | Voluntary sales terminations with five to ten years of service as a percentage of sales voluntary separations | Sales Voluntary Separations - 5+ to 10 Yrs / Sales Voluntary Separations |
| Voluntary Separations by LOS - 10+ Years - Sales | Voluntary sales terminations with more than ten years of service as a percentage of ales voluntary separations | Sales Voluntary Separations - 10+ Yrs / Sales Voluntary Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Office & Clerical | Voluntary office & clerical terminations with less than one year of service as a percentage of office & clerical voluntary separations | Office & Clerical Voluntary Separations - 0 to 1 Yr / Office & Clerical Voluntary Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Office & Clerical | Voluntary office & clerical terminations with one to three years of service as a percentage of office & clerical voluntary separations | Office & Clerical Voluntary Separations - 1+ to 3 Yrs / Office & Clerical Voluntary Separations |

| Name | Description | Formula |
|---|---|---|
| Voluntary Separations by LOS - 3+ to 5 Years - Office & Clerical | Voluntary office & clerical terminations with three to five years of service as a percentage of office & clerical voluntary separations | Office & Clerical Voluntary Separations - 3+ to 5 Yrs / Office & Clerical Voluntary Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Office & Clerical | Voluntary office & clerical terminations with five to ten years of service as a percentage of office & clerical voluntary separations | Office & Clerical Voluntary Separations - 5+ to 10 Yrs / Office & Clerical Voluntary Separations |
| Voluntary Separations by LOS - 10+ Years - Office & Clerical | Voluntary office & clerical terminations with more than ten years of service as a percentage of office & clerical voluntary separations | Office & Clerical Voluntary Separations - 10+ Yrs / Office & Clerical Voluntary Separations |
| Voluntary Separations by LOS - 0 to 1 Year - Operatives | Voluntary operatives' terminations with less than one year of service as a percentage of operatives voluntary separations | Operatives Vol Separations - 0 to 1 Yr / Operatives Vol Separations |
| Voluntary Separations by LOS - 1+ to 3 Years - Operatives | Voluntary operatives' terminations with one to three years of service as a percentage of operatives voluntary separations | Operatives Vol Separations - 1+ to 3 Yrs / Operatives Vol Separations |
| Voluntary Separations by LOS - 3+ to 5 Years - Operatives | Voluntary operatives' terminations with three to five years of service as a percentage of operatives voluntary separations | Operatives Vol Separations - 3+ to 5 Yrs / Operatives Vol Separations |
| Voluntary Separations by LOS - 5+ to 10 Years - Operatives | Voluntary operatives' terminations with five to ten years of service as a percentage of operatives voluntary separations | Operatives Vol Separations - 5+ to 10 Yrs / Operatives Vol Separations |
| Voluntary Separations by LOS - 10+ Years - Operatives | Voluntary operatives' terminations with more than years of service as a percentage of operatives voluntary separations | Operatives Vol Separations - 10+ Yrs / Operatives Vol Separations |

## Staffing

***Table A2.6*** *Staffing Measures*

| Name | Description | Formula |
|---|---|---|
| Accession Rate - Total | All hires as a percentage of regular employee headcount | Total Hires / Regular Employee Headcount |
| Accession Rate - Exempt | All exempt hires as a percentage of exempt regular employee headcount | Exempt Hires / Exempt Regular Headcount |
| Accession Rate - Nonexempt | All nonexempt hires as a percentage of nonexempt regular employee headcount | Nonexempt Hires / Nonexempt Regular Headcount |
| Accession Rate - External - Total | External new hire employees as a percentage of regular employee headcount | External Hires / Regular Employee Headcount |

| Name | Description | Formula |
|---|---|---|
| Accession Rate - External - Exempt | External exempt new hire employees as a percentage of exempt regular employee headcount | Exempt External Hires / Exempt Regular Headcount |
| Accession Rate - External - Nonexempt | External nonexempt new hire employees as a percentage of nonexempt regular employee headcount | Nonexempt External Hires / Nonexempt Regular Headcount |
| Accession Rate - Internal - Total | Internal new hire employees as a percentage of regular employee headcount | Internal Hires / Regular Employee Headcount |
| Accession Rate - Internal - Exempt | Internal exempt new hire employees as a percentage of exempt regular employee headcount | Exempt Internal Hires / Exempt Regular Headcount |
| Accession Rate - Internal - Nonexempt | Internal nonexempt new hire employees as a percentage of nonexempt regular employee headcount | Nonexempt Internal Hires / Nonexempt Regular Headcount |
| Accession Rate - College - Total | All college hires as a percentage of regular employee headcount | College Hires / Regular Employee Headcount |
| Add Rate - Total | Employees hired to fill new positions as a percentage of regular employee headcount | Total Add Hires / Regular Employee Headcount |
| Add Rate - Exempt | Exempt employees hired to fill new positions as a percentage of exempt regular headcount | Exempt Add Hires / Exempt Regular Headcount |
| Add Rate - Nonexempt | Nonexempt employees hired to fill new positions as a percentage of nonexempt regular headcount | Nonexempt Add Hires / Nonexempt Regular Headcount |
| Add Rate - External - Total | External employees hired to new positions as a percentage of regular employee headcount | External Add Hires / Regular Employee Headcount |
| Add Rate - External - Exempt | External exempt employees hired to new positions as a percentage of exempt regular employee headcount | Exempt External Add Hires / Exempt Regular Headcount |
| Add Rate - External - Nonexempt | External nonexempt employees hired to new positions as a percentage of nonexempt regular employee headcount | Nonexempt External Add Hires / Nonexempt Regular Headcount |
| Add Rate - Internal - Total | Internal employees hired to new positions as a percentage of regular employee headcount | Internal Add Hires / Regular Employee Headcount |
| Add Rate - Internal - Exempt | Internal exempt employees hired to new positions as a percentage of exempt regular employee headcount | Exempt Internal Add Hires / Exempt Regular Headcount |
| Add Rate - Internal - Nonexempt | Internal nonexempt employees hired to new positions as a percentage of nonexempt regular employee headcount | Add Rate - Internal - Nonexempt |

| Name | Description | Formula |
|---|---|---|
| Replacement Rate - Total | Employees hired to fill existing positions as a percentage of regular employee headcount | Total Replacement Hires / Regular Employee Headcount |
| Replacement Rate - Exempt | Exempt employees hired to fill existing positions as a percentage of exempt regular employee headcount | Exempt Replacement Hires / Exempt Regular Headcount |
| Replacement Rate - Nonexempt | Nonexempt employees hired to fill existing positions as a percentage of nonexempt regular employee headcount | Nonexempt Replacement Hires / Nonexempt Regular Headcount |
| Replacement Rate - External - Total | External employees hired to fill existing positions as a percentage of regular employee headcount | External Replacement Hires / Regular Employee Headcount |
| Replacement Rate - External - Exempt | External exempt employees hired to fill existing positions as a percentage of exempt regular employee headcount | Exempt External Replacement Hires / Exempt Regular Headcount |
| Replacement Rate - External - Nonexempt | External nonexempt employees hired to fill existing positions as a percentage of nonexempt regular employee headcount | Nonexempt External Replacement Hires / Nonexempt Regular Headcount |
| Replacement Rate - Internal - Total | Internal employees hired to fill existing positions as a percentage of regular employee headcount | Internal Replacement Hires / Regular Employee Headcount |
| Replacement Rate - Internal - Exempt | Internal exempt employees hired to fill existing positions as a percentage of exempt regular employee headcount | Exempt Internal Replacement Hires / Exempt Regular Headcount |
| Replacement Rate - Internal - Nonexempt | Internal nonexempt employees hired to fill existing positions as a percentage of nonexempt regular employee headcount | Nonexempt Internal Replacement Hires / Nonexempt Regular Headcount |
| Career Path Ratio - Total | Promotions as a percentage of all movement within the organization | Total Promotions / (Total Promotions + Total Transfers) |
| Career Path Ratio - Exempt | Exempt promotions as a percentage of exempt movement within the organization | Exempt Promotions / (Exempt Promotions + Exempt Transfers) |
| Career Path Ratio - Nonexempt | Nonexempt promotions as a percentage of nonexempt movement within the organization | Nonexempt Promotions / (Nonexempt Promotions + Nonexempt Transfers) |
| Cost per Hire - Total | Average dollars spent on hiring costs per employee hired | (Total Hiring Costs * 1.1 Factor) / Total Hires |
| Cost Per Hire - Exempt | Average dollars spent on exempt employee hiring costs per exempt employee hired | (Exempt Hiring Costs * 1.1 Factor) / Exempt Hires |
| Cost Per Hire - Nonexempt | Average dollars spent on nonexempt employee hiring costs per nonexempt employee hired | (Nonexempt Hiring Costs * 1.1 Factor) / Nonexempt Hires |

| Name | Description | Formula |
|---|---|---|
| Cost Per Hire - External - Total | Average dollars spent on external employee hiring costs per external employee hired | (External Hiring Costs * 1.1 Factor) / External Hires |
| Cost Per Hire - External - Exempt | Average dollars spent on exempt external employee hiring costs per exempt external employee hired | (Exempt External Hiring Costs * 1.1 Factor) / Exempt External Hires |
| Cost Per Hire - External - Nonexempt | Average dollars spent on nonexempt external employee hiring costs per nonexempt external employee hired | (Nonexempt External Hiring Costs * 1.1 Factor) / Nonexempt External Hires |
| Cost Per Hire - Internal - Total | Average dollars spent on internal employee hiring costs per internal employee hired | (Internal Hiring Costs * 1.1 Factor) / Internal Hires |
| Cost Per Hire - Internal - Exempt | Average dollars spent on exempt internal employee hiring costs per exempt internal employee hired | (Exempt Internal Hiring Costs * 1.1 Factor) / Exempt Internal Hires |
| Cost Per Hire - Internal - Nonexempt | Average dollars spent on nonexempt internal employee hiring costs per nonexempt internal employee hired | (Nonexempt Internal Hiring Costs * 1.1 Factor) / Nonexempt Internal Hires |
| Cost Per Hire - College - Total | Average dollars spent on college employee hiring costs per college employee hired | (College Hiring Costs * 1.1 Factor) / College Hires |
| Cost Per Hire - Advertising | Advertising costs as a percentage of total new hire cost | Cost Per Hire - Advertising |
| Cost Per Hire - Agency | Agency costs as a percentage of total new hire cost | Agency Hiring Costs / Total Hiring Costs |
| Cost Per Hire - Referral Bonuses | Referral bonuses costs as a percentage of total new hire cost | Referral Bonuses Hiring Costs / Total Hiring Costs |
| Cost Per Hire - Travel | Travel costs as a percentage of total new hire cost | Travel Hiring Costs / Total Hiring Costs |
| Cost Per Hire - Relocation | Relocation costs as a percentage of total new hire cost | Relocation Hiring Costs / Total Hiring Costs |
| Cost Per Hire - Recruiter | HR recruiter costs as a percentage of total new hire cost | Recruiter Hiring Costs / Total Hiring Costs |
| Cost Per Hire - External - Advertising | External advertising costs as a percentage of total external new hire cost | External Advertising Hiring Costs / External Hiring Costs |
| Cost Per Hire - External - Agency | External agency costs as a percentage of total external new hire cost | External Agency Hiring Costs / External Hiring Costs |
| Cost Per Hire - External - Referral Bonuses | External referral bonuses costs as a percentage of total external new hire cost | External Referral Bonuses Hiring Costs / External Hiring Costs |
| Cost Per Hire - External - Travel | External travel costs as a percentage of total external new hire cost | External Travel Hiring Costs / External Hiring Costs |

| Name | Description | Formula |
|---|---|---|
| Cost Per Hire - External - Relocation | External relocation costs as a percentage of total external new hire cost | External Relocation Hiring Costs / External Hiring Costs |
| Cost Per Hire - External - Recruiter | External HR recruiter costs as a percentage of total external new hire cost | External Recruiter Hiring Costs / External Hiring Costs |
| Cost Per Hire - Internal - Advertising | Internal advertising costs as a percentage of total internal new hire cost | Internal Advertising Hiring Costs / Internal Hiring Costs |
| Cost Per Hire - Internal - Travel | Internal travel costs as a percentage of total internal new hire cost | Internal Travel Hiring Costs / Internal Hiring Costs |
| Cost Per Hire - Internal - Relocation | Internal relocation costs as a percentage of total internal new hire cost | Internal Relocation Hiring Costs / Internal Hiring Costs |
| Cost Per Hire - Internal - Recruiter | Internal HR recruiter costs as a percentage of total internal new hire cost | Internal Recruiter Hiring Costs / Internal Hiring Costs |
| Time to Fill - Total | Average number of calendar days from requisition date to offer acceptance per hire | Total Days to Fill / Total Hires |
| Time to Fill - Exempt | Average number of calendar days from requisition date to offer acceptance per exempt hire | Exempt Days to Fill / Exempt Hires |
| Time to Fill - Nonexempt | Average number of calendar days from requisition date to offer acceptance per nonexempt hire | Nonexempt Days to Fill / Nonexempt Hires |
| Time to Fill - External - Total | Number of calendar days from requisition date to offer acceptance per new external hire | External Days to Fill / External Hires |
| Time to Fill - External - Exempt | Number of calendar days from requisition date to offer acceptance per new external exempt hire | Exempt External Days to Fill / Exempt External Hires |
| Time to Fill - External - Nonexempt | Number of calendar days from requisition date to offer acceptance per new external nonexempt hire | Nonexempt External Days to Fill / Nonexempt External Hires |
| Time to Fill - Internal - Total | Number of calendar days from requisition date to offer acceptance per new internal hire | Internal Days to Fill / Internal Hires |
| Time to Fill - Internal - Exempt | Number of calendar days from requisition date to offer acceptance per new internal exempt hire | Exempt Internal Days to Fill / Exempt Internal Hires |
| Time to Fill - Internal - Nonexempt | Number of calendar days from requisition date to offer acceptance per new internal nonexempt hire | Nonexempt Internal Days to Fill / Nonexempt Internal Hires |
| Time to Start - Total | Average number of calendar days from requisition date to employee start date per hire | Total Days to Start / Total Hires |

| Name | Description | Formula |
|---|---|---|
| Time to Start - Exempt | Average number of calendar days from requisition date to employee start date per exempt hire | Exempt Days to Start / Exempt Hires |
| Time to Start - Nonexempt | Average number of calendar days from requisition date to employee start date per nonexempt hire | Nonexempt Days to Start / Nonexempt Hires |
| Time to Start - External - Total | Average number of calendar days from requisition date to employee start date per new external hire | External Days to Start / External Hires |
| Time to Start - External - Exempt | Average number of calendar days from requisition date to employee start date per new external exempt hire | Exempt External Days to Start / Exempt External Hires |
| Time to Start - External - Nonexempt | Average number of calendar days from requisition date to employee start date per new external nonexempt hire | Nonexempt External Days to Start / Nonexempt External Hires |
| Time to Start - Internal - Total | Average number of calendar days from requisition date to employee start date per new internal hire | Internal Days to Start / Internal Hires |
| Time to Start - Internal - Exempt | Average number of calendar days from requisition date to employee start date per new internal exempt hire | Exempt Internal Days to Start / Exempt Internal Hires |
| Time to Start - Internal - Nonexempt | Average number of calendar days from requisition date to employee start date per new internal nonexempt hire | Nonexempt Internal Days to Start / Nonexempt Internal Hires |
| Offer Acceptance Rate | Offers accepted as a percentage of offers made | Total Offers Accepted / Total Offers Extended |
| Offer Acceptance Rate - External | External new hire offers accepted as a percentage of external new hire offers made | External Offers Accepted / External Offers Extended |
| Offer Acceptance Rate - College | New college hire offers accepted as a percentage of new college hire offers made | College Offers Accepted / College Offers Extended |
| Sign-On Bonus Percent | New hires receiving a sign-on bonus as a percentage of total new external and college hires | Total Hires Receiving Sign-On Bonuses / (External Hires + College Hires) |
| Sign-On Bonus Percent - Executive | New executive hires receiving a sign-on bonus as a percentage of total executive new hires | Executive Hires Receiving Sign-On Bonuses / Executive Hires |
| Sign-On Bonus Percent - Manager | New manager hires receiving a sign-on bonus as a percentage of total manager new hires | Manager Hires Receiving Sign-On Bonuses / Manager Hires |
| Sign-On Bonus Factor | Average sign-on bonus amount for each new hire who received sign-on bonus | Total Sign-On Bonus Cost / Total Hires Receiving Sign-On Bonuses |

| Name | Description | Formula |
|---|---|---|
| Sign-On Bonus Factor - Executive | Average sign-on bonus amount for each new executive hire who received sign-on bonus | Executive Sign-On Bonus Cost / Executive Hires Receiving Sign-On Bonuses |
| Sign-On Bonus Factor - Manager | Average sign-on bonus amount for each new manager hire who received sign-on bonus | Manager Sign-On Bonus Cost / Manager Hires Receiving Sign-On Bonuses |

## Training and Development

*Table A2.7 Training and Development Measures*

| Name | Description | Formula |
|---|---|---|
| Employees Trained Percent | Employees receiving training as a percentage of total headcount | Workforce Trained / Total Headcount |
| Employees Trained Percent - Regular Employees | Employees receiving training as a percentage of regular employee headcount | Workforce Trained / Regular Employee Headcount |
| Training Cost Factor - Total | Dollars spent on training for each employee who received training | Total Training Cost / Workforce Trained |
| Training Cost Factor - Excluding Trainee Pay & Benefits | Average dollars spent on training for each employee who received training excluding Trainee Pay & Benefits | Total Training Cost (ETPB) / Workforce Trained |
| Training Cost Percent - Total | Total training costs as a percentage of operating expense | Total Training Cost / Operating Expense |
| Training Cost Percent - Excluding Trainee Pay & Benefits - Total | Total training costs as a percentage of operating expense excluding Trainee Pay & Benefits (ETPB) | Total Training Cost (ETPB) / Operating Expense |
| Training Cost Percent - Excluding Trainee Pay & Benefits - External | External training cost as a percentage of total training cost excluding Trainee Pay & Benefits | External Training Cost (ETPB) / Total Training Cost (ETPB) |
| Training Cost Percent - Excluding Trainee Pay & Benefits - Internal | Internal training cost as a percentage of total training cost excluding Trainee Pay & Benefits | Internal Training Cost (ETPB) / Total Training Cost (ETPB) |
| Training Cost HR Expense % - Total | Training cost as a percentage of HR expense | Total Training Cost / HR Expense |
| Training Cost HR Expense Percent - Excluding Trainee Pay & Benefits | Training cost as a percentage of HR expense excluding Trainee Pay & Benefits | Total Training Cost (ETPB) / HR Expense |

| Name | Description | Formula |
|---|---|---|
| Training Compensation % - Total | Training cost as a percentage of workforce on payroll compensation cost | Total Training Cost / Compensation Cost (Workforce on Payroll) |
| Training Compensation Percent - Regular Employees | Training cost as a percentage of regular employee compensation cost | Total Training Cost / Comp. Cost (Regular Employees) |
| Training Compensation Percent - Excluding Trainee Pay & Benefits (ETPB) | Training cost, excluding Trainee Pay & Benefits (ETPB) as a percentage of workforce on payroll compensation cost | Total Training Cost (ETPB) / Compensation Cost (Workforce on Payroll) |
| Training Compensation Percent - Regular Employees (ETPB) | Training cost, excluding trainee pay and benefits (ETPB) as a percentage of regular employee compensation cost | Total Training Cost ETPB / Comp. Cost (Regular Employees) |
| Training Headcount Investment Factor | Average dollars spent on training per headcount employee | Total Training Cost / Total Headcount |
| Training Headcount Investment Factor - Regular Employees | Average dollars spent on training per regular employee headcount | Total Training Cost / Regular Employee Headcount |
| Training Headcount Investment Factor - Excluding Trainee Pay & Benefits | Average dollars spent on training excluding trainee pay and benefits (ETPB) per headcount employee | Total Training Cost (ETPB) / Total Headcount |
| Training Headcount Investment Factor - Regular Employees (ETPB) | Average dollars spent on training excluding trainee pay and benefits (ETPB) per regular employee headcount | Total Training Cost (ETPB) / Regular Employee Headcount |
| Headcount Training Factor - Total | Average number of hours of training per headcount employee | Total Training Hours / Total Headcount |
| Headcount Training Factor - Regular Employees | Average number of hours of training per regular employee headcount | Total Training Hours / Regular Employee Headcount |
| FTE Training Factor - Total | Average number of hours of training per FTE employee | Total Training Hours / Total FTE |
| FTE Training Factor - Regular Employees | Average number of hours of training per regular FTE | Total Training Hours / Regular FTE |
| Training FTE Investment Factor - Total | Average dollar amount spent on training per FTE employee | Total Training Cost / Total FTE |
| Training FTE Investment Factor - Regular Employees | Average dollar amount spent on training per regular FTE | Total Training Cost / Regular FTE |
| Training FTE Investment Factor - Total - Exempt | Average dollar amount spent on exempt training per exempt FTE employee | Total Exempt Training Cost / Exempt FTE |
| Training FTE Investment Factor - Total - Nonexempt | Average dollar amount spent on nonexempt training per nonexempt FTE employee | Total Nonexempt Training Cost / Nonexempt FTE |

| Name | Description | Formula |
|---|---|---|
| Training FTE Investment Factor - Excl. Trainee Pay & Benefits - Total | Average dollar amount spent on training, excluding Trainee Pay & Benefits (ETPB), per FTE employee | Total Training Cost (ETPB) / Total FTE |
| Training FTE Investment Factor - ETPB - Regular Employees | Average dollar amount spent on training, excluding Trainee Pay & Benefits (ETPB), per regular FTE employee | Total Employee Training Cost (ETPB) / Regular FTE |
| Training FTE Investment Factor - ETPB - Exempt | Average dollar amount spent on exempt training, excluding Trainee Pay & Benefits (ETPB), per exempt FTE employee | Exempt Training Cost (ETPB) / Exempt FTE |
| Training FTE Investment Factor - ETPB - Nonexempt | Average dollar amount spent on nonexempt training, excluding Trainee Pay & Benefits (ETPB), per nonexempt FTE employee | Nonexempt Training Cost (ETPB) / Nonexempt FTE |
| Training Staff Ratio - Total | Average number of FTE employees supported by each training staff FTE | Total FTE / Training Staff FTE |
| Training Staff Ratio - Regular Employees | Average number of regular FTE employees supported by each training staff FTE | Regular FTE / Training Staff FTE |
| Training Cost Per Hour - Total | Average dollars spent on training per hour of training provided | Total Training Cost / Total Training Hours |
| Training Cost Per Hour - Excluding Trainee Pay & Benefits | Average dollars spent on training, excluding Trainee Pay and Benefits (ETPB) per hour of training provided | Total Training Cost (ETPB) / Total Training Hours |
| Training Hours Percent - Internal Staff | Internal training hours as a percentage of total training hours | Internal Staff Training Hours / Total Training Hours |
| Training Hours Percent - External Staff | External training hours as a percentage of total training hours | External Staff Training Hours / Total Training Hours |

*Appendix 3*

# The Public API

**Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **706**

**Hierarchy for all Packages** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **707**
Package Hierarchies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 707
Class Hierarchy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 707
Interface Hierarchy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 707

**API Index** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **708**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 708
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 708
G . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 708
H . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 711
I . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 711
P . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 712
S . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 712

**Packages** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **713**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 713
Package com.sas.solutions.hcm.publicapi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 714
Package com.sas.solutions.hcm.publicapi.beans . . . . . . . . . . . . . . . . . . . . . . . . . 714
Package com.sas.solutions.hcm.publicapi.connector . . . . . . . . . . . . . . . . . . . . . . 714
Package com.sas.solutions.hcm.publicapi.exceptions . . . . . . . . . . . . . . . . . . . . . 714
Package com.sas.solutions.hcm.publicapi.models . . . . . . . . . . . . . . . . . . . . . . . . 715

**Hierarchies** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **715**
Hierarchy For Package com.sas.solutions.hcm.publicapi . . . . . . . . . . . . . . . . . . 715
Hierarchy For Package com.sas.solutions.hcm.publicapi.beans . . . . . . . . . . . . . . 715
Hierarchy For Package com.sas.solutions.hcm.publicapi.connector . . . . . . . . . . . 715
Hierarchy For Package com.sas.solutions.hcm.publicapi.exceptions . . . . . . . . . . 716
Hierarchy For Package com.sas.solutions.hcm.publicapi.models . . . . . . . . . . . . . 716

**Classes** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **716**
Class CategoryBean . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 716
Class GenericBean . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 718
Class GenericResultBean . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 720
Class PublicAPITableModel . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 721
Class ProfileBean . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 722
Class PublicAPIFactory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 725
Class HCMPublicAPIException . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 726

**Interface PublicAPIInterface** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **727**
Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 727
getEmployeeList . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 729
getEmployeeDetails . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 729

getTableList . . . . . 729
getEmployeeList . . . . . 729
getEmployeeDetails . . . . . 730
getEmployeeList . . . . . 730
getEmployeeDetails . . . . . 731
getProfileDetails . . . . . 731
getHeaderDetails . . . . . 731
getEmployeePhotographSrc . . . . . 731
getConfigValue . . . . . 732
getHierarchyTree . . . . . 732
getTableModel . . . . . 732
getAuthorizedColumns . . . . . 733
isActionPermitted . . . . . 733
getCategoryLink . . . . . 733

**Serialized Form . . . . . 734**

# Overview

This appendix contains documentation for the public API that SAS Human Capital Management provides for customizing employee profile templates. For information about using this API, see "Customizing the Employee Profile Templates" on page 101.

SAS Human Capital Management supports custom employee profile templates in the form of Java Server Pages (JSPs) that use classes and methods from the HCM Public API. Public API classes are divided into various packages according to their function.

The most important interface is com.sas.solutions.hcm.publicapi.PublicAPIInterface. This interface is the gateway to the employee data for the client JSPs, which use the interface to obtain employee data in the form of beans (such as the GenericBean and CategoryBean) or standard Java API objects (such as List or String).

Sample Usage:

```
PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();
GenericResultBean employeeDetailsBean = hcmPublicAPI.getEmployeeDetails(
   "empmast", "10433", request);
```

| Packages | Description |
|---|---|
| com.sas.solutions.hcm.publicapi on page 714 | This is the parent-level package. It contains key public API interfaces. |
| com.sas.solutions.hcm.publicapi.beans on page 714 | Provides Public API classes that act as beans for holding data. |
| com.sas.solutions.hcm.publicapi.connector on page 714 | Provides Public API factory classes. |
| com.sas.solutions.hcm.publicapi.exceptions on page 714 | Provides classes for exceptions in Public API. |
| com.sas.solutions.hcm.publicapi.models on page 715 | Provides Public API model classes. |

# Hierarchy for all Packages

## Package Hierarchies

The following package hierarchies are available:

- com.sas.solutions.hcm.publicapi on page 715
- com.sas.solutions.hcm.publicapi.beans on page 715
- com.sas.solutions.hcm.publicapi.connector on page 715
- com.sas.solutions.hcm.publicapi.exceptions on page 716
- com.sas.solutions.hcm.publicapi.models on page 716

## Class Hierarchy

The following class hierarchy is available:

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.CategoryBean
   com.sas.solutions.hcm.publicapi.beans.GenericBean
   com.sas.solutions.hcm.publicapi.beans.GenericResultBean
   HcmDefaultTableModel
      com.sas.solutions.hcm.publicapi.models.PublicAPITableModel
   com.sas.solutions.hcm.publicapi.beans.ProfileBean
   com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory
   java.lang.Throwable (implements java.io.Serializable)
      java.lang.Exception
         com.sas.solutions.hcm.publicapi.exceptions.HCMPublicAPIException
```

See the following for more information about each class:

- “Class CategoryBean” on page 716
- “Class GenericBean” on page 718
- “Class GenericResultBean” on page 720
- “Class PublicAPITableModel” on page 721
- “Class ProfileBean” on page 722
- “Class PublicAPIFactory” on page 725
- “Class HCMPublicAPIException” on page 726

## Interface Hierarchy

The following interface hierarchy is available:

```
com.sas.solutions.hcm.publicapi.PublicAPIInterface
```

See the following for more information about the PublicAPIInterface.

- “Interface PublicAPIInterface” on page 727

# API Index

## *Overview*

This index contains an alphabetical list of all classes, interfaces, constructors, methods, and fields.

### *C*

**`CategoryBean`**
Class in com.sas.solutions.hcm.publicapi.beans. This bean contains category details such as category name, category label, and category attributes for fixed and custom categories. See "Class CategoryBean" on page 716 for more information.

**`CategoryBean()`**
Constructor for class com.sas.solutions.hcm.publicapi.beans.CategoryBean. See Table A3.6 on page 716 for more information.

**`com.sas.solutions.hcm.publicapi`**
Package com.sas.solutions.hcm.publicapi. See "Package com.sas.solutions.hcm.publicapi" on page 714 for more information.

**`com.sas.solutions.hcm.publicapi.beans`**
Package com.sas.solutions.hcm.publicapi.beans. See "Package com.sas.solutions.hcm.publicapi.beans" on page 714 for more information.

**`com.sas.solutions.hcm.publicapi.connector`**
Package com.sas.solutions.hcm.publicapi.connector. See "Package com.sas.solutions.hcm.publicapi.connector" on page 714 for more information.

**`com.sas.solutions.hcm.publicapi.exceptions`**
Package com.sas.solutions.hcm.publicapi.exceptions. See "Package com.sas.solutions.hcm.publicapi.exceptions" on page 714 for more information.

**`com.sas.solutions.hcm.publicapi.models`**
Package com.sas.solutions.hcm.publicapi.models. See "Package com.sas.solutions.hcm.publicapi.models" on page 715 for more information.

### *G*

**`GenericBean`**
Class in com.sas.solutions.hcm.publicapi.beans. This is a generic bean that contains name, value, and label. See "Class GenericBean" on page 718 for more information.

**`GenericBean()`**
Constructor for class com.sas.solutions.hcm.publicapi.beans.GenericBean. See Table A3.6 on page 716 for more information.

**`GenericResultBean`**
Class in com.sas.solutions.hcm.publicapi.beans. This bean contains column name list and row data. See "Class GenericResultBean" on page 720 for more information.

**`GenericResultBean()`**

Constructor for class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See Table A3.12 on page 720 for more information.

**`getAuthorizedColumns(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns a list of authorized column names, given a table name. See “getAuthorizedColumns” on page 733 for more information.

**`getCategoriesList()`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the categories list with their attributes. See “getCategoriesList” on page 724 for more information.

**`getCategoryAttributesList()`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the list of attributes of the selected fixed category. See “getCategoryAttributesList” on page 717 for more information.

**`getCategoryLink(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns a link to view a category. See “getCategoryLink” on page 733 for more information.

**`getColumnInfo(int, String)`**

Method in class com.sas.solutions.hcm.publicapi.models.PublicAPITableModel. Gets the column label of a column with the passed index. See “getColumnInfo” on page 722 for more information.

**`getColumnInfoNames()`**

Method in class com.sas.solutions.hcm.publicapi.models.PublicAPITableModel. Gets all column labels from column information. See “getColumnInfoNames” on page 722 for more information.

**`getColumnNameList()`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See “getColumnNameList” on page 721 for more information.

**`getConfigValue(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns the value corresponding to the key passed, from the HCM configuration. See “getConfigValue” on page 732 for more information.

**`getCustomCategoryAttributesTableModel()`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the table model of the selected custom category. See “getCustomCategoryAttributesTableModel” on page 718 for more information.

**`getEmployeeDetails(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns the specified employee's details from configured default table. See “getEmployeeDetails” on page 729 for more information.

**`getEmployeeDetails(String, String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns specified employee's details from the specified table. See “getEmployeeDetails” on page 730 for more information.

**`getEmployeeDetails(String, String, List<String>, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns details of an employee for specified columns and table. See "getEmployeeDetails" on page 731 for more information.

**`getEmployeeList(HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns the details of all employees from configured default table. See "getEmployeeList" on page 729 for more information.

**`getEmployeeList(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns details of all employees from the specified table. See "getEmployeeList" on page 729 for more information.

**`getEmployeeList(String, List<String>, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm. publicapi.PublicAPIInterface. Returns details of all employees from specified table for a list of columns. See "getEmployeeList" on page 730 for more information.

**`getEmployeePhotographSrc(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns the relative path to a photograph of the specified employee according to configuration. See "getEmployeePhotographSrc" on page 731 for more information.

**`getEmployeeProfilePublicAPI()`**

Static method in class com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory. This method returns an instance of a class implementing PublicAPIInterface to get employee details. See "getEmployeeProfilePublicAPI" on page 725 for more information.

**`getEmpPhotoSrc()`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the source of employee photograph. See "getEmpPhotoSrc" on page 724 for more information.

**`getExternalActions()`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the list of external actions. See "getExternalActions" on page 724 for more information.

**`getHeaderAttributesList()`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the headerAttributes. See "getHeaderAttributesList" on page 724 for more information.

**`getHeaderDetails(HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns employee profile's header details. See "getHeaderDetails" on page 731 for more information.

**`getHierarchyTree(String, HttpServletRequest)`**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns an object of FastRelationshipTree type, given a hierarchy code and a table name. See "getHierarchyTree" on page 732 for more information.

**`getLabel()`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the categoryLabel. See "getLabel" on page 718 for more information.

**getLabel()**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "getLabel" on page 720 for more information.

**getName()**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the categoryName. See "getName" on page 718 for more information.

**getName()**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "getName" on page 719 for more information.

**getProfileDetails(String, HttpServletRequest)**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns employee details for specified category of current profile selected by the user. See "getProfileDetails" on page 731 for more information.

**getRowDataList()**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See "getRowDataList" on page 721 for more information.

**getTableList(HttpServletRequest)**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns a list of all authorized tables to the current user. See "getTableList" on page 729 for more information.

**getTableModel(String[], String, String, HttpServletRequest)**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns a PublicAPITableModel based on the parameters passed. See "getTableModel" on page 732 for more information.

**getValue()**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "getValue" on page 719 for more information.

***H***

**HCMPublicAPIException**

Exception in com.sas.solutions.hcm.publicapi.exceptions. A wrapper exception to hold any other exception that might be generated from HCM code. See "Class HCMPublicAPIException" on page 726 for more information.

**HCMPublicAPIException()**

Constructor for exception com.sas.solutions.hcm.publicapi.exceptions.HCMPublicAPIException. See Table A3.24 on page 726 for more information.

**HCMPublicAPIException(String, Throwable)**

Constructor for exception com.sas.solutions.hcm.publicapi. exceptions.HCMPublicAPIException. See Table A3.24 on page 726 for more information.

***I***

**isActionPermitted(String, HttpServletRequest)**

Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. See "isActionPermitted" on page 733 for more information.

**`isCustomCategorySelected()`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Returns whether selected category is fixed or custom. See "isCustomCategorySelected" on page 724 for more information.

## *P*

**`ProfileBean`**

Class in com.sas.solutions.hcm.publicapi.beans. This bean contains the employee profile-related details such as header attributes, category details, and external actions. See "Class ProfileBean" on page 722 for more information.

**`ProfileBean()`**

Constructor for class com.sas.solutions.hcm.publicapi.beans.ProfileBean. See Table A3.18 on page 722 for more information.

**`PublicAPIFactory`**

Class in com.sas.solutions.hcm.publicapi.connector. This is a factory class for getting instances of classes implementing Public API interfaces. See "Class PublicAPIFactory" on page 725 for more information.

**`PublicAPIFactory()`**

Constructor for class com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory. See Table A3.21 on page 725 for more information.

**`PublicAPIInterface`**

Interface in com.sas.solutions.hcm.publicapi. This interface contains Public API methods that HCM exposes to external APIs. See "Interface PublicAPIInterface" on page 727 for more information.

**`PublicAPITableModel`**

Class in com.sas.solutions.hcm.publicapi.models. This class, through inheritance, is an extension of javax.swing.table.DefaultTableModel. See "Class PublicAPITableModel" on page 721 for more information.

**`PublicAPITableModel(HcmDefaultTableModel)`**

Constructor for class com.sas.solutions.hcm.publicapi.models.PublicAPITableModel. This constructor is used by internal classes to create an instance of this class. See "Class PublicAPITableModel" on page 721 for more information.

## *S*

**`setCategoriesList(List<CategoryBean>)`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the categories list with their attributes. See "setCategoriesList" on page 724 for more information.

**`setCategoryAttributesList(List<GenericBean>)`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the list of attributes of the selected fixed category. See "setCategoryAttributesList" on page 718 for more information.

**`setColumnNameList(List)`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See "setColumnNameList" on page 721 for more information.

**`setCustomCategoryAttributesTableModel(DefaultTableModel)`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the table model of the selected custom category. See "setCustomCategoryAttributesTableModel" on page 718 for more information.

**`setCustomCategorySelected(boolean)`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the selected category's type. See "setCustomCategorySelected" on page 724 for more information.

**`setEmpPhotoSrc(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the source of employee photograph. See "setEmpPhotoSrc" on page 724 for more information.

**`setExternalActions(List<GenericBean>)`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the list of external actions. See "setExternalActions" on page 725 for more information.

**`setHeaderAttributesList(List<GenericBean>)`**

Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the headerAttributes. See "setHeaderAttributesList" on page 724 for more information.

**`setLabel(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the categoryLabel. See "setLabel" on page 718 for more information.

**`setLabel(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setLabel" on page 720 for more information.

**`setName(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the categoryName. See "setName" on page 718 for more information.

**`setName(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setName" on page 719 for more information.

**`setRowDataList(List)`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See "setRowDataList" on page 721 for more information.

**`setValue(String)`**

Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setValue" on page 719 for more information.

# Packages

## *Overview*

This section contains the following types of information about available packages:

- Interfaces
- Classes
- Enums

- Exceptions
- Errors
- Annotation Types

## Package com.sas.solutions.hcm.publicapi

This is the parent-level package. It contains key public API interfaces.

***Table A3.1*** *Interface Summary*

| | |
|---|---|
| PublicAPIInterface on page 727 | This interface contains Public API methods that HCM exposes to external APIs. |

## Package com.sas.solutions.hcm.publicapi.beans

Provides Public API classes that act as beans for holding data.

***Table A3.2*** *Class Summary*

| | |
|---|---|
| CategoryBean on page 716 | This bean contains category details such as category name, category label, and category attributes for fixed and custom categories. |
| GenericBean on page 718 | This is a generic bean that contains name, value, and label. |
| GenericResultBean on page 720 | This bean contains column name list and row data. |
| ProfileBean on page 722 | This bean contains the employee profile-related details such as header attributes, category details, and external actions. |

## Package com.sas.solutions.hcm.publicapi.connector

Provides Public API factory classes.

***Table A3.3*** *Class Summary*

| | |
|---|---|
| PublicAPIFactory on page 725 | This is a factory class for getting instances of classes implementing Public API interfaces |

## Package com.sas.solutions.hcm.publicapi.exceptions

Provides classes for exceptions in Public API.

***Table A3.4*** *Exception Summary*

| | |
|---|---|
| HCMPublicAPIException on page 726 | This is a wrapper exception to hold any other exception that might be generated from HCM code. |

### *Package com.sas.solutions.hcm.publicapi.models*

Provides Public API model classes.

***Table A3.5*** *Class Summary*

| | |
|---|---|
| PublicAPITableModel on page 721 | This class, through inheritance, is an extension of javax.swing.table.DefaultTableModel. |

## Hierarchies

### *Hierarchy For Package com.sas.solutions.hcm.publicapi*

#### *Package Hierarchies*

All Packages

#### *Interface Hierarchy*

```
com.sas.solutions.hcm.publicapi.PublicAPIInterface
```

### *Hierarchy For Package com.sas.solutions.hcm.publicapi.beans*

#### *Package Hierarchies*

All Packages

#### *Class Hierarchy*

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.CategoryBean
   com.sas.solutions.hcm.publicapi.beans.GenericBean
   com.sas.solutions.hcm.publicapi.beans.GenericResultBean
   com.sas.solutions.hcm.publicapi.beans.ProfileBean
```

### *Hierarchy For Package com.sas.solutions.hcm.publicapi.connector*

#### *Package Hierarchies*

All Packages

### Class Hierarchy

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory
```

## Hierarchy For Package com.sas.solutions.hcm.publicapi.exceptions

### Package Hierarchies

All Packages

### Class Hierarchy

```
java.lang.Object
   java.lang.Throwable (implements java.io.Serializable)
      java.lang.Exception
         com.sas.solutions.hcm.publicapi.exceptions.HCMPublicAPIException
```

## Hierarchy For Package com.sas.solutions.hcm.publicapi.models

### Package Hierarchies

All Packages

### Class Hierarchy

```
java.lang.Object
   HcmDefaultTableModel
      com.sas.solutions.hcm.publicapi.models.PublicAPITableModel
```

# Classes

## Class CategoryBean

### Overview

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.CategoryBean
```

```
public class CategoryBean
```

```
extends java.lang.Object
```

This bean contains category details such as category name, category label, and category attributes for fixed and custom categories.

***Table A3.6*** *Constructor Summary*

| |
|---|
| **`CategoryBean()`** |

*Table A3.7 Method Summary*

| | |
|---|---|
| **java.util.List<GenericBean>** | **getCategoryAttributesList()**<br>Gets the list of attributes of the selected fixed category. |
| **javax.swing.table.DefaultTableModel** | **getCustomCategoryAttributesTableModel()**<br>Gets the table model of the selected custom category. |
| **java.lang.String** | **getLabel()**<br>Gets the categoryLabel. |
| **java.lang.String** | **getName()**<br>Gets the categoryName. |
| **void** | **setCategoryAttributesList(java.util.List<GenericBean> categoryAttributesList)**<br>Sets the list of attributes of the selected fixed category. |
| **void** | **setCustomCategoryAttributesTableModel(javax.swing.table.DefaultTableModel customCategoryAttributesTableModel)**<br>Sets the table model of the selected custom category. |
| **void** | **setLabel(java.lang.String label)**<br>Sets the categoryLabel. |
| **void** | **setName(java.lang.String name)**<br>Sets the categoryName. |

*Table A3.8 Methods inherited from class java.lang.Object*

| |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait** |

### Constructor

```
CategoryBean

public CategoryBean()
```

### getCategoryAttributesList

```
public java.util.List<GenericBean> getCategoryAttributesList()

    // Returns categoryAttributesList, which is the list of attributes
    // of the selected fixed category.
```

### setCategoryAttributesList

```
public void setCategoryAttributesList(java.util.List<GenericBean>
   categoryAttributesList)

   // Sets categoryAttributesList, which is the list of attributes
   // of the selected fixed category.
```

### getCustomCategoryAttributesTableModel

```
public javax.swing.table.DefaultTableModel
   getCustomCategoryAttributesTableModel()

   // Returns customCategoryAttributesTableModel, which is the
   // table model of the selected custom category.
```

### setCustomCategoryAttributesTableModel

```
public void setCustomCategoryAttributesTableModel(
   javax.swing.table.DefaultTableModel customCategoryAttributesTableModel)

   // Sets customCategoryAttributesTableModel, which is the
   // table model of the selected custom category.
```

### getName

```
public java.lang.String getName()

   // Returns categoryName.
```

### setName

```
public void setName(java.lang.String name)

   // Sets categoryName.
```

### getLabel

```
public java.lang.String getLabel()

   // Returns categoryLabel.
```

### setLabel

```
public void setLabel(java.lang.String label)

   // Sets categoryLabel.
```

## Class GenericBean

### Overview

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.GenericBean

public class GenericBean

extends java.lang.Object
```

This is a generic bean that contains name, value, and label.

*Table A3.9* Constructor Summary

| |
|---|
| **GenericBean()** |

*Table A3.10* Method Summary

| | |
|---|---|
| **java.lang.String** | **getLabel()** |
| **java.lang.String** | **getName()** |
| **java.lang.String** | **getValue()** |
| **void** | **setLabel(java.lang.String label)** |
| **void** | **setName(java.lang.String name)** |
| **void** | **setValue(java.lang.String value)** |

*Table A3.11* Methods inherited from class java.lang.Object

| |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait** |

## Constructor

```
GenericBean

public GenericBean()
```

## getName

```
public java.lang.String getName()

    // Returns the name.
```

## setName

```
public void setName(java.lang.String name)

    // Sets the name.
```

## getValue

```
public java.lang.String getValue()

    // Returns the value.
```

## setValue

```
public void setValue(java.lang.String value)

    // Sets the value.
```

### getLabel

```
public java.lang.String getLabel()

   // Returns the label.
```

### setLabel

```
public void setLabel(java.lang.String label)

   // Sets the label.
```

## Class GenericResultBean

### Overview

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.GenericResultBean

public class GenericResultBean

extends java.lang.Object
```

This bean contains column name list and row data.

***Table A3.12*** *Constructor Summary*

| |
|---|
| **GenericResultBean()** |

***Table A3.13*** *Method Summary*

| | |
|---|---|
| **java.util.List** | **getColumnNameList()** |
| **java.util.List** | **getRowDataList()** |
| **void** | **setColumnNameList(java.util.List columnNameList)** |
| **void** | **setRowDataList(java.util.List rowDataList)** |

***Table A3.14*** *Methods inherited from class java.lang.Object*

| |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait** |

### Constructor

```
GenericResultBean

public GenericResultBean()
```

### getColumnNameList

```
public final java.util.List getColumnNameList()

   // Returns columnNameList.
```

### setColumnNameList

```
public final void setColumnNameList(java.util.List columnNameList)

   // Sets columnNameList.
```

### getRowDataList

```
public final java.util.List getRowDataList()

   // Returns rowDataList.
```

### setRowDataList

```
public final void setRowDataList(java.util.List rowDataList)

   // Sets rowDataList.
```

## Class PublicAPITableModel

### Overview

```
javax.swing.table.DefaultTableModel
   HcmDefaultTableModel
      com.sas.solutions.hcm.publicapi.models.PublicAPITableModel

public class PublicAPITableModel

extends HcmDefaultTableModel
```

This class, through inheritance, is an extension of javax.swing.table.DefaultTableModel. It also provides column information, such as column labels from the underlying table model.

***Table A3.15*** *Constructor*

| |
|---|
| **PublicAPITableModel(HcmDefaultTableModel tableModel)**<br>This constructor is used by internal classes to create an instance of this class. |

***Table A3.16*** *Method Summary*

| | |
|---|---|
| **java.lang.Object** | **getColumnInfo(int index, java.lang.String columnInfoName)**<br>Gets the column label of a column with the passed index. |
| **java.lang.Object[]** | **getColumnInfoNames()**<br>Gets all column labels from column information. |

*Table A3.17 Methods inherited from class java.lang.Object*

|  |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait** |

### Constructor

```
PublicAPITableModel

public PublicAPITableModel(HcmDefaultTableModel tableModel)
```

This constructor is used by internal classes to create an instance of this class. Currently there is no identified requirement for the clients of this Public API to create this table model. Hence, this constructor is intended to be used by internal classes only, and not supported for public API clients.

### getColumnInfo

```
public java.lang.Object getColumnInfo(int index,
   java.lang.String columnInfoName)

   // The parameter index is the index of a column. The first
   // index has a value of 1. The parameter columnInfoName is a
   // required information field about the column. This function
   // uses the values of index and columnInfoName to return a
   // string object containing desired column information. For
   // example, to return a string object containing column label
   // information, the value of index should be the matching
   // index position of the column, and the value of
   // columnInfoName should be COLUMN_NAME.
```

### getColumnInfoNames

```
public java.lang.Object[] getColumnInfoNames()

   // Returns a string array that contains column labels.
```

## Class ProfileBean

### Overview

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.beans.ProfileBean

public class ProfileBean

extends java.lang.Object
```

This bean contains the employee profile-related details such as header attributes, category details, and external actions.

*Table A3.18 Constructor Summary*

|  |
|---|
| **ProfileBean()** |

*Table A3.19* Method Summary

| | |
|---|---|
| `java.util.List<CategoryBean>` | `getCategoriesList()`<br>Gets the categories list with their attributes. |
| `java.lang.String` | `getEmpPhotoSrc()`<br>Gets the source of employee photograph. |
| `java.util.List<GenericBean>` | `getExternalActions()`<br>Gets the list of external actions. |
| `java.util.List<GenericBean>` | `getHeaderAttributesList()`<br>Gets the headerAttributes. |
| `boolean` | `isCustomCategorySelected()`<br>Returns whether selected category is fixed or custom. |
| `void` | `setCategoriesList(java.util.List<CategoryBean> categoriesList)`<br>Sets the categories list with their attributes. |
| `void` | `setCustomCategorySelected(boolean isCustomCategorySelected)`<br>Sets the selected category's type. |
| `void` | `setEmpPhotoSrc(java.lang.String empPhotoSrc)`<br>Sets the source of employee photograph. |
| `void` | `setExternalActions(java.util.List<GenericBean> externalActions)`<br>Sets the list of external actions. |
| `void` | `setHeaderAttributesList(java.util.List<GenericBean> headerAttributesList)`<br>Sets the headerAttributes. |

*Table A3.20* Methods inherited from class java.lang.Object

| |
|---|
| `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait` |

## Constructor

```
ProfileBean

public ProfileBean()
```

### setHeaderAttributesList

```
public void setHeaderAttributesList(java.util.List<GenericBean>
   headerAttributesList)

   // Sets headerAttributes with headerAttributesList.
```

### getHeaderAttributesList

```
public java.util.List<GenericBean> getHeaderAttributesList()

   // Returns headerAttributesList, which contains headerAttributes.
```

### getCategoriesList

```
public java.util.List<CategoryBean> getCategoriesList()

   // Returns categoriesList, which contains categories.
```

### setCategoriesList

```
public void setCategoriesList(java.util.List<CategoryBean> categoriesList)

   // Sets categories with categoriesList.
```

### isCustomCategorySelected

```
public boolean isCustomCategorySelected()

   // Returns the value of isCustomCategorySelected, which indicates
   // whether the selected category is fixed or custom.
```

### setCustomCategorySelected

```
public void setCustomCategorySelected(boolean isCustomCategorySelected)

   // Sets the value of isCustomCategorySelected, which indicates
   // whether the selected category is fixed or custom.
```

### getEmpPhotoSrc

```
public java.lang.String getEmpPhotoSrc()

   // Returns the value of empPhotoSrc, which indicates the source
   // of an employee photograph.
```

### setEmpPhotoSrc

```
public void setEmpPhotoSrc(java.lang.String empPhotoSrc)

   // Sets the value of empPhotoSrc, which indicates the source
   // of an employee photograph.
```

### getExternalActions

```
public java.util.List<GenericBean> getExternalActions()

   // Returns externalActions, which is a list of external actions.
```

### setExternalActions

```
public void setExternalActions(java.util.List<GenericBean> externalActions)

   // Sets externalActions, which is a list of external actions.
```

## Class PublicAPIFactory

### Overview

```
java.lang.Object
   com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory

public class PublicAPIFactory

extends java.lang.Object
```

This is a factory class for getting instances of classes implementing Public API interfaces

***Table A3.21*** *Constructor Summary*

| |
|---|
| **PublicAPIFactory()** |

***Table A3.22*** *Method Summary*

| | |
|---|---|
| **static PublicAPIInterface** | **getEmployeeProfilePublicAPI()**<br>This method returns an instance of a class implementing PublicAPIInterface to get employee details. |

***Table A3.23*** *Methods inherited from class java.lang.Object*

| |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait** |

### Constructor

```
PublicAPIFactory

public PublicAPIFactory()
```

### getEmployeeProfilePublicAPI

```
public static PublicAPIInterface getEmployeeProfilePublicAPI()

   // Returns an instance of a class that implements PublicAPIInterface
   // to get employee details.
```

## Class HCMPublicAPIException

### Overview

```
java.lang.Object
   java.lang.Throwable
      java.lang.Exception
         com.sas.solutions.hcm.publicapi.exceptions.HCMPublicAPIException

All Implemented Interfaces:
   java.io.Serializable

public class HCMPublicAPIException

extends java.lang.Exception
```

A wrapper exception to hold any other exception that might be generated from HCM code.

See Also: "Serialized Form" on page 734.

*Table A3.24 Constructor Summary*

| |
|---|
| **HCMPublicAPIException()** |
| **HCMPublicAPIException(java.lang.String message, java.lang.Throwable cause)** |

*Table A3.25 Methods inherited from class java.lang.Throwable*

| |
|---|
| **fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString** |

*Table A3.26 Methods inherited from class java.lang.Object*

| |
|---|
| **clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait** |

### Constructor

```
HCMPublicAPIException

public HCMPublicAPIException()

HCMPublicAPIException

public HCMPublicAPIException(java.lang.String message,
   java.lang.Throwable cause)
```

# Interface PublicAPIInterface

## *Overview*

```
public interface PublicAPIInterface

com.sas.solutions.hcm.publicapi

// This interface contains Public API methods that are
// exposed to external APIs.
```

***Table A3.27*** *Method Summary*

| | |
|---|---|
| **java.util.List<GenericBean>** | **getAuthorizedColumns(java.lang.String tableName, HttpServletRequest request)**<br>Returns a list of authorized column names, given a table name. |
| **java.lang.String** | **getCategoryLink(java.lang.String categoryName, HttpServletRequest request)**<br>Returns a link to view a category. |
| **java.lang.String** | **getConfigValue(java.lang.String configKey, HttpServletRequest request)**<br>Returns the value corresponding to the key passed from the HCM configuration. |
| **GenericResultBean** | **getEmployeeDetails(java.lang.String employeeId, HttpServletRequest request)**<br>Returns the specified employee's details from configured default table. |
| **GenericResultBean** | **getEmployeeDetails(java.lang.String tableName, java.lang.String employeeId, HttpServletRequest request)**<br>Returns specified employee's details from the specified table. |

| | |
|---|---|
| **`GenericResultBean`** | **`getEmployeeDetails(java.lang.String tableName, java.lang.String employeeId, java.util.List<java.lang.String> columnList, HttpServletRequest request)`**<br>Returns details of an employee for specified columns and table. |
| **`GenericResultBean`** | **`getEmployeeList(HttpServletRequest request)`**<br>Returns the details of all employees from configured default table. |
| **`GenericResultBean`** | **`getEmployeeList(java.lang.String tableName, HttpServletRequest request)`**<br>Returns details of all employees from the specified table. |
| **`GenericResultBean`** | **`getEmployeeList(java.lang.String tableName, java.util.List<java.lang.String> columnList, HttpServletRequest request)`**<br>Returns details of all employees from specified table for a list of columns. |
| **`java.lang.String`** | **`getEmployeePhotographSrc(java.lang.String employeeId, HttpServletRequest request)`**<br>Returns the relative path to a photograph of the specified employee according to configuration. |
| **`java.util.List<GenericBean>`** | **`getHeaderDetails(HttpServletRequest request)`**<br>Returns employee profile's header details. |
| **`FastRelationshipTree`** | **`getHierarchyTree(java.lang.String hierarchyCode, HttpServletRequest request)`**<br>Returns an object of FastRelationshipTree type, given a hierarchy code and a table name. |
| **`ProfileBean`** | **`getProfileDetails(java.lang.String categoryId, HttpServletRequest request)`**<br>Returns employee details for specified category of current profile selected by the user. |
| **`java.util.List<java.lang.String>`** | **`getTableList(HttpServletRequest request)`**<br>Returns a list of all authorized tables to the current user |

| | |
|---|---|
| **`PublicAPITableModel`** | **`getTableModel(java.lang.String[] columnNames, java.lang.String tableName, java.lang.String where, HttpServletRequest request)`**<br>Returns a PublicAPITableModel based on the parameters passed. |
| **`boolean`** | **`isActionPermitted(java.lang.String actionName, HttpServletRequest request)`** |

### *getEmployeeList*

```
GenericResultBean getEmployeeList(HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter request is an HTTPServletRequest object for the
   // current request. This function returns a GenericResultBean
   // instance that contains the details of all employees from a
   // configured default table.
```

### *getEmployeeDetails*

```
GenericResultBean getEmployeeDetails(java.lang.String employeeId,
                                     HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter employeeId is the employee ID of the employee
   // whose details are required.  The parameter request is the
   // HTTPServletRequest object for the current request.  This
   // function returns a GenericResultBean that contains the
   // specified employee's details from a configured default table.
```

### *getTableList*

```
java.util.List<java.lang.String> getTableList(HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter request is an HTTPServletRequest object for the
   // current request. This function returns a list of string type
   // objects that contain table names of all authorized tables for
   // the current user.
```

### *getEmployeeList*

```
GenericResultBean getEmployeeList(java.lang.String tableName,
                                  HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter tableName is the name of a table.  The parameter
   // request is an HTTPServletRequest object for the current request.
   // This function returns a GenericResultBean that contains employee
   // details of all employees in the specified table.
```

### getEmployeeDetails

```
GenericResultBean getEmployeeDetails(java.lang.String tableName,
                                     java.lang.String employeeId,
                                     HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter tableName is the name of the table from where
   // to get details.  The parameter employeeId is the employee Id
   // of the employee whose details are to be returned.  The
   // parameter request is an HTTPServletRequest object for the
   // current request. This function returns a GenericResultBean
   // that contains column names and values (employee details) of
   // the specified employee in the specified table.

   // Sample Usage:

      PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();
      GenericResultBean employeeDetailsBean = hcmPublicAPI.getEmployeeDetails(
         "empmast", "10433", request);

      List columnNameList = employeeDetailsBean.getColumnNameList();
      Iterator itrColumnName = columnNameList.iterator();
      List rowDataList = employeeDetailsBean.getRowDataList();
      Iterator itrRowDataList = rowDataList.iterator();

      while (itrColumnName.hasNext()) {
         String columnName = (String) itrColumnName.next();
         System.out.print(columnName);
      }

      while (itrRowDataList.hasNext()) {
         List rowList = (List) itrRowDataList.next();
         Iterator itrRowList = rowList.iterator();

         while (itrRowList.hasNext()) {
            String value = (String) itrRowList.next();
            System.out.print(value);
         }
      }
```

### getEmployeeList

```
GenericResultBean getEmployeeList(java.lang.String tableName,
                                  java.util.List<java.lang.String> columnList,
                                  HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter tableName is the name of the table from which values
   // are to be returned.  The parameter columnList is a list of column names.
   // The parameter request is an HTTPServletRequest object for the current
   // request. This function returns details of all employees from the specified
   // table and for the specified list of columns using a GenericResultBean that
   // contains column names and list of values.
```

### getEmployeeDetails

```
GenericResultBean getEmployeeDetails(java.lang.String tableName,
                                     java.lang.String employeeId,
                                     java.util.List<java.lang.String> columnList,
                                     HttpServletRequest request)
   throws HCMCoreException, HCMException

   // The parameter tableName is the name of a table from which values are
   // to be returned.  The parameter employeeId is the employee Id of the
   // employee.  The parameter columnList is a list of string objects that
   // contains column names. The parameter request is an HTTPServletRequest
   // object for the current request. This function returns details of an
   // employee for the specified columns and table using a GenericResultBean
   // that contains column names and lists of values.
```

### getProfileDetails

```
ProfileBean getProfileDetails(java.lang.String categoryId,
                              HttpServletRequest request)
   throws HCMCoreException, HCMConfigurationException

   // The parameter categoryId is the Id of the desired category. The
   // parameter request is an HTTPServletRequest object for the current
   // request.  This function returns employee details for the specified
   // category of the current profile selected by the user using a
   // ProfileBean, which is populated with category details for the
   // specified category.
```

### getHeaderDetails

```
java.util.List<GenericBean> getHeaderDetails(HttpServletRequest request)
   throws HCMCoreException, HCMConfigurationException

   // The parameter request is an HTTPServletRequest object for the current
   // request. This function returns an employee profile's header details
   // using a list of GenericBean objects.
```

### getEmployeePhotographSrc

```
java.lang.String getEmployeePhotographSrc(java.lang.String employeeId,
                                          HttpServletRequest request)

   // The parameter employeeId is the employee Id of the employee.  The
   // paramater request is an HTTPServletRequest object for the current
   // request. This function uses the value of employeeId to return the
   // relative path to a photograph of the specified employee, according
   // to configuration.
```

### getConfigValue

```
java.lang.String getConfigValue(java.lang.String configKey,
                                HttpServletRequest request)

   // The parameter configKey passes a string that contains a key to get
   // the value of. The parameter request is an HTTPServletRequest object
   // for the current request. This function returns a string that contains
   // the value corresponding to the key specified by configKey. This value
   // is dependent on the HCM configuration.

   // If there are multiple keys with same name, this function will return
   // the first entry that matches.

   // Sample Usage:

      PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();
      String configValue = hcmPublicAPI.getConfigValue("ENABLESPM", request);
```

### getHierarchyTree

```
FastRelationshipTree getHierarchyTree(java.lang.String hierarchyCode,
                                      HttpServletRequest request)
   throws HCMPublicAPIException

   // The parameter hierarchyCode is a string that contains a hierarchy
   // code. The parameter request is an HTTPServletRequest object for the
   // current request. This function returns a FastRelationshipTree object
   // that contains the hierarchy tree structure that corresponds to the
   // hierarchy code and table name.

   // Sample Usage:

      PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();
      FastRelationshipTree tree = hcmPublicAPI.getHierarchyTree("INTORG_HR",
                                                                 request);
```

### getTableModel

```
PublicAPITableModel getTableModel(java.lang.String[] columnNames,
                                  java.lang.String tableName,
                                  java.lang.String where,
                                  HttpServletRequest request)
   throws HCMPublicAPIException

   // The parameter columnNames is a string array that contains names of
   // columns to be obtained. The parameter tableName is a string that
   // contains the name of the table. The parameter where is a string that
   // contains an SQL where clause to be applied. The parameter request is
   // an HTTPServletRequest object for the current request. This function
   // returns a PublicAPITableModel object that contains data based upon
   // the parameters passed into this function. Results are filtered
   // according to a security mechanism.
   // Sample Usage:

      PublicAPIInterface hcmPublicAPI = PublicAPIFactory.getPublicAPI();
      String [] columnNames = {"EMPLOYEE_ID","EMPLOYEE_NAME","AGE"};
      String tableName = "EMPMAST";
      String where = "INTORG_HR = \'QA\'";
      PublicAPITableModel tableModel = hcmPublicAPI.getTableModel(columnNames,
         tableName, where, request);
```

### getAuthorizedColumns

```
java.util.List<GenericBean> getAuthorizedColumns(java.lang.String tableName,
                                                 HttpServletRequest request)
   throws HCMPublicAPIException

   // The parameter tableName is a string that contains the table name. The
   // parameter request is an HTTPServletRequest object for the current request.
   // This function returns a list of authorized column names for the specified
   // table.

   // Sample Usage:

      List columnsList = hcmPublicAPI.getAuthorizedColumns("EMPMAST",request);
      Iterator iter = columnsList.iterator();

      while (iter.hasNext()){
         String columnName = ((GenericBean)iter.next()).getName();
         out.println("AuthorizedColumn:"+columnName);
         out.println(" ");
      }
```

### isActionPermitted

```
boolean isActionPermitted(java.lang.String actionName,
                          HttpServletRequest request)
   throws HCMPublicAPIException
```

### getCategoryLink

```
java.lang.String getCategoryLink(java.lang.String categoryName,
                                 HttpServletRequest request)
   throws HCMPublicAPIException

   // The parameter request is an HTTPServletRequest object for the current
   // request. This function returns a link to view a category.

   // Sample Usage:

      String categoryName = categoryBean.getName();
```

# Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can access this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

```
Package com.sas.solutions.hcm.publicapi.exceptions

Class com.sas.solutions.hcm.publicapi.exceptions.HCMPublicAPIException
extends java.lang.Exception implements Serializable

// serialVersionUID: -2166472308204638622L
```

# *Appendix 4*
# MySQL Reserved Words

MySQL reserved words cannot be used as column names of MySQL tables. Therefore, you must not use any MySQL reserved word as a dimension code for a dimension that will be represented by a column in a metric table. The safest approach is to avoid defining any dimension code that is identical to a MySQL reserved word.

There is a complete list of MySQL reserved words at the following Web location:

`http://dev.mysql.com/doc/refman/5.0/en/reserved-words.html`

For a discussion of the process of loading metric tables from an external source, see "Loading Metrics" on page 257.

For a discussion of the process of loading metric tables from the HCM Data Mart, see "Loading HCM Metrics into a Metric Table" on page 291 .

For a discussion of dimension codes, see "Creating a Dimension" on page 187.

# *Appendix 5*
# Configuring the SAS Environment Files

**Overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **737**
About the SAS Environment Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 737
**Deploying the SAS Environment Files to an HTTP Server** . . . . . . . . . . . . . . . . . . **738**
**Defining Additional SAS Environments** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **739**
**Defining Additional Solutions Environments** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **739**
Overview of the EnvironmentFactory.xml File . . . . . . . . . . . . . . . . . . . . . . . . . . 739
Customizing the EnvironmentFactory.xml File . . . . . . . . . . . . . . . . . . . . . . . . . . 739

## Overview

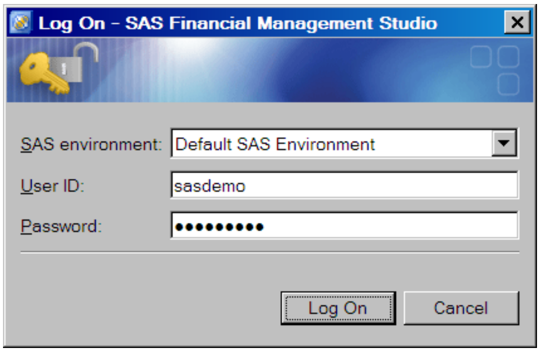
### *About the SAS Environment Files*

SAS Solutions Services and the solutions use two environment files:

- **The SAS environment file** (sas-environment.xml) applies to the SAS Financial Management Studio and SAS Solutions Dimension Editor client applications.
- **The Solutions environment file** (EnvironmentFactory.xml) applies to the SAS Solutions Services Add-In for Microsoft Office and the SAS Financial Management Add-In for Microsoft Excel.

These files enable desktop client applications to determine the location of required services on the middle tier and to obtain a list of services available in the environment.

When you log on to SAS Financial Management Studio or SAS Solutions Dimension Editor, you are asked to select a SAS environment in which the application will operate. Here is an example logon box for SAS Financial Management Studio. The **SAS environment** drop-down list contains a list of available environments. You select an environment and connect to the middle tier for that installation.



SAS environment: Default SAS Environment

User ID: sasdemo

Password: •••••••••

When you log on to the middle tier from Microsoft Word or Microsoft Excel, you are presented with a similar dialog box.

For information about specifying these values during an installation, see “Installing Client Applications” in the installation guide.

## Deploying the SAS Environment Files to an HTTP Server

For the purposes of validating the installation and configuration at a site, use the default sas-environment.xml and EnvironmentFactory.xml locations. After validation, you should deploy those files to a location where they are accessible by all possible clients. The recommended deployment location is an HTTP server.

You must deploy the sas-environment.xml and EnvironmentFactory.xml files to a central location if any of the following situations is true:

- At your site, the solutions are used in multiple environments such as development, test, and production environments. You want your users to be able to select from a list of available environments.
- Secure Sockets Layer (SSL) is configured for the solutions at your site.
- Web authentication is configured for the solutions at your site.

Follow these steps:

1. Deploy the sas-environment.xml file to the HTTP server.

   This file is located in the ***`SAS-config-dir`***`\Lev1\Web\Common` directory.

2. Deploy the EnvironmentFactory.xml file to the HTTP server.

   This file is located in the ***`SAS-config-dir`***`\Lev1\Web\Applications\SASSolutionsServices5.3` directory.

3. On each client machine, edit the appropriate INI file:

   - for SAS Solutions Services Add-In for Microsoft Office and SAS Financial Management Add-In for Microsoft Excel: ***`SAS-install-dir`***`\SASSolutionsServicesAdd-InforMicrosoftOffice\5.3\SASSolutionsOfficeClient.ini`

     Change the URL to the environment file so that it points to the HTTP server. For example:

```
[Environment Factory]
http://myhttpserver:port/EnvironmentFactory.xml
```

- for SAS Solutions Dimension Editor: ***SAS-install-dir*** **\SASSolutionsDimensionEditor\5.3\soldimedit.ini**

4. If your end users perform their own client installations, make them aware of the URLs to these files.

## Defining Additional SAS Environments

The SAS environment file (sas-environment.xml) applies to the SAS Financial Management Studio and SAS Solutions Dimension Editor client applications.

If you want your end users to select from multiple SAS environments, modify the sas-environment.xml file that you deployed to an HTTP server. Include information about each of the available environments.

For more information, see “Configuring the SAS Environment File” in the *SAS Intelligence Platform: Web Application Administration Guide*. That book is available at **support.sas.com/92administration**.

## Defining Additional Solutions Environments

### *Overview of the EnvironmentFactory.xml File*

The Solutions environment file (EnvironmentFactory.xml) applies to the add-in clients for Microsoft Office applications. By default, the EnvironmentFactory.xml file defines a single Solutions environment that represents the installation in which it resides. The file has the following structure:

```
<environments>
  <environment name="environment-name">
    <object>
      ...
      </object>
      ... [additional object definitions]
  </environment>
  ... [additional environment definitions]
</environments>
```

The ***environment-name*** is an identifier that appears in the selection list when a user logs on to the middle tier from SAS Solutions Services Add-In for Microsoft Office or SAS Financial Management Add-In for Microsoft Excel.

### *Customizing the EnvironmentFactory.xml File*

To customize the EnvironmentFactory.xml file to support more than one configuration of the solutions, follow these steps:

1. Open the EnvironmentFactory.xml file that you deployed to an HTTP server.

2. Copy the default environment definition (from **`<environment name="default">`** through **`</environment>`**) and paste it directly after the **`</environment>`** tag of the first entry.

3. Give this second environment definition a new environment-name, such as **`test`**.

   Names must be valid as XML attributes.

4. Modify the server name (and port number, if necessary) of each entry for the **`test`** environment definition.

   In a WebLogic configuration, the result might resemble the following:

***Example Code A5.1*** *EnvironmentFactory.xml Example with Two Environments*

```
<environments>
  <environment name="test">
    <object>
      <name>default</name>
      <java.naming.factory.initial>
        weblogic.jndi.WLInitialContextFactory
      </java.naming.factory.initial>
      <java.naming.provider.url>
        t3://server1:7201
      </java.naming.provider.url>
    </object>
    <object>
      <name>login</name>
         <webservice.url>
        http://server1:7201/SASSolutionsServices/services/AuthenticationService
      </webservice.url>
        </object>
      ...
   </environment>
<environment name="production">
    <object>
      <name>default</name>
      <java.naming.factory.initial>
        weblogic.jndi.WLInitialContextFactory
      </java.naming.factory.initial>
      <java.naming.provider.url>
        t3://server2:7201
      </java.naming.provider.url>
    </object>
    <object>
      <name>login</name>
         <webservice.url>
        http://server2:7201/SASSolutionsServices/services/AuthenticationService
      </webservice.url>
        </object>
      ...
     </environment>
 </environments>
```

5. Follow steps 2–4 (with a different ***`environment-name`*** and different server names) to create an environment definition for the production environment.

6. Save your changes.

7. Restart the server.

It is not necessary to restart the managed servers or redeploy the SAS Solutions Services application. The next time users try to log on to the middle tier from one of the Microsoft Office add-ins, the new environment choices are available.

*Note:* The EnvironmentFactory.xml file should be synchronized with the sas-environment.xml file. That is, if you define multiple SAS environments for SAS Financial Management Studio, you should define comparable Solutions environments for the add-ins.

# *Appendix 6*
# Default Port Usage

## *Overview*

The servers in the SAS Intelligence Platform communicate with clients and other servers using TCP/IP. Each server listens on a particular port or ports for incoming requests. During installation, the SAS Deployment Wizard enables you to either accept the default ports or to specify different port numbers for some servers.

## *Default Port Numbers for SAS Servers and Spawners*

The following table shows the default port numbers for SAS servers and spawners that are installed in a `Lev1` SAS environment that includes the SAS Performance Management solutions. The table also includes default third-party ports.

Your site might use different port numbers than the ones that are shown here. For a complete list, see the pre-install checklist for your site.

***Table A6.1*** *Default Port Numbers and Descriptions*

| Port Number | Description |
|---|---|
| 25 | SMTP mail: Port used by mailhost or Simple Mail Transfer Protocol (SMTP). Used to send administrative e-mail notices and end-user alert notifications. |
| 80 | HTTP Server: Handles proxy requests to application server. Also used for static assets such as themes, style sheets, and images. |
| 2171 | SAS Table Server port. |
| 3306 | Database server port. All JDBC access from the managed servers goes through this port to the MySQL server. SAS/ACCESS to MySQL also uses this port. |
| 5091 | SAS Remote Services application port.<br>All client access to remote Foundation Services is directed through this port. In solutions deployments, only middle-tier clients communicate via RMI. Therefore, it is not necessary to open this port to external access (that is, to other clients on the network) in a firewall-protected environment. |
| 5451 | SAS OLAP Server port. |
| 5556 | (Oracle WebLogic Server) NodeManager port. |

| Port Number | Description |
|---|---|
| 6051 | Event Broker service: Listen port for administrator. |
| 7001, 7101, 7201, 7301, 7401 | (Oracle WebLogic Server) Nonsecure listen ports for managed servers. Additional managed server port numbers are incremented by 100. Used by Web applications and by many of the client applications, such as SAS Financial Management Studio. |
| 7002, 7102, 7202, 7302, 7402 | (Oracle WebLogic Server) Secure listen ports for managed servers. Additional managed server port numbers are incremented by 100. |
| 7501 | (Oracle WebLogic Server) Listen port for the administration server. |
| 7551 | SAS/CONNECT Server port. |
| 8111 | Event Broker service: Used by SAS Solutions Services for HTTP transports into the Foundation Services Event Broker. Events fired by SAS code into the middle tier are communicated via this port. |
| 8451 | Operating System Services scheduler port. |
| 8551 | SAS/SHARE Server. |
| 8561 | SAS Metadata Server: Default port for metadata access. This is also the default multicast UDP port number. |
| 8571 | SAS Object Spawner Load Balancing: Load-balancing requests from SAS Object Spawner go through this port. |
| 8581 | SAS object spawner: Operator port. |
| 8591 | SAS Workspace Server port. Might also be shared by Metadata utilities SAS Workspace Server port. |
| 8601 | SAS Stored Process Server: Bridge connection. |
| 8611, 8621, 8631 | SAS Stored Process Server: Load balancing connections 1, 2, and 3 (MultiBridge). |
| 8701 | SAS Pooled Workspace Server port. |
| 8801, 8811, 8821 | SAS object spawner: Pooled workspace server port banks 1, 2, and 3. |
| 9000 | Port used to register SAS BI portlets with the portal. |
| 9876 | The default port on which the in-process RMI registry is hosted by ODCS and through which the query processors make the bootstrap contact. |
| 10021 | SAS Deployment Tester server port. |

When you set up a multiple-level SAS environment (for example, an environment that consists of separate levels for development, test, and production), the SAS Deployment Wizard increments each port number by 1 for each level. For example, the default **`Lev1`** port number for the SAS Metadata Server is 8561. A **`Lev2`** environment would use port 8562.

*Note:* SAS PC Files Server uses port 8621 by default, but this port is also used by the SAS Stored Process Server. If you installed SAS PC Files Server and need to change its port number, see "Configure PC Files Server" on page 343.

For additional information, see the "Default SAS Ports" appendix of the *SAS Intelligence Platform: System Administration Guide*.

## *Default Port Numbers for Third-Party Software*

***Table A6.2*** *Default Port Numbers and Descriptions for Third-Party Software*

| Software | Port Number | Description |
|---|---|---|
| Oracle WebLogic Server | 5556 | NodeManager port. |
| | 7001, 7101, 7201, 7301, 7401 | Nonsecure listen ports for managed servers. Additional managed server port numbers are incremented by 100. Used by Web applications and by many of the client applications, such as SAS Financial Management Studio. |
| | 7002, 7102, 7202, 7302, 7402 | Secure listen ports for managed servers. Additional managed server port numbers are incremented by 100. |
| | 7501 | Listen port for the administration server. |
| IBM WebSphere Application Server | 8879 | SOAP port for administrative console. |
| | 8880, 8881 | SOAP port for application servers (additional application servers increment by 1). |
| | 9043 | Secure HTTPS port for administrative console. |
| | 9044, 9045, 9046, 9047, 9048 | Secure HTTPS ports for application server (additional application servers increment by 1). |
| | 9060 | Non-secure HTTP port for administrative console. |
| | 9080, 9081, 9082, 9083, 9084 | Non-secure HTTP ports for application server (additional application servers increment by 1). |
| | 9809 | RMI port for administrative console. |
| | 9811, 9812 | RMI ports for application servers (additional administrative consoles increment by 1). |

# Index

## A

actions 52
add-on configuration 361
  DSX files 362
adding columns to tables 178, 295
adding HCM cubes 300
adding tables to the data model 178, 299
Administration application
  overview 6
AGEFRACT macro variable for SAS Human Capital Management 313
alerts
  Alerts portlets 374
  types of 374
Alerts portlets 374
ALL organization member 198
ANAL family of macro variables for SAS Human Capital Management 304
APP_DIMENSION table 189
APP_FORMULA_READ_MEMBER table 201
APP_FORMULA_TARGET table 201
APP_FORMULA_WRITE_MEMBER table 201
APP_FORMULA table 201
APP_GROUP_ACTIONS table 201
APP_MEMBER_PROPERTY_MAP table 206
APP_PROPERTY table 206
APP_USER_ACTIONS table 201
APP_USER_X_MEMBER table 200
application configuration properties 96
ASSOC_TYPE tables 194
ASSOC tables 194
authentication
  user security and 363

## B

batch job rights 358
Batch Model Facility (BMF) 178
Batch Model Facility for StM 165
Book of Record Currency Code column 198

## C

cache
  refreshing 7
categories, fixed 48
child-parent tables 194, 236
CHRNDROP macro variable for SAS Human Capital Management 313
CHRNINDS macro variable for SAS Human Capital Management 313
CHRNVARS macro variable for SAS Human Capital Management 314
Clear User Cache 371
CODE_LANGUAGE table 181
column filters
  identities, adding 85
  modifying 84
column security 83
  applying 83
columns
  *See also* tables
  adding to tables 178, 295
  increasing character length of 298
competency tables for SAS Human Capital Management 266
configuration directory 166, 339
configuration properties 94
  application properties 96
  custom properties 98
  system properties 94
conventions 166, 339
Create Application Dimension job 189
Create Dimension transformation 188
CUBE_DELETE_TYPE macro variable for SAS Human Capital Management 314

CUBE_DRILL macro variable for SAS Human Capital Management 314
cube jobs for SAS Human Capital Management 281
cube wizard 25
CUBED family of macro variables for SAS Human Capital Management 305
cubes 7
- *See also* dimensions
- adding to HCM Data Mart 300
- creating 25
- deleting 30
- description 25
- information maps, creating 30
- labels 16
- managing 30
- rebuilding 30, 31
- refreshing 30, 31
- viewing 30

cubes, UNIX permissions
- for SAS Human Capital Management 348

CUBEV family of macro variables for SAS Human Capital Management 306
currency hierarchies
- structure of 197

custom configuration properties 98
custom objects
- security 72

custom properties of members 205
CUTOFF family of macro variables for SAS Human Capital Management 307

## D

data encodings 172
data locales 181
data marts 171
data pathways 171, 178
Data Validation transformation 176
data-tier server
- definition 166
- security 169

DEBUG macro variable for SAS Human Capital Management 315
default language flag 181
default member of a hierarchy 196
detail data store (DDS) jobs
- load techniques in 176
- modifying 176
- role and context 171
- scheduling 177
- testing 177
- validation in 176

detail table jobs for SAS Human Capital Management 281
diagnostic utility
- e-mailed results 92
- launchDiagnostics_cmd.bat 91
- launchDiagnostics_UI.bat 91
- overview 87
- running from Administration application 92
- saving results to PDF 92
- stand-alone mode 90
- summary of tests 88
- viewing the results 92

diagnostics configuration files
- for SAS Human Capital Management 347
- localizing 360

dimension management 337
dimension tables 194
dimension types
- adding 215
- definition 187

dimensions 22
- creating 29, 187, 251
- definition 187
- loading members and hierarchies into 193, 251

DIRECT macro variable for SAS Human Capital Management 315
Document Manager 337
documentation 167
documentation conventions 166, 339
documentation links 339

## E

e-mail addresses
- for notifications 370

EDU macro variable for SAS Human Capital Management 316
Employee ID column 198
employee list 46
- for an employee profile, customizing 50

employee profile
- default 55
- default category 56

employee profile templates
- *See also* Public API
- adding to SAS Human Capital Management 113
- available templates 102
- category details 108
- category list 107
- custom categories 109

customizing 101, 104
deploying 112
employee images 106
external actions list 107
fixed categories 108
header details 105
PublicAPIInterface methods 109
required imports 104
saving to PDF 112
styles 104
employee profiles
actions 52
assigning to users or groups 46
contents 46
copying 52
creating 48
customizing 45, 50
defined 45
deleting 56
fixed categories 48, 51
required permissions 46
employee search 47
EMPPOP macro variable for SAS Human Capital Management 316
encodings of data 172
End Date column 200
environment files 737
*See also* SAS environments
*See also* Solutions environments
EnvironmentFactory.xml 739
exporting
hierarchies 209
members 209
EXT organization member 198
extracting data from source systems 173

F
FACTORSITE table in HCM Data Mart 285, 291
FCTRVARS macro variable for SAS Human Capital Management 291, 317
fixed categories 48
adding 51
FMT_UPDATE macro variable for SAS Human Capital Management 317
folders and files
securing, in operating system 350
forecasting
customizing 124
defining a stored process 120
overview 115
preparing the data 117
process overview 116
formats 35
adding 36
assigning to a column 37
catalog, updating 38
deleting 38
editing 38
predefined 38
viewing 38
formats catalog
updating 38
formats for SAS Human Capital Management 275
formula tables 201

G
general search
default columns 63
geographic analysis 56
actions 52
creating additional maps 59
drill level hierarchy 58
drilling down 56
link fields 56
group and role assignments
examples 368
group data
loading 185
groups
SAS Human Capital Management 367
SAS Solutions Services 365
searching for 73
synchronizing users, groups, and roles 370

H
HCBNCHMRK table in HCM Data Mart 288, 291
HCM data flow 266
HCM Data Mart 171, 279, 285
hcmlabels.properties file 16
hcmmetacolumn table for SAS Human Capital Management 299
hcmmetatable table for SAS Human Capital Management 299
hcmtitles.properties file 16
HEDUNIT macro variable for SAS Human Capital Management 291
hierarchical filters 80
applying 81
hierarchy, selecting 80
overriding 82
power user group 82
hierarchies 22
*See also* hierarchy mappings
definition 187
exporting to staging tables 209

loading into a dimension 193, 251
promoting 201
viewing 23
hierarchy identification tables 194, 227
hierarchy mappings
adding 23
description 22
viewing 24
hierarchy structure tables 194, 236
HIERORDS macro variable for SAS Human Capital Management 318
HIERS macro variable for SAS Human Capital Management 268, 318
HR measures 33
creating 33
deleting 35
editing 35
recalculating 35
viewing 35

I
IACTION internal formats for SAS Human Capital Management 329
IBM WebSphere Server
log file size 380
modifications 379
ICHURN internal formats for SAS Human Capital Management 330
identities
searching 73
IEEOCL internal formats for SAS Human Capital Management 330
IEMPSTA internal formats for SAS Human Capital Management 331
IEXEMPT internal formats for SAS Human Capital Management 331
Import Users 66
Import Users and Groups 370
Import Users and Groups stored process 185
INCOPEN macro variable for SAS Human Capital Management 318
information maps
creating, from cubes 30
creating, from tables 20
opening, in SAS Web Report Studio 33
rebuilding 33
information maps for SAS Human Capital Management 293
INTERNAL_ORG table 198
internal formats for SAS Human Capital Management 329
Internal Organization ADK column 198
INTORG_DIMENSION_CD macro variable for SAS Human Capital Management 268, 319
IONPAYRL internal formats for SAS Human Capital Management 331
IPAYPER internal formats for SAS Human Capital Management 332
IREGTMP internal formats for SAS Human Capital Management 332
ISTECLS internal formats for SAS Human Capital Management 332
ITERM internal formats for SAS Human Capital Management 333

J
JUnit JAR file
for SAS Human Capital Management 347

K
key performance indicator (KPI) 337
KPI Viewer 165, 337
localization 361
post-configuration tasks 350

L
labels for SAS Human Capital Management 266, 274
languages 181
Legal Entity Flag column 198
link fields
for a geographic analysis 56
load techniques in detail data store jobs
in detail data store jobs 176
locales 181
localization
for SAS Human Capital Management 359
log files
additional 382
configuring 381
dynamic configuration 382
location 381
on middle tier 381
log on as a batch job right 358

M
M family of macro variables for SAS Human Capital Management 307
macro files that affect HCM data 274
macro variables for SAS Human Capital Management 304

managed servers 377
map tables 9, 59
marts 171
master tables 8
MAXLEVL macro variable for SAS Human Capital Management 319
MEAS_LEV macro variable for SAS Human Capital Management 291, 319
measures 253
  *See also* HR measures
  *See also* planning measures
  predefined 683
  Saratoga Institute 683
member tables 194, 221, 247
members
  custom properties of 205
  definition 187
  exporting to staging tables 209
  loading into a dimension 193, 251
  promoting 201
metadata identities 363
  creating 358
metadata repository 164
metadata server 166
metrics
  exporting from SAS Financial Management 257
  loading from an external source 257
  loading from the HCM Data Mart 291
Microsoft Office integration 337
middle tier
  log files on 381
middle-tier server 166
migrations 342
MPRINT macro variable for SAS Human Capital Management 320
My Favorites portlets 337, 373
MySQL
  installation directory 166
MySQL reserved words 735
MySQL server
  installation directory 339
  overview 383
  securing access 352
  security 352
  UNIX configuration 384
  Windows configuration 383

**N**

NLS member tables 194, 247
NOTES macro variable for SAS Human Capital Management 320
NUMBER_OF_HIERS macro variable for SAS Human Capital Management 268, 321

**O**

object security 67
  custom objects 72
  list of objects 673
  permissions, adding 70
  permissions, interpreting 68
OLAP schema
  permissions, for SAS Human Capital Management 346
ONPAYRL macro variable for SAS Human Capital Management 321
Oracle WebLogic Server
  modifications 378
order number in a hierarchy 196
organization analysis 60
  defaults 60
organization charts
  *See* organization analysis
organization hierarchies
  for SAS Human Capital Management 268
  structure of 198
ORGMEMBER_EXCLUDES macro variable for SAS Human Capital Management 321

**P**

parent-child tables 194, 236
Performance Aggregate Table portlet 375
Performance Association portlet 375
Performance Dashboard portlet 374
Performance Diagram portlet 375
performance improvement
  SAS Human Capital Management 378
Performance Table portlet 374
PERIOD_TYPE_CD macro variable for SAS Human Capital Management 291, 322
Period Type Code column 200
permanent generation size 343
permissions
  role permissions 364
planning form security 200
planning measures 152
  for new positions 150
  for supplemental schedules 151
  managing plans 158
  overview 150
  SAS code 154, 155
  slicer-dependent 151
Planning Workflow alerts 374
portal customization 373
  Alerts portlets 374
  My Favorites portlets 373
portlets

- Alerts 374
- for KPIs 374
- for scorecards 374
- My Favorites 373
- SAS Strategy Management 375

ports
- default usage 743

POSDIR macro variable for SAS Human Capital Management 322

post-configuration tasks
- after a migration 342
- after an upgrade in place 342
- configuring Quick Help 345
- creating users and groups 358
- for add-on configurations 361
- loading production data 358
- localization 361
- overview 342
- SAS Human Capital Management 346

POSVAR macro variable for SAS Human Capital Management 322

POSVDIR macro variable for SAS Human Capital Management 323

power user group 82

prebuild.sas macro file for SAS Human Capital Management 268, 278, 304

predefined data 171

predictive modeling 129

primary member tables 194, 221

production data, loading 358

profile header 47
- customizing 50

profiles
- *See* employee profiles

promoting members and hierarchies 201, 209

properties files for SAS Human Capital Management 266, 274

properties of members 205

Public API 101
- *See also* employee profile templates

## Q

Quick Help
- configuring 345

## R

Refresh Cache 7

registering users 370

related documentation 167

Reporting Currency Code column 198

reserved words in MySQL 735

retention analysis
- creating a cube 147
- customizing the %MODLPB macro 140
- EMPSCORES table 146, 147
- extraction stage 131
- generating the model 139
- generating the scoring table 138
- HRVANLY2 table 137
- introduction 129
- merge stage 146
- merging the tables 146
- MODELSCORES table 146
- modifying the %PREBUILD macro 134
- modifying the %VTAPB macro 135
- process overview 130
- reporting 147
- scoring table 145
- transformation stage 138

role cache
- flushing (SAS Financial Management) 371

role-based user interface 337

roles 364
- enforcing permissions 364
- SAS Human Capital Management 367
- SAS Solutions Services 366
- searching for 73
- synchronizing users, groups, and roles 370

Roll Up to Parent Flag column 196

row-level security 74
- *See also* hierarchical filters
- default filters 75
- enabling 77
- filters, adding 78
- filters, deleting 79
- filters modifying 79
- how filters are applied 75

runhcm.sas macro file for SAS Human Capital Management 284

## S

SALCVAR macro variable for SAS Human Capital Management 323

SALUNIT macro variable for SAS Human Capital Management 291

sample data
- SAS Human Capital Management 353
- verifying with 353

Saratoga benchmarks for HCM metrics 288

Saratoga Institute
- measures 683

SAS_CURRENCY table 197

SAS_DIMENSION_TYPE table 215

SAS_MEASURE table 253
SAS_PERIOD_TYPE table 200
SAS_USER_EMPLOYEE table 367
SAS BI Dashboard
  manual configuration of DSX files 362
SAS environment files
  deploying to an HTTP server 738
SAS environments
  multiple 739
SAS for Workforce Planning & Budgeting
  *See also* planning measures
  *See also* supplemental schedules
  administering, overview 150
  creating a form set 156
  data entry 158
  overview 149
  planning measures 152
SAS Human Capital Management
  administration overview 3
  date format 359
  diagnostics configuration files 347, 360
  groups 367
  JUnit JAR file 347
  localization 359
  non-English languages 360
  OLAP schema permissions 346
  performance improvement 378
  post-configuration tasks 346
  remote services permissions 346
  roles 367
  sample data 353
  typical users 368
  UNIX permissions for cubes 348
  user qualifications 367
  verifying the installation 353
SAS PC Files Server 343
SAS Solutions Dimension Editor 337
SAS Solutions Services 337
  changes 338
  groups 365
  required skills 338
  roles 366
  user identities 364
SAS Strategy Management portlets 375
SAS Web Report Studio
  modifying properties 349
SAS Web Report Studio and SAS Human Capital Management 293
sas-environment.xml
  *See* SAS environments
SCD Type 2 Loader transformation 176
scheduling
  detail data store jobs 177
  SDM jobs 177
scorecards 165
search criteria
  for an employee profile, customizing 50
search symbols 22
secondary member tables 194, 247
securing folders and files
  UNIX 352
  Windows 351
security 66
  *See also* column security
  *See also* object security
  *See also* row-level security
  additional security measures 66
  authentication and user security 363
  custom objects 72
  data-tier server 169
  importing users 66
  MySQL server 352
  operating system 350
  planning forms 200
  SAS_USER_EMPLOYEE table 66
  searching for identities 73
  Security tab data 201
  user criteria 66
  user data 185
servers 166
slowly changing dimensions 175
Solutions Data Mart (SDM) jobs
  introduction 171
  scheduling 177
  testing 177
Solutions environments 739
  multiple 739
SORT family of macro variables for SAS Human Capital Management 308
SOURCE_DIMENSION_TYPE table 216
SOURCE macro variable for SAS Human Capital Management 324
source systems
  extracting data from 173
SPMEXPSC macro 361
SRVFRACT macro variable for SAS Human Capital Management 324
staging tables
  introduction 171
  loading 173
Start Date column 200
START family of macro variables for SAS Human Capital Management 309
STAT family of macro variables for SAS Human Capital Management 309
STGACUT macro variable for SAS Human Capital Management 288, 291, 324

STGACUTA macro variable for SAS Human Capital Management 288, 291, 324
STGAMAX macro variable for SAS Human Capital Management 291, 326
stgamets.sas macro file for SAS Human Capital Management 285
stored processes
  creating from SAS Data Integration Studio jobs 263
  forecasting 120
Strategy Management portlet 374
summary tables 9
supplemental schedules
  creating 156
  planning measures 151
synchronizing users, groups, and roles 370
system configuration properties 94

**T**

Table Loader transformation 176
tables
  adding 11
  attributes 14
  column attributes 14
  copying 17
  data types 10
  deleting 21
  detail tables 7
  exporting 10, 21
  formatted columns, exporting 10
  formatted columns, importing 10
  hcmtitles.properties file 16
  history tables 7
  importing 10, 11
  importing, custom 13
  information maps, creating 20
  location, for importing 10
  map tables 7, 9
  master tables 7, 8
  properties files 16
  search symbols 22
  summary tables 7, 9
  types 7
  viewing 18
  warning messages, on export 11
templates
  adding 62
  employee profiles 61
  Home page 61
  selecting 61
testing
  detail data store jobs 177
  SDM jobs 177
TIME_PERIOD table 200
time hierarchies
  for SAS Human Capital Management 268
  structure of 200
titles for SAS Human Capital Management 266, 274
total transaction lifetime timeout 379

**U**

UH family of macro variables for SAS Human Capital Management 310
umbrella job for SAS Human Capital Management 282
UNIT family of macro variables for SAS Human Capital Management 311
UNIX security settings 352
upgrade in place 342
user data 185
user IDs for data administrators 169
User Opt-in alerts 374
user security
  authentication and 363
users
  importing 66
  registering 370
  searching for 73
  synchronizing users, groups, and roles 370
users, registering 370
users, synchronizing 370
Users tab data 200
UTF-8 encoding 172

**V**

V family of macro variables for SAS Human Capital Management 312
Valid From Datetime column 175
Valid To Datetime column 175
validation in detail data store jobs 176
VARS family of macro variables for SAS Human Capital Management 312
verification 353

**W**

Web application servers
  IBM WebSphere Server 379
  managing 378
  Oracle WebLogic Server 378
Web applications
  configuring 378
Windows
  log on as a batch job right 358

Windows security settings 351