SAS® Human Capital Management 5.2
Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® Human Capital Management 5.2: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Human Capital Management 5.2: Administrator's Guide

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

1st electronic book, November 2010

Contents

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 1.

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/92administration`.

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

- *SAS Human Capital Management: User's Guide*
- *SAS Solutions Services: System Administration Guide*
- *SAS Solutions Services: Data Administration Guide*
- *SAS Solutions Services: Data Model Reference*
- *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 at:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: 1-800-727-3228
Fax: 1-919-531-9439
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Chapter 1

About SAS Human Capital Management Administration

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

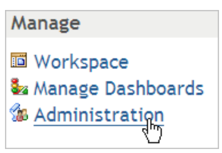
Introduction

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.

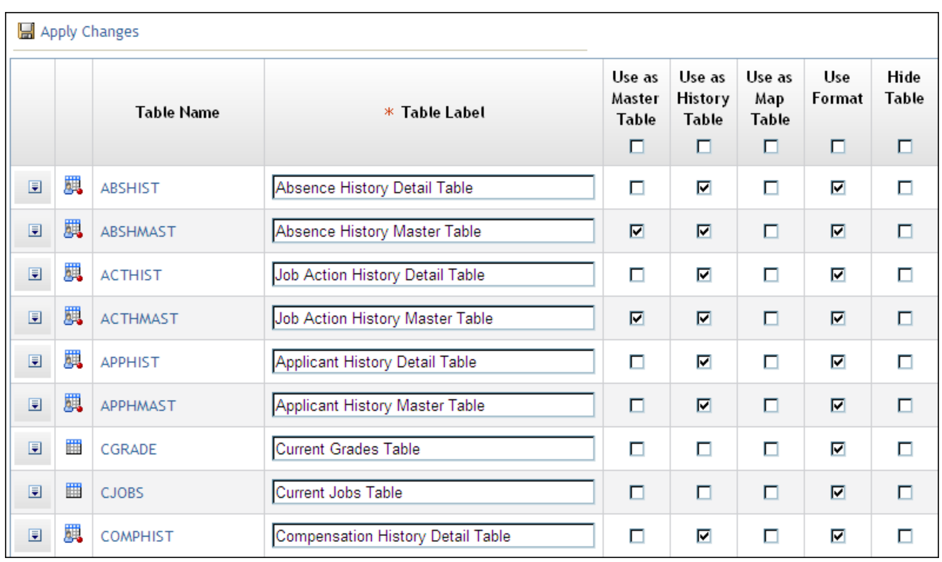


Table Name	* Table Label	Use as Master Table	Use as History Table	Use as Map Table	Use Format	Hide Table
ABSHIST	Absence History Detail Table	☐	☑	☐	☑	☐
ABSHMAST	Absence History Master Table	☑	☑	☐	☑	☐
ACTHIST	Job Action History Detail Table	☐	☑	☐	☑	☐
ACTHMAST	Job Action History Master Table	☑	☑	☐	☑	☐
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 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 the *SAS Solutions Services: Data Administration Guide*.

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. 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 11.
2. Export the table.

   For details, see "Export a Table" on page 18 or "View a Table" on page 16.
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 18.

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** function 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 11.

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 7.

      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 5.
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 12.

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 68.

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 5.)

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 11, and for more information about the SAS_HCMMETACOLUMN table, see "Modify Column Attributes" on page 12.

Note: If you add a table to the HCM database, you must use the **Add Table** function 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**.

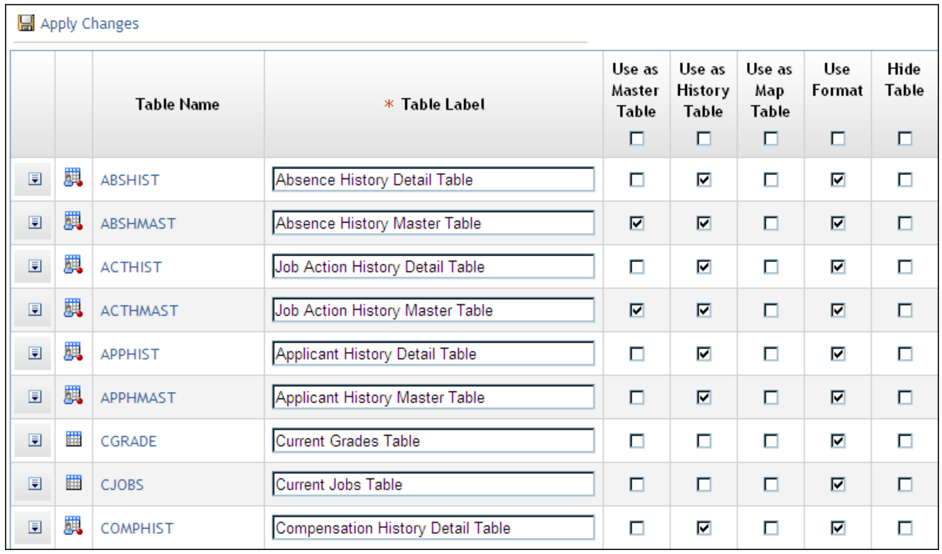

Apply Changes

		Table Name	* Table Label	Use as Master Table	Use as History Table	Use as Map Table	Use Format	Hide Table
		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 12.

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.

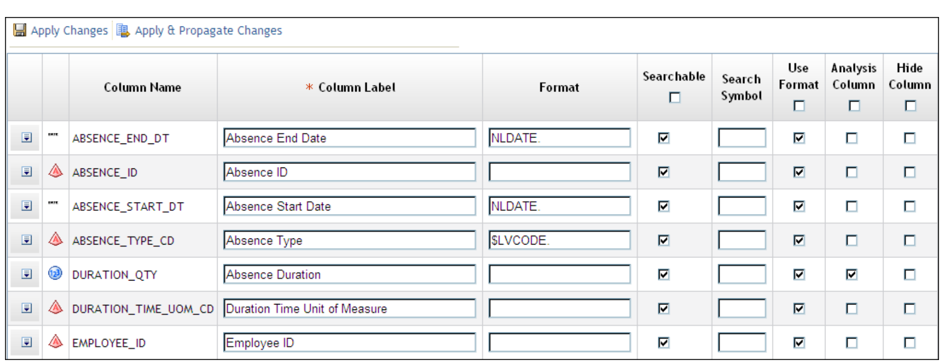
Modify Column Attributes

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.


Apply Changes | Apply & Propagate Changes

		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.	☑		☑	☐	☐
		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. 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. In order for a table's columns to use the display format, the table's **Use Format** property must also be selected. 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.
Analysis Column	Select this check box to make the column available for analysis. For a geographic analysis or a cube, this selection applies only to numeric values (data types of NUM, DATA, or CURRENCY). 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.

Attribute	Description
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: • **Count**: number of non-missing values • **Min**: minimum value • **Max**: maximum value • **Sum**: sum of values • **Number Missing**: number of missing values • **Uncorrected Sum Sq**: uncorrected sum of squares • **Avg**: arithmetic mean or average of values • **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:

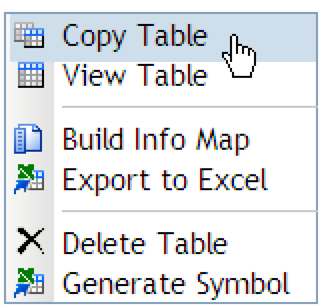
1. Change your current directory to **`!sasroot\hrds\sasmisc`**.
2. Open the appropriate version of the file for editing. The sasmisc folder contains localized versions of the file, in the form hcmtitles_*locale*.properties.

   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

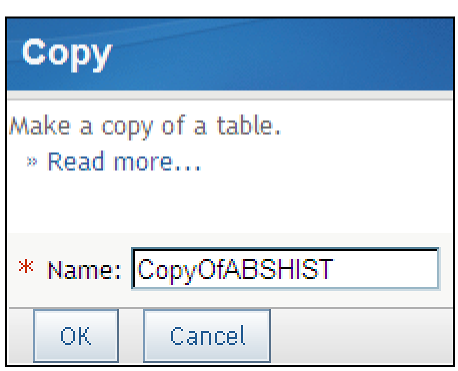
On the **Data** tab of the Administration application, you can copy a table as a backup or for some other purpose. Follow these steps:

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.



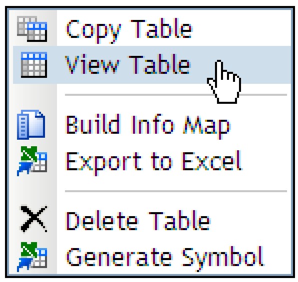
4. Click **OK**.

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 9.

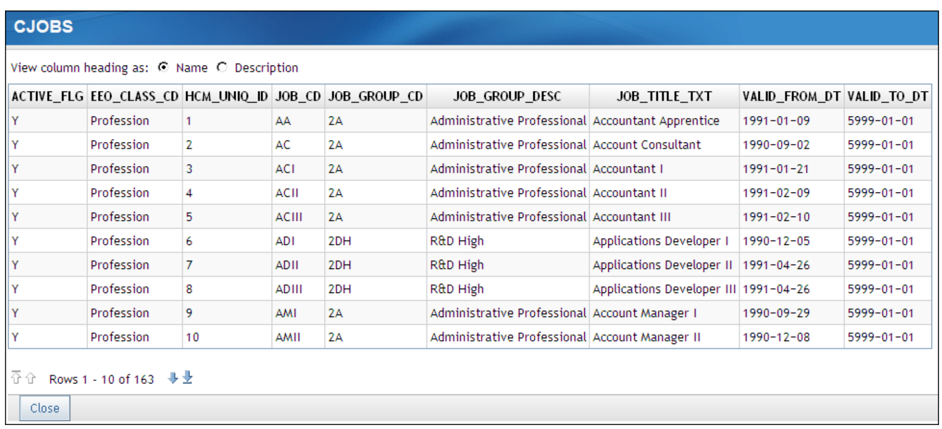
View a Table

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**.

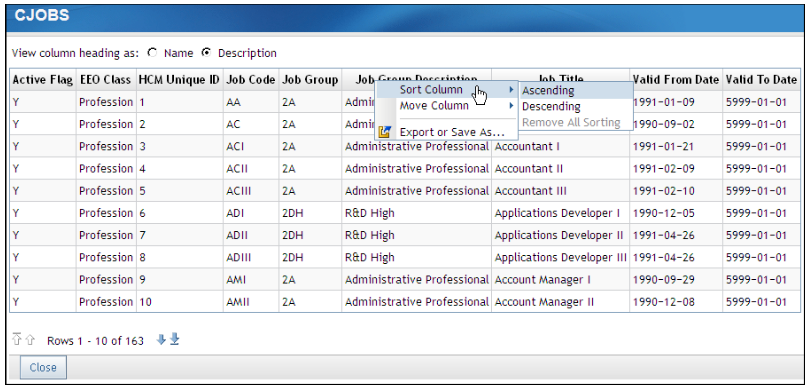


A separate browser window displays the table data.



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

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**.

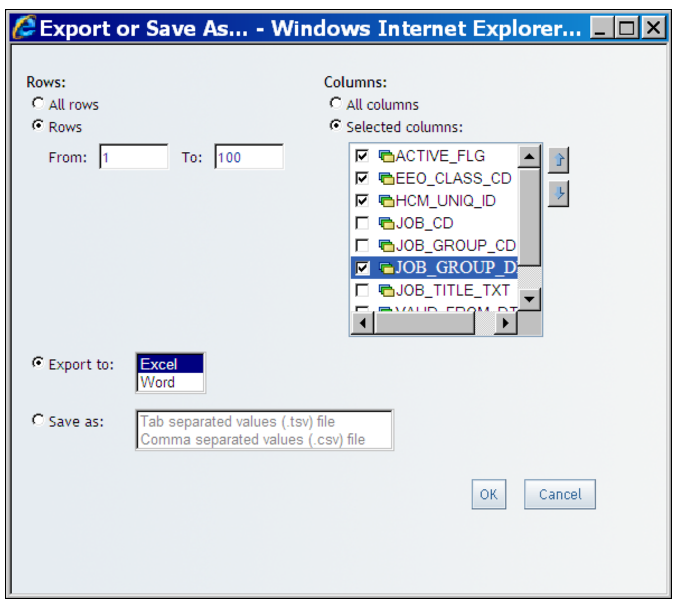


Active Flag	EEO Class	HCM Unique ID	Job Code	Job Group	Job Group Description	Job Title	Valid From Date	Valid To Date
Y	Profession	1	AA	2A	Admi		1991-01-09	5999-01-01
Y	Profession	2	AC	2A	Admi		1990-09-02	5999-01-01
Y	Profession	3	ACI	2A	Administrative Professional	Accountant I	1991-01-21	5999-01-01
Y	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

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:



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.

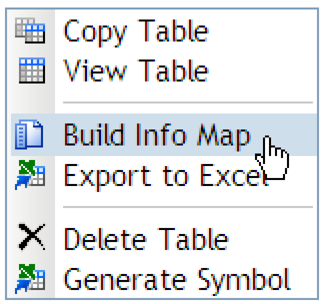
Save as
Select this radio button to save the data as a tab-separated values (TSV) file or as a comma-separated values (CSV) file.

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 7.

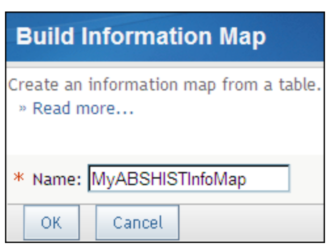
Create an Information Map from a Table

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**.



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 5.)

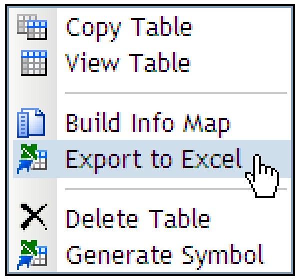
Note: Users must have both ReadMetadata and Read permission for an information map in order to access its data in SAS Web Report Studio. If the information map is built on a cube, users must also have Read permission for the cube.

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.

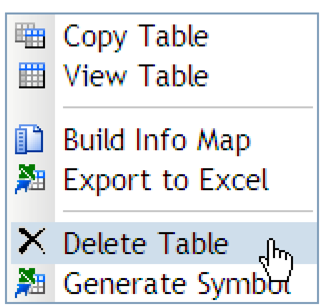
All rows are exported (subject to security provisions), and the column names are used as headings. (To export a subset of table rows, see "View a Table" on page 16.)

For more information about exporting tables, see "About Importing and Exporting Tables" on page 7.

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 21.
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 12.

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.

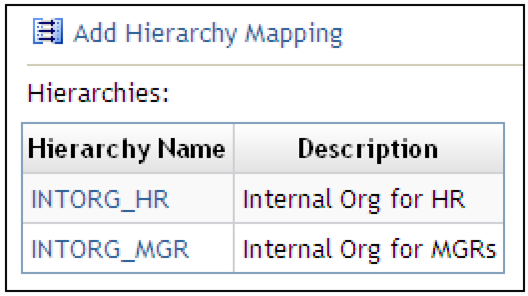
The records are selected by matching the hierarchy code to values in an information table, such as the Employee Master table. The hierarchy mapping specifies which column of the information table contains the matching codes—in this case, the department code.

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.



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 21.

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
Select a hierarchy for this mapping.

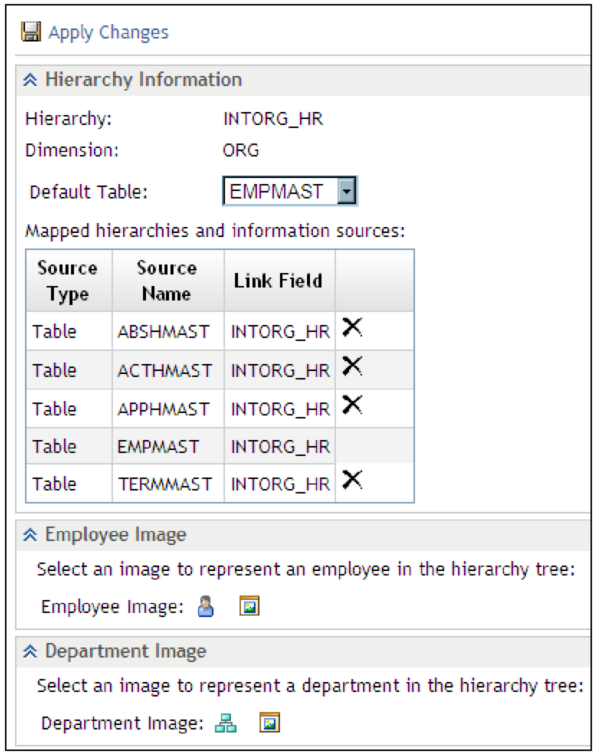
Table
Select an information table for this mapping.

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.



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

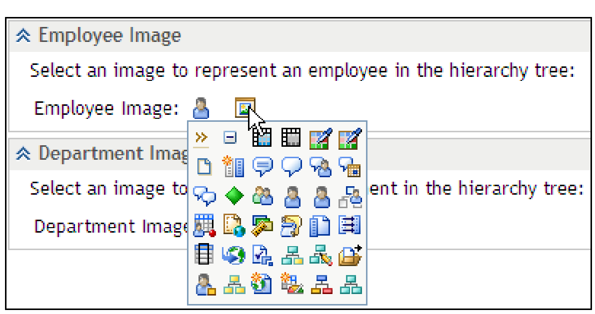
   When a user creates an organization analysis and selects this hierarchy, the default mapping determines which table is presented as the default. However, the user can override the default and select a different table from the available mappings.

   Note: The default mapping applies only to organization analysis. To select defaults for an employee profile, see "Create an Employee Profile" on page 44.

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:



   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

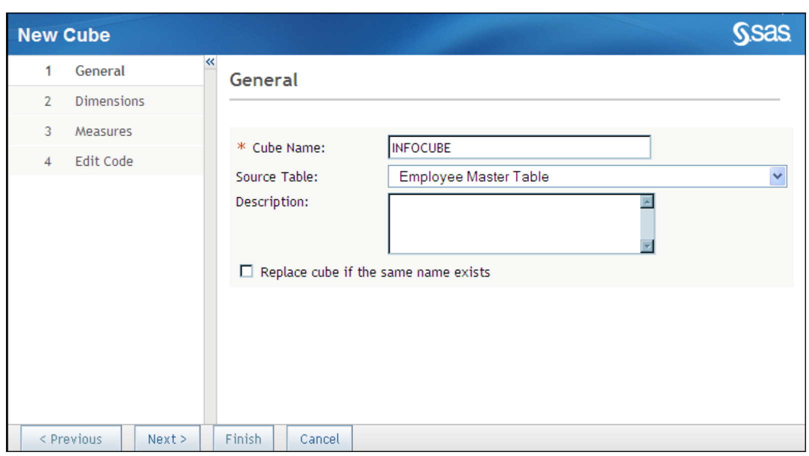
A cube is a set of data that is organized and structured in a hierarchical, multidimensional arrangement that provides more efficient access to data than traditional relational databases. SAS cubes are designed to offer fast data access and efficient data storage. In many cases, a basic cube without 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.



2. Provide general information for your cube.
   a. Type a cube name, such as **`INFOCUBE`**.

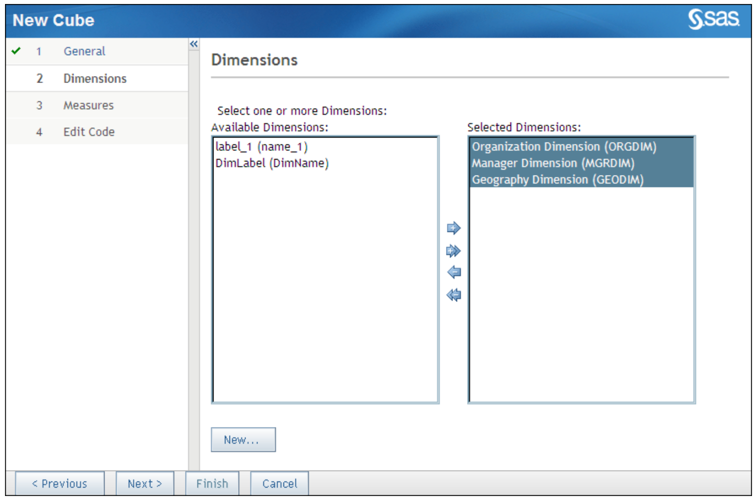
      Note: If you want to be able to run a SAS Data Integration Studio job to refresh the new cube, then the cube name needs to be all upper case.

   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.



   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.

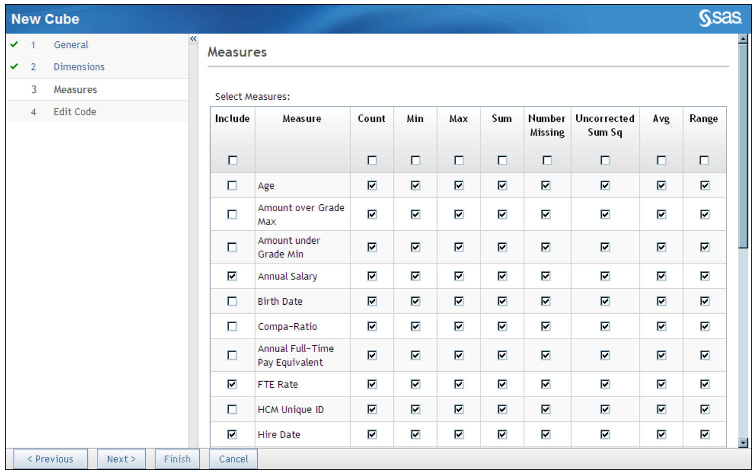
      The ORGDIM, MGRDIM, and GEODIM dimensions will display whether or not they are available in the source table. If one or more of these dimensions are selected, and the source table doesn't have the columns associated with them, they will not be included in the new cube.

      To create additional dimensions for the cube, click **New**. (See "Create a New Dimension for a Cube" on page 26.)

      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.



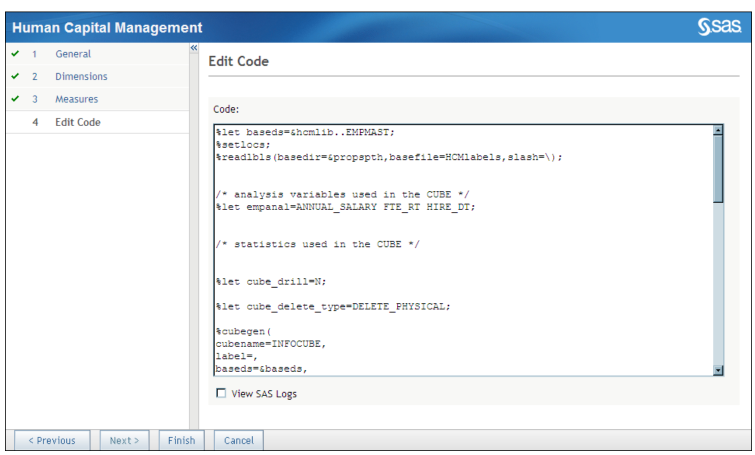
a. Select the **Include** check box for each measure you want to include in your cube.

b. For each included measure, select the check box for each statistic option 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.

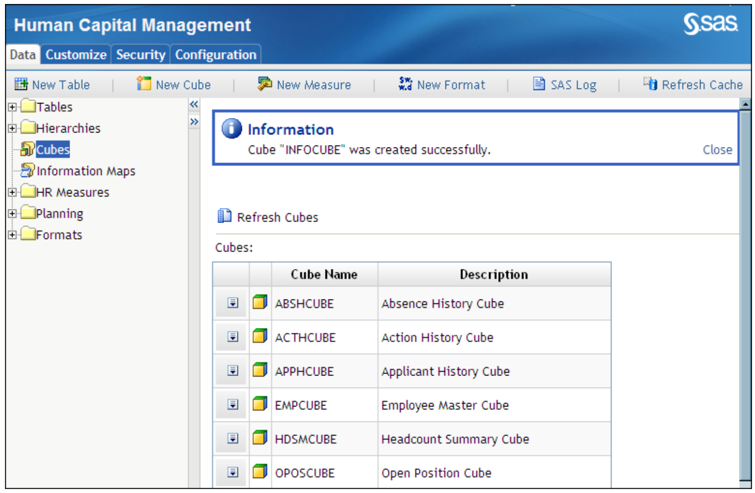


a. If your site uses double-byte character sets (DBCS), you must modify the SAS code so that the names and labels for the cube and any dimensions, measures, hierarchies, and columns do not use DBCS (DBCS are not supported in the creation of cubes).

(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.



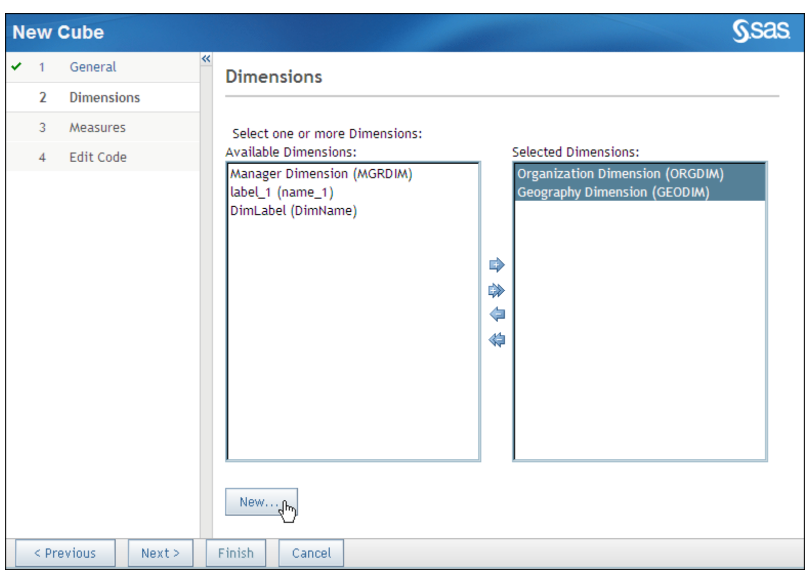
Your cube is written to the default folder location in the workspace. (See "Default Folder Locations" on page 5.)

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 28.

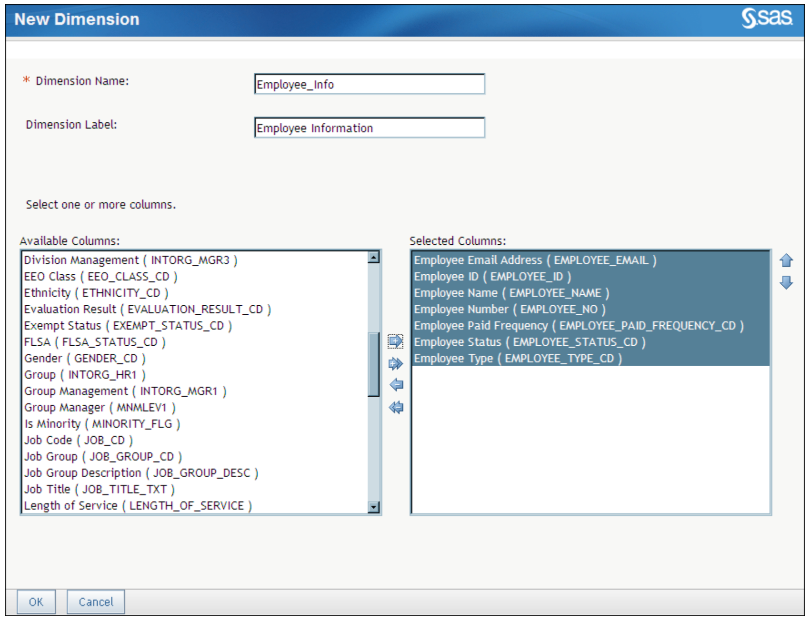
Create a New Dimension for a Cube

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.



2. Provide information for the new dimension.



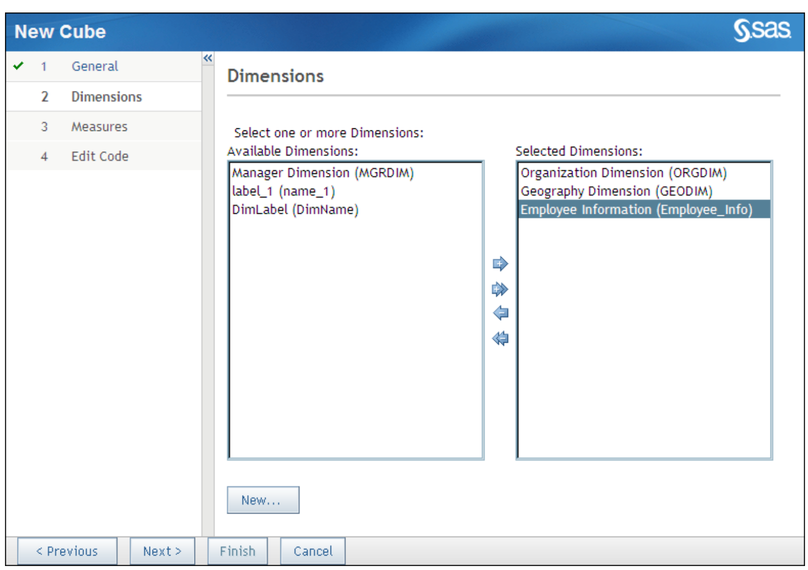
a. Provide a name for your dimension.

b. Provide a label for your dimension.

c. Select the columns 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.

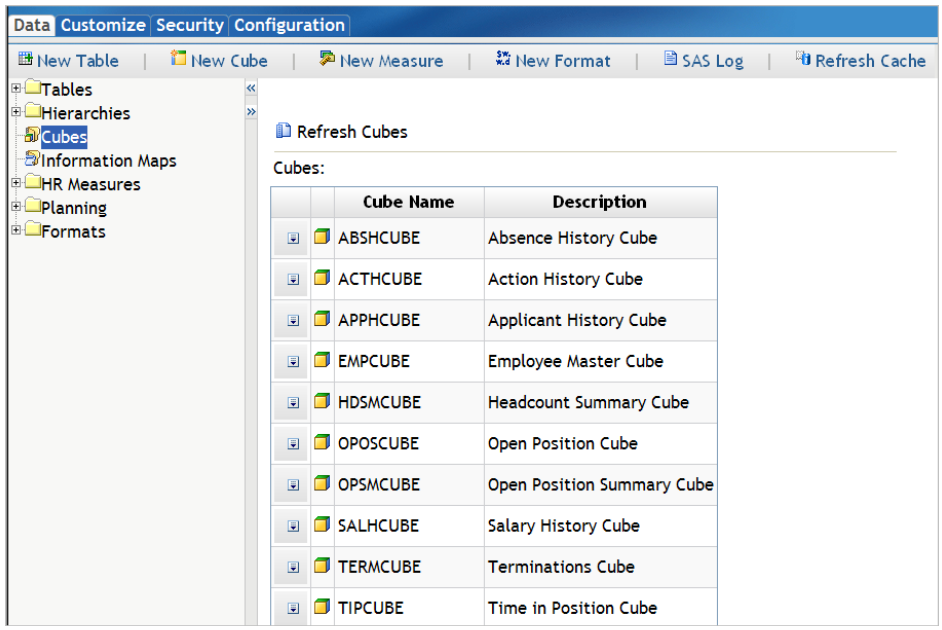


4. Complete the remaining steps in the New Cube wizard to create your cube.

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.



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 23.)

- 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.

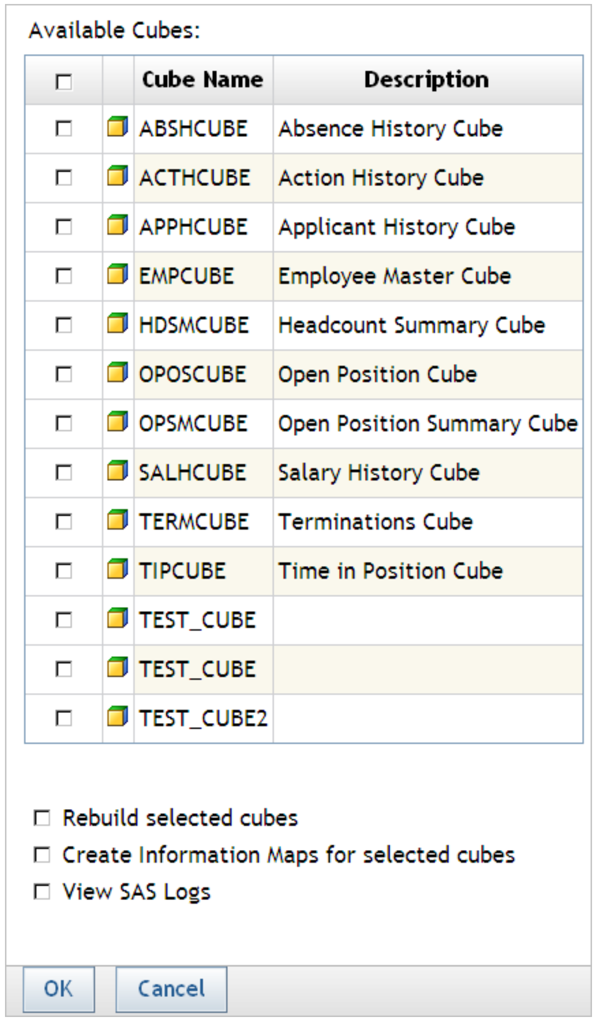
- To create an information map from a cube, select **Build Information Maps** from the action menu.

  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.



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 28.
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.

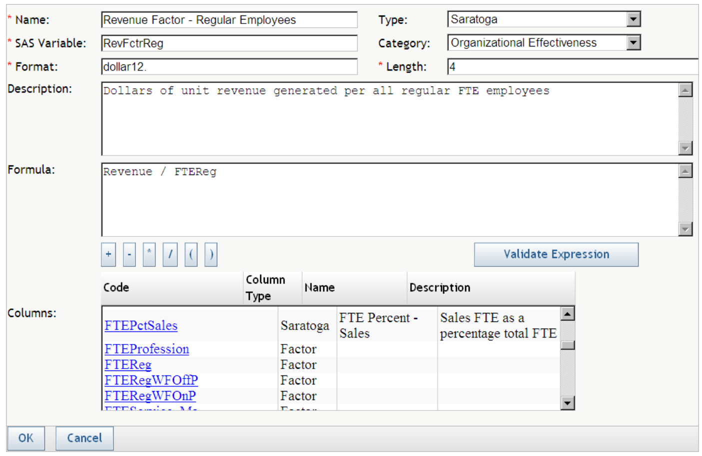
Working with HR Measures

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 161.) 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
: 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.

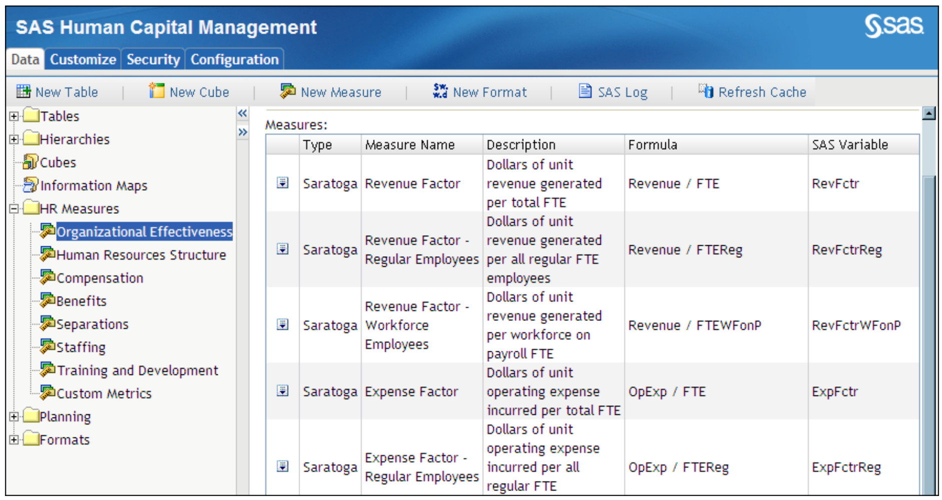
Validate Expression
: Click this button to validate the expression syntax. (This function checks only the open and closed parentheses in the expression.)

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.



For information about the fields in this display, see "Create an HR Measure" on page 30.

- 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:

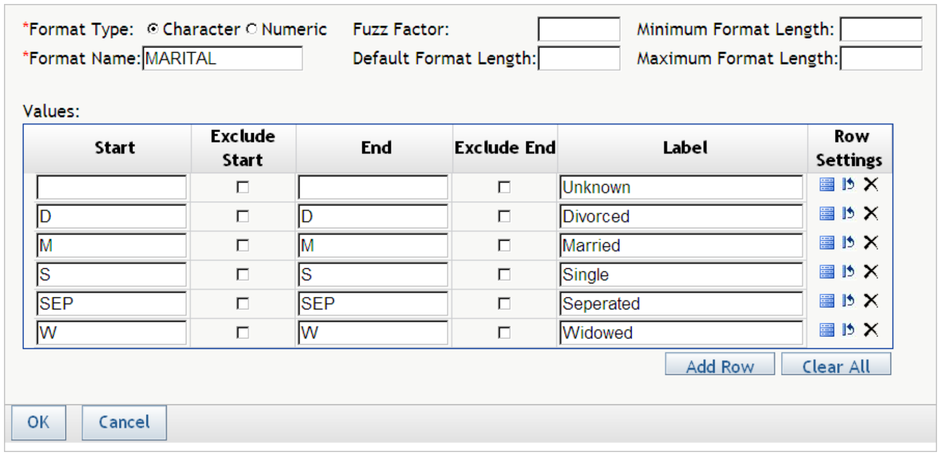
  1. Factors (in the order in which the factors were defined)
  2. Saratoga measures (in the order in which the measures were defined)
  3. HCM measures (in the order in which the measures were defined)

  (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:

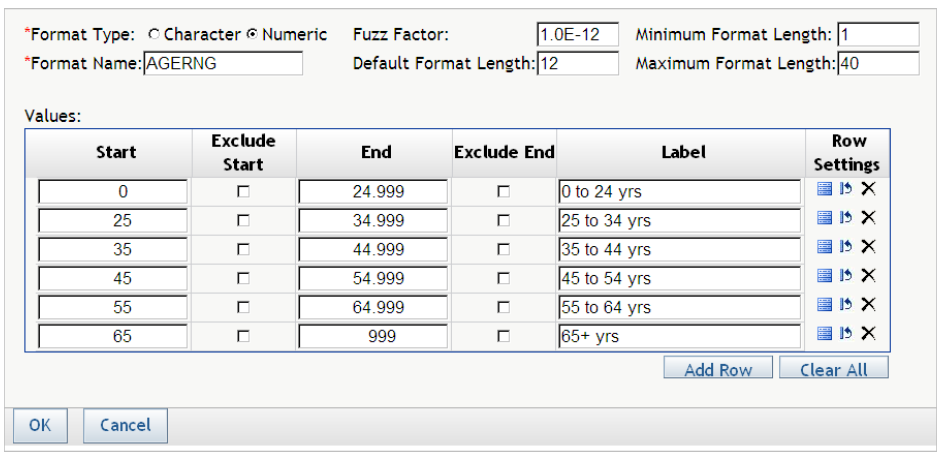
Start	Exclude Start	End	Exclude End	Label	Row Settings
	☐		☐	Unknown	
D	☐	D	☐	Divorced	
M	☐	M	☐	Married	
S	☐	S	☐	Single	
SEP	☐	SEP	☐	Seperated	
W	☐	W	☐	Widowed	

Add Row Clear All

OK Cancel

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	

Add Row Clear All

OK Cancel

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.

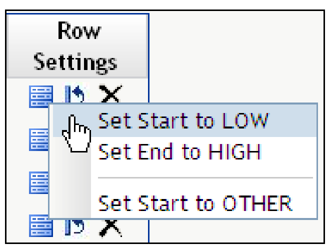
Maximum Format Length
Enter the maximum length of a label that represents a format value.

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 32.)

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

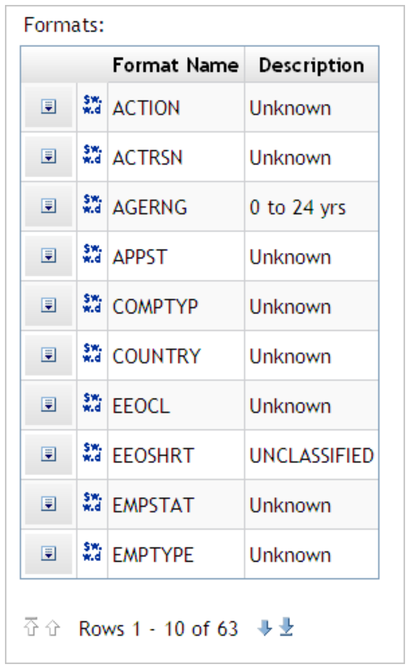
Formats are column attributes. To assign a format, see “Modify Column Attributes” on page 12.

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 33.

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.”.
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".

Format Name	Description
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”.
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".

Format Name	Description
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.
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.

Format Name	Description
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

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

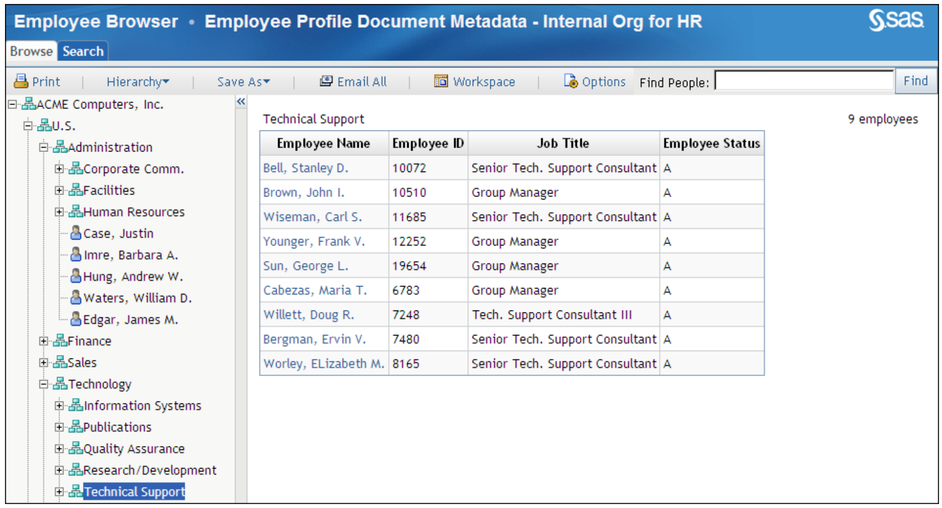
As an HCM Administrator, you assign an employee profile to a user or group by granting them ReadMetadata access to the profile and to the folder that contains the profile.

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:



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

Profile Header

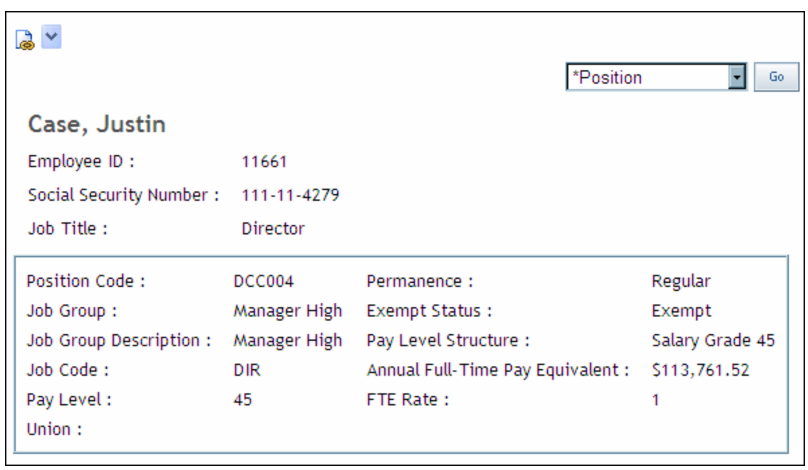
The **Profile Header** specifies the columns that are displayed in the header area of an employee profile. In this case, the header contains four columns: employee name, employee ID, Social Security number, and job title:

Employee Search

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

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:



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 47.)
- 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 42 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**.

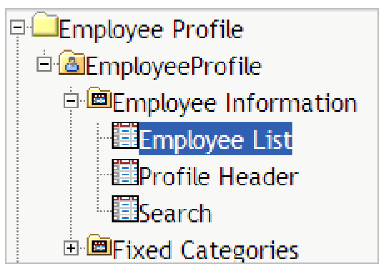
When you create an employee profile, no columns are selected for the profile header, search criteria, or fixed categories. You must edit the profile to assign these columns. Otherwise, no information will be displayed in the Employee Browser for this profile. You can also create actions (links) that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

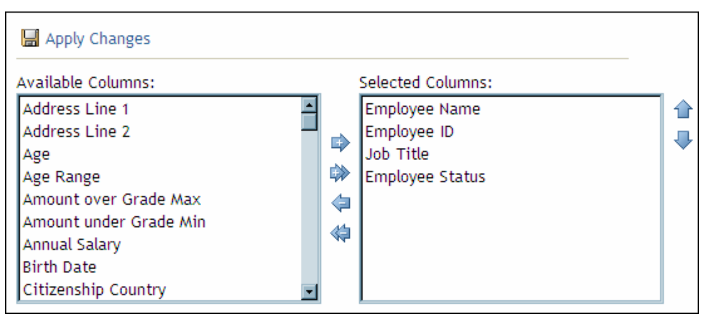
To add columns to a profile component:

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 44.



   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 5.)

Add Fixed Categories

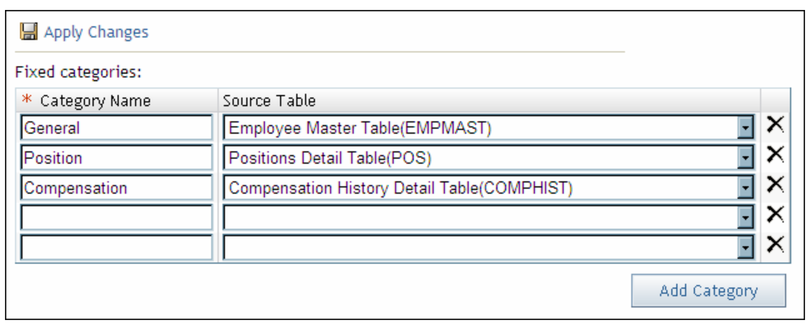
In an employee profile, the fixed categories determine the columns that are displayed when a user views employee information.

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:

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 46 to add or remove columns from the category.

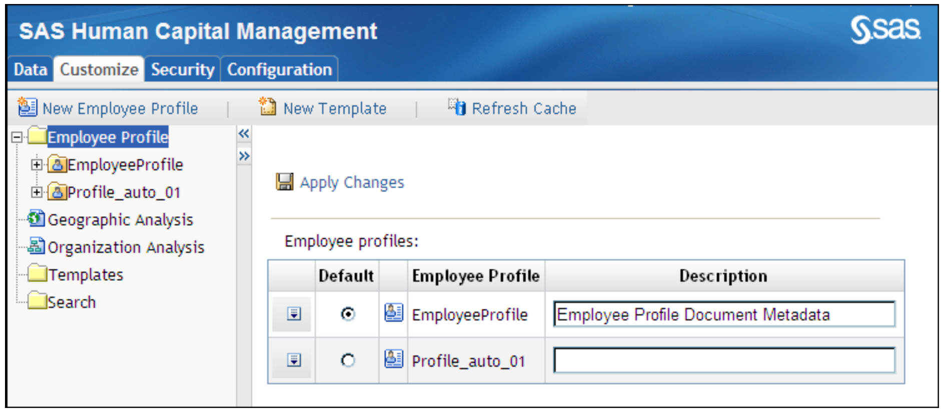
Note: On the Options page of the Employee Browser, users can add additional categories to their personal profiles. If you do not want some users to customize their profiles, you can use object security to deny access to the Options page. For more information, see "Securing Objects" on page 61.

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.



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.

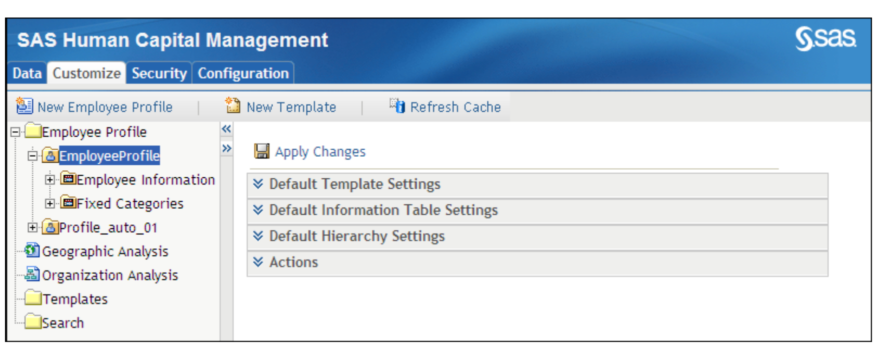
In an employee profile, you can define multiple actions, and you can specify whether they are available at the employee level, at the department level, or both.

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**.

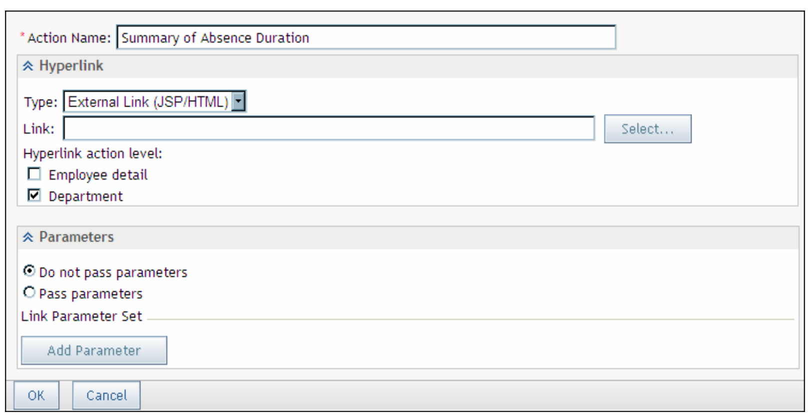
5. In the **Action Name** box, type a name for the action.

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.

   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.

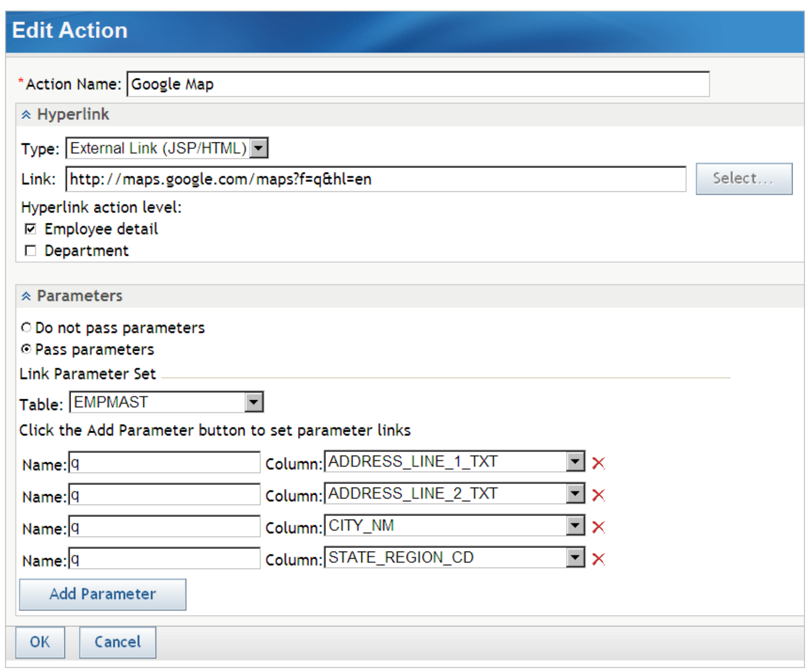
e. When you have finished adding parameters, click **OK**.

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:



Select the Default Employee Profile

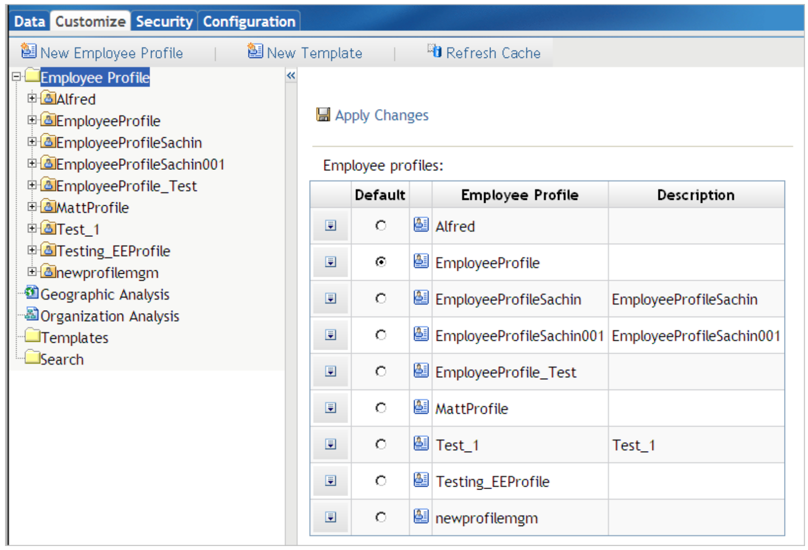
The default employee profile is displayed when a user first opens the Employee Browser. The user can then select a different profile (if other profiles are available) on the Options page of the Employee Browser.

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.



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

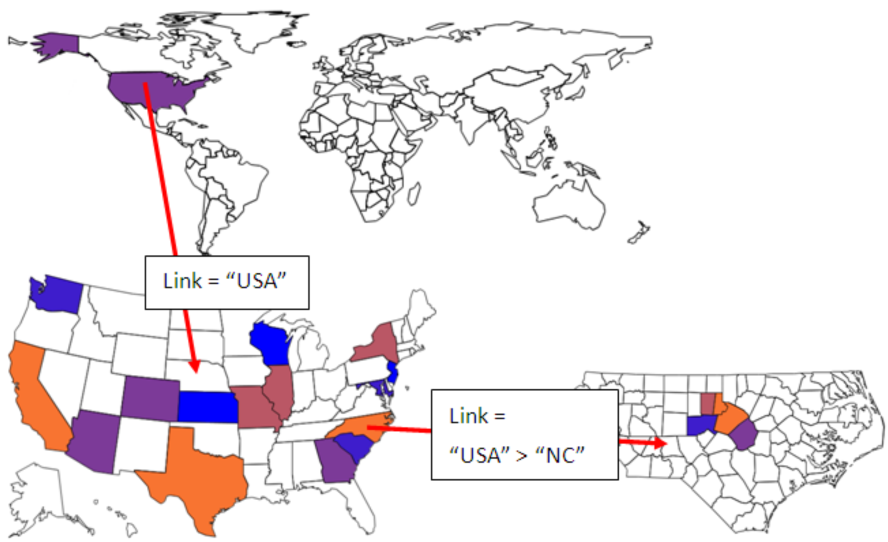
A geographic analysis displays employee information that is associated with an image map. Colored regions on the map indicate areas where employees are located. A legend on the display associates each color with a range of values that represent the number of employees. As the mouse pointer hovers over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

Level	Map Table	Link Columns
State	HRV_STATE	COUNTRY_CD STATE_REGION_CD 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.

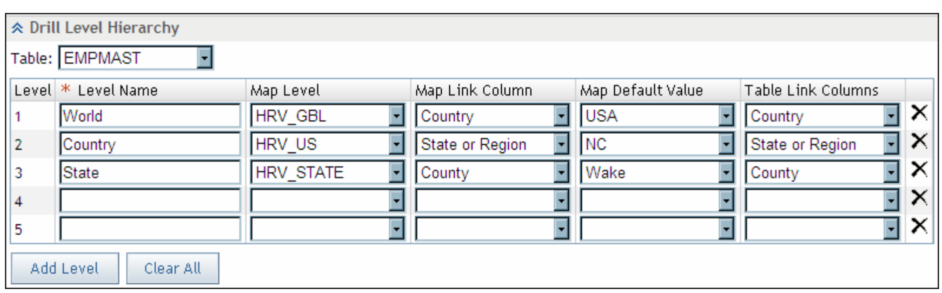
An information table contains the data that is displayed in a geographic analysis, whether in map view or table view. When users create a geographic analysis, they select an information table, which must be a master table or a history table. It must contain columns that correspond to the map link columns.

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.



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 48.

Create Maps for Geographic Analysis

The default map tables are listed in Table 3.1 on page 52. 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 9. If your SAS data set contains columns with formats attached to them, see "Importing SAS Data Sets with Formatted Columns" on page 8.
6. Refresh the cache.
7. On the **Customize** tab, modify the **Drill Level Hierarchy** section appropriately. (See "Modify the Drill Level Hierarchy" on page 53.)
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:

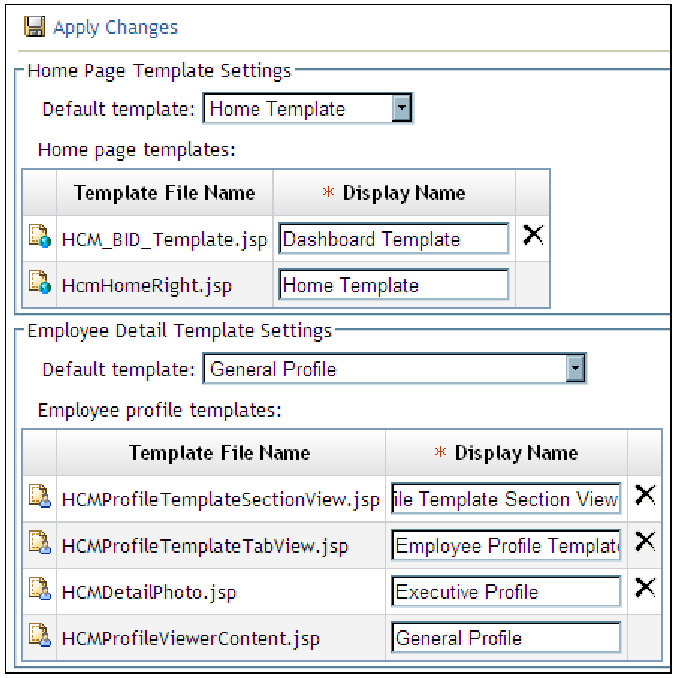
- HcmHomeRight.jsp displays the general search user interface (see “The General Search Utility” in the *SAS Human Capital Management: User's Guide*).
- HCM_BID_Template.jsp displays the general search user interface and a BI Dashboard portlet, from which users can select among available BI dashboards (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 Chapter 6, “Customizing the Employee Profile Templates,” on page 95.

Select Templates

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



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

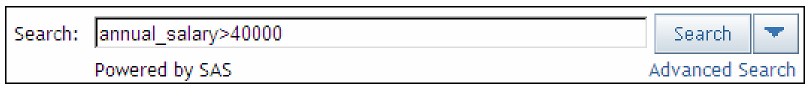
To add a template to the list of predefined templates, click **New Template**. In the New Templates dialog box, follow these steps for each template you want to define:

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 90).



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

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 datamart, 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.

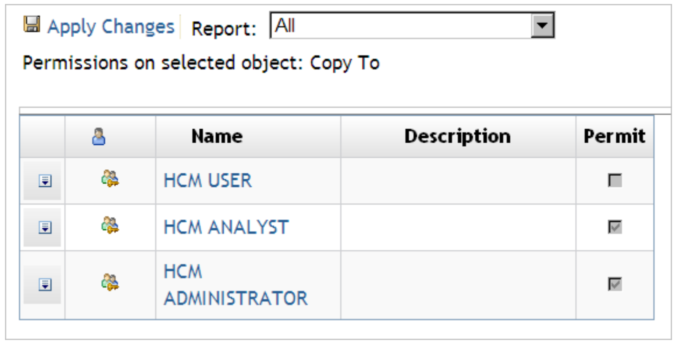
Import Users loads the SAS_USER_EMPLOYEE table with the user ID and employee ID of all active employees who are members of the HCM Solution Users group and who also have entries in the EMPGEN table. It does not reflect changes in employee status (from active to inactive, or vice versa). In addition, 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 datamart.

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.

As an 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:



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 log out, 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.

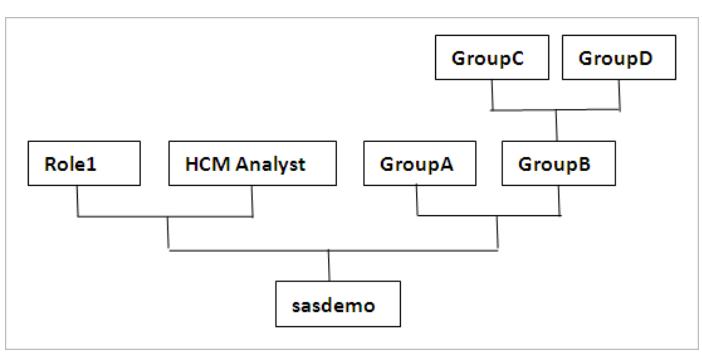
Note: Object security should not be considered equivalent to data security. Use object security to restrict the actions that a user can perform and the menu items that are available to a user. Use data security, such as row-level security or column security, to restrict the data that a user can access.

For descriptions of the objects that are included in object security, see "About Object Security" on page 151.

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 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).

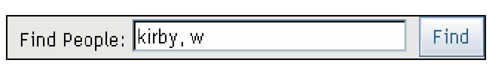
   The result becomes the GroupB permission.

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

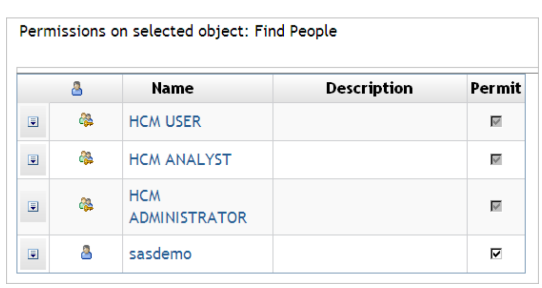
4. The direct permissions for Role1 and HCM Analyst are merged, becoming the inherited role permissions for sasdemo. Again, the most restrictive permission applies.

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:



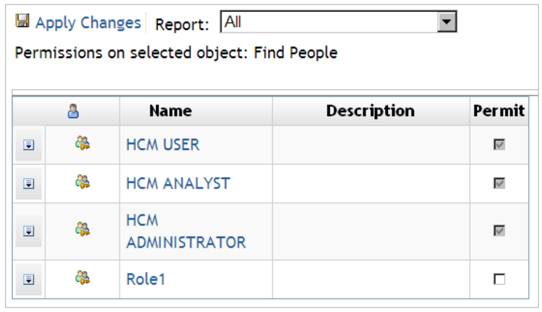
- In the first example, the sasdemo user is granted permission for the **Find People** object.

  
  Permissions on selected object: Find People

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

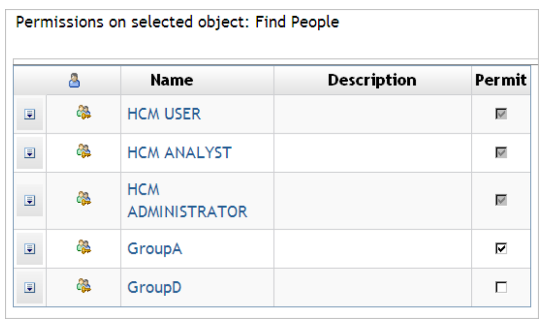
  Because the user was assigned a direct permission for the object, that permission is applied and all other permissions are ignored. As a result, the **Find People** box appears in the Employee Browser.

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



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.



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 67.)
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.

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 62.

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:

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

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

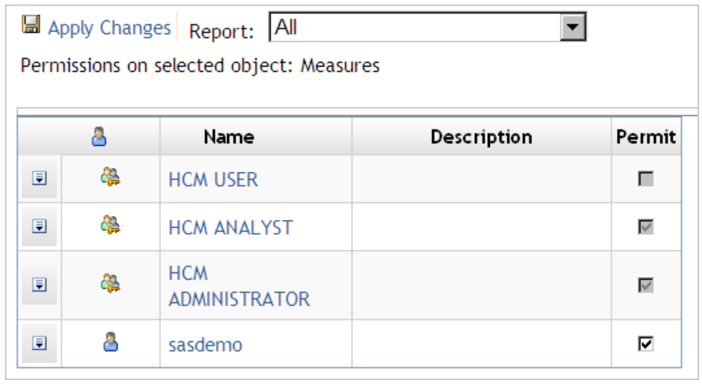
Modify Direct Permissions for an Object

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:



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 48.

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 64.

   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
   <% } %>
```

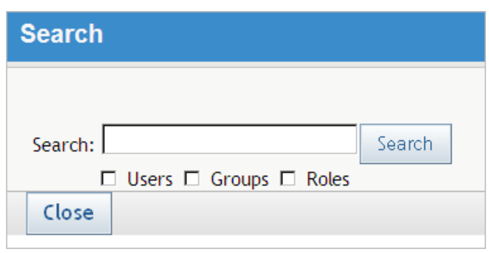
   The code to apply security to a button would be similar: embed the button code within an IF block in the JSP.

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.

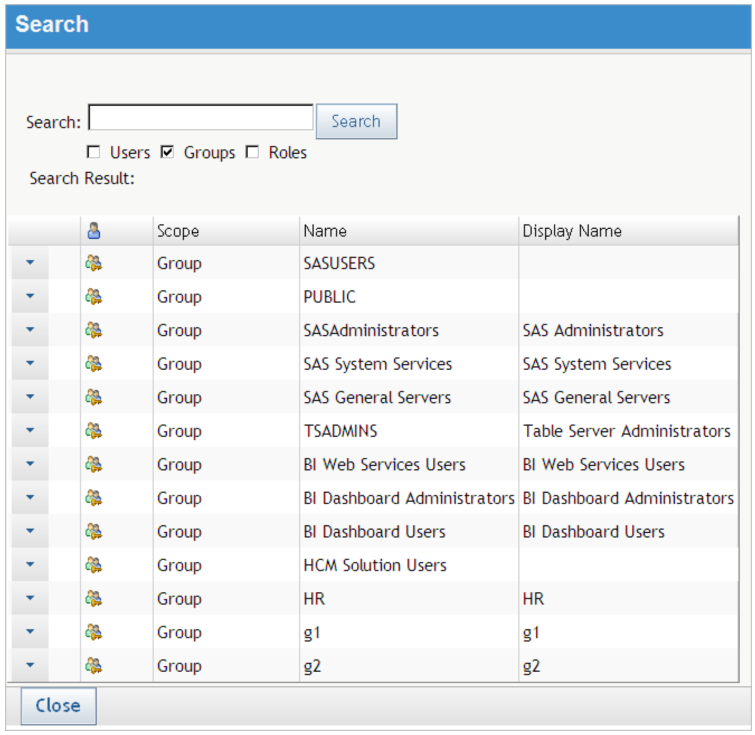
   The search string is not case-sensitive. You can enter the name of an identity (such as **sasdemo**) or its display name (such as **SAS Demo User**), or a partial search string (such as **demo**). Do not use quotation marks.

   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.

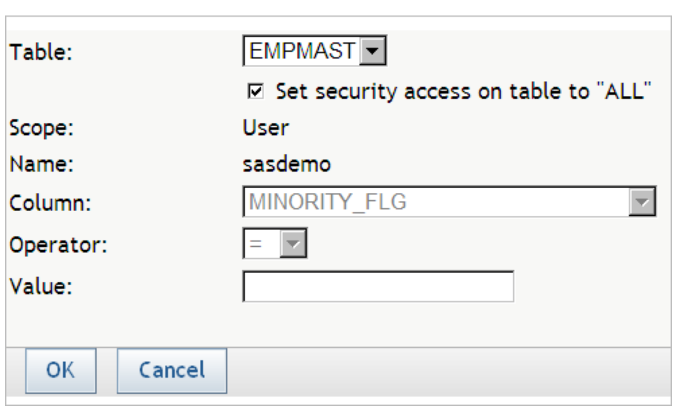


4. From the action menu beside an identity, select an action. The available actions depend on the context of the search.

Securing Table Rows

About Row-Level Security

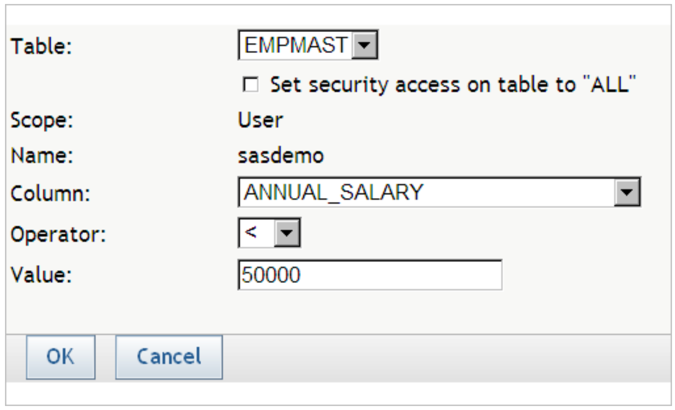
Row-level filters are one way of securing access to tables in SAS Human Capital Management. Each table has its own set of filters, each of which is associated with a user, group, or role. The most common filter grants access to all rows of a table, as in this example:



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 74) 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.



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 71.) 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 a 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 Chapter 9, "SAS for Workforce Planning & Budgeting," on page 141).
- the HCM public API (see Chapter 6, "Customizing the Employee Profile Templates," on page 95).

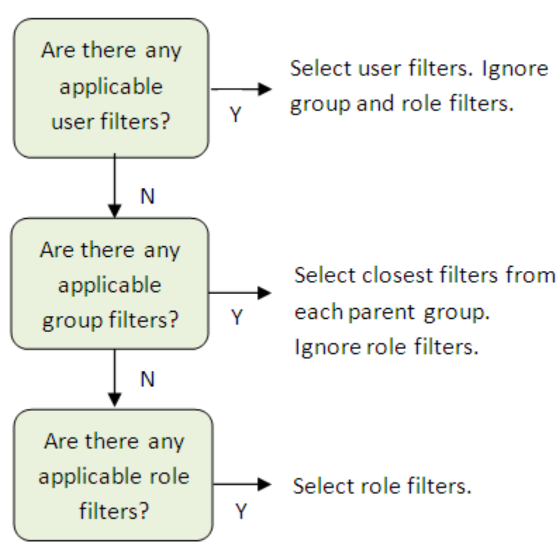
When a user tries to access a secured table, the security code retrieves the filters that are associated with that user, based on user identity and group and role memberships.

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:



   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.

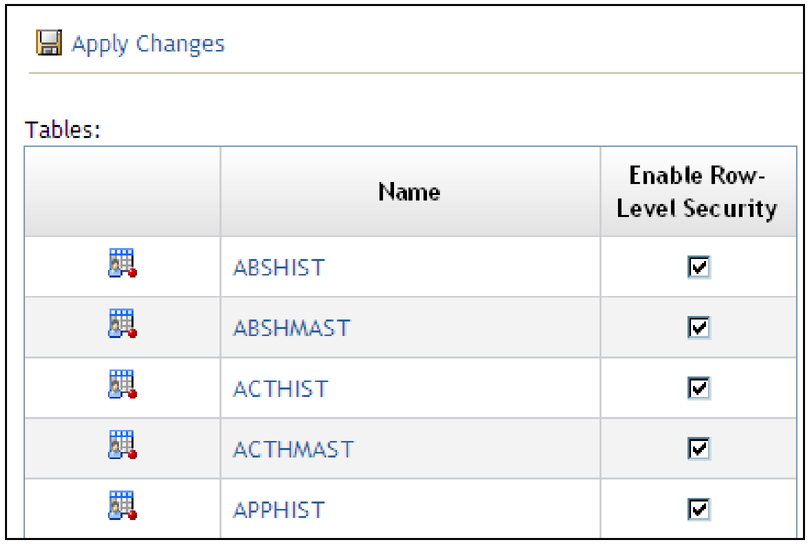
Note: You cannot disable the hierarchical filter. If the table does not contain the link field (such as INTORG_MGR), then only the employee ID is considered. For more information about hierarchical filters and the link field, see "Hierarchical Filters" on page 74.

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:



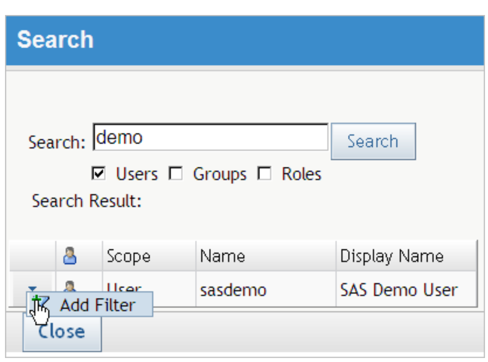
3. To enable row-level security for a table, select the check box to the right of the table name. To disable row-level security, clear the check box.
4. Click **Apply Changes**.

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 67.)
4. In the search results, select **Add Filter** from the action menu beside an identity.



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 72.

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 72.

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.

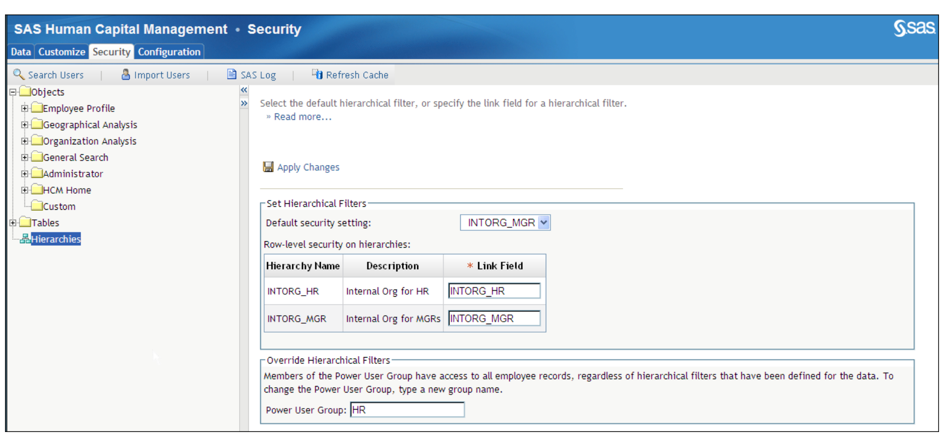
Consider the Employee Browser as an example. When users are browsing employee information, you want them to be able to view their own employee profiles. You also want managers to be able to view the profiles for their subordinates. However, you want to restrict users from viewing information about other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



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 75.

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 76.

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 about the link field, see “Select a Hierarchy” on page 74 and “Tables Without Hierarchical Filters” on page 75.

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 69.) 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 76.)

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 77.)

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 74.

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 runtime, 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

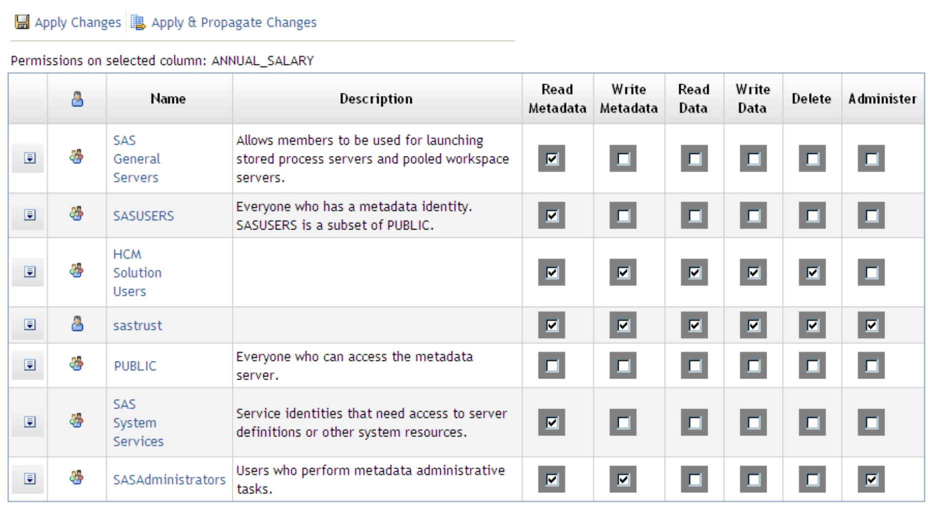
CAUTION:

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.



Apply Changes | Apply & Propagate Changes

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.

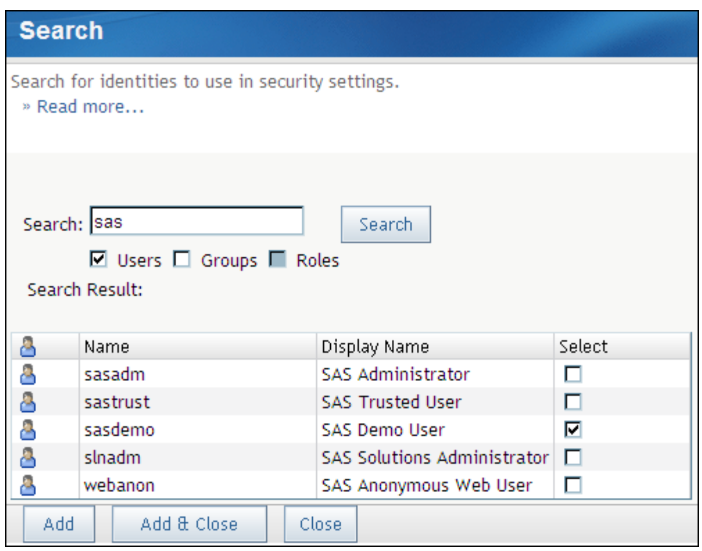
For information about the meaning of column permissions, see "About Column Security" on page 77. For information about the way permissions are applied, see "How Column Permissions Are Applied" on page 77.

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 67.)
4. From the search results, select the check box for one or more identities.



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 78 to set permissions for the identities you added.

Chapter 5

Configuring SAS Human Capital Management

Dashboards

After you create a dashboard, you can add it to the SAS Human Capital Management home page using the Administration application. Perform the following steps to add a dashboard to the home page:

1. Click **Administration** on the SAS Human Capital Management home page.
2. Click the **Configuration** tab.
3. Select **Custom Properties** ⇨ **Custom Property** from the navigation tree.
4. Provide a path in the Value column for the HCM_HOME_DASHBOARD entry. The following is an example of a path:

   ```
   /Users/sasdemo/My Folder/HCM_DASHBOARD.dcx
   ```

   Note: You will need to modify the path depending on your user name, the location of the dashboard, and the name of the dashboard.
5. Click **Apply Changes**.

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. 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 the “(Optional) Update <Files> Node of Configuration File” section in the Post-Configuration Steps chapter of the *SAS Solutions Services: System Administration Guide*. 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 Chapter 2 of the *SAS Solutions Services: System Administration Guide*, which is available at **support.sas.com/documentation/solutions/admin/index.html**.

Summary of Diagnostic Tests

The following table provides information about the diagnostic tests you can choose to run by module. For more information about selecting tests in stand-alone mode, see "Select Tests" on page 85. For more information about selecting tests in SAS Human Capital Management, see "Running the Diagnostic Utility from the Administration Application" on page 86.

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.

Test	Tab	Module	Description
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.
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 83.
3. Select the check box for each test 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 86.

Note: If you encounter errors running the Diagnostic Utility, see the *SAS Solutions Services: System Administration Guide* for help 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 *SAS Solutions Services: System Administration Guide* for help 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 on 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 83.

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:

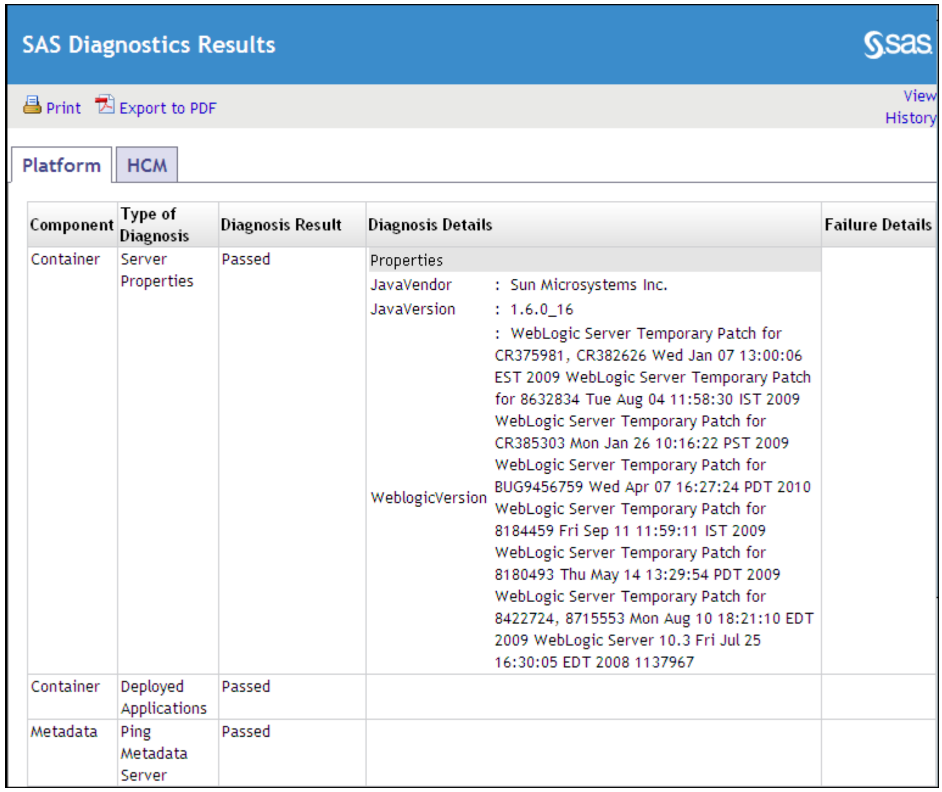
- If you encounter errors running the Diagnostic Utility, see the *SAS Solutions Services: System Administration Guide* for help resolving them.
- The Diagnostic Utility cannot be run from the SAS Human Capital Management Administration application when SAS Human Capital Management and SAS Financial 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 86.

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 60.

Viewing the Diagnostic Results

After running a diagnostic, results are grouped and presented in a table format:



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 88)
- **application properties** that apply to SAS Human Capital Management applications (see "Application Properties" on page 90)
- **custom properties** that are defined at a site (see "Custom Properties" on page 92)

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: • **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. • **HCMDefaultEEPDocumentLocation**: Default location in which employee profiles are stored. 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. In order to update a profile, a user must also have WriteMetadata permission for the folder and its contents. • **IMPORT_CONTENT_DAV**: The WebDAV location that holds documents that are imported into the workspace. • **employeeImagesPath**: The path to the employee image files, within the deployed HCM application. The default is `images/EmployeeImages`. • **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*. • **noPhotoFileName**: The file name 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. 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: • **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: • **HCMFontForNonDBCSCharacter** • **HCMFontForDBCSCharacter** 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. 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.) The following DBCS fonts are recommended: • For Chinese Simplified: STSong-Light and STSongStd-Light • For Chinese Traditional: MHei-Medium, MSung-Light and MSungStd-Light • For Japanese: HeiseiMin-W3, HeiseiKakuGo-W5 and KozMinPro-Regular • For Korean: HYGoThic-Medium, HYSMyeongJo-Medium and HYSMyeongJoStd-Medium *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. 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: • **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. • **Employee_Status_Cd_Column**: Name of the column that contains the employee status code. Read-only. • **Employee_Status_Cd_Value**: The value that signifies active employees (such as **A**). 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: • **eep_emailColumn**: The column that contains employees' e-mail addresses. • **eep_emailTable**: The table that contains employees' e-mail addresses. 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. 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. • **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. • **DefaultFixedCategoryName**: When you create an employee profile (on the **Customize** tab), a default fixed category is created. This value specifies the category name. • **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: • **search_dataSource**: Data source on which the general search is performed. The default is the Employee Master table (EMPMAST). • **search_emailColumn**: Column (within the specified data source) that contains employees' e-mail addresses. • **search_keyAttribute**: Column (such as employee ID) that is used to retrieve an employee record. • **search_linkField**: Column (such as employee name) that is used to link to an employee record. The key attribute and link attribute always appear in the search results.
Geographic Analysis Default Settingss	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:
	• **org_Measures**: The default measures to display in an organization analysis. The following values are valid:
	  • **HCNT** (headcount)
	  • **MN!*column-name*** (mean)
	  • **MAXN!*column-name*** (maximum)
	  • **MINN!*column-name*** (minimum)
	  • **NN!*column-name*** (count)
	  • **SN!*column-name*** (sum)
	Separate multiple values by spaces—for example, **AGE!MIN AGE!MAX**.
	• **org_Unassigned**: Text to display for hierarchy nodes that are not assigned to a department or manager.
	• **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 Graph Category variable
BIPBarResponse	COUNT	Bar Graph response variable
BIPBarLineCategory	INTORG_HR	Bar Line Graph Category variable
BIPBarLineResponse	COUNT	Bar Line Graph 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 Graph Category variable
BIPPieResponse	COUNT	Pie Graph 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

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.

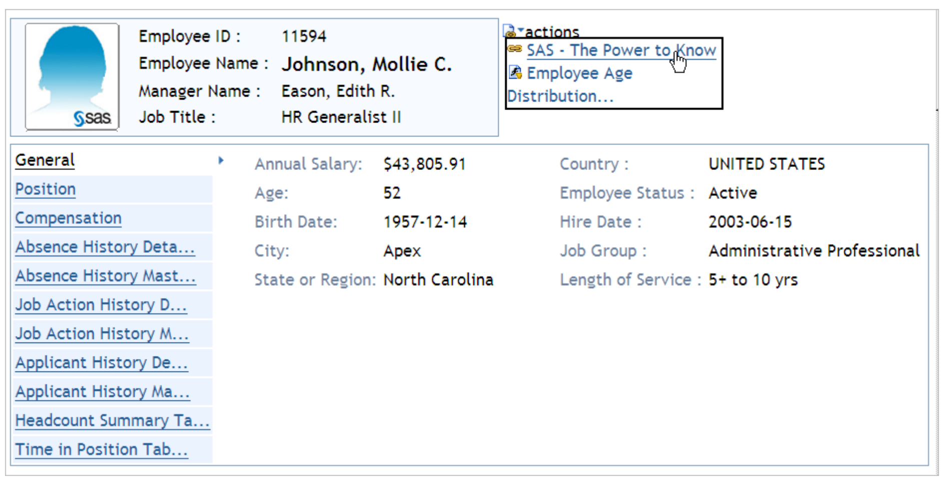
After you create or modify a template, you need to deploy it and then add it to the list of available templates. For instructions, see "Making a Template Available" on page 106.

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*



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.

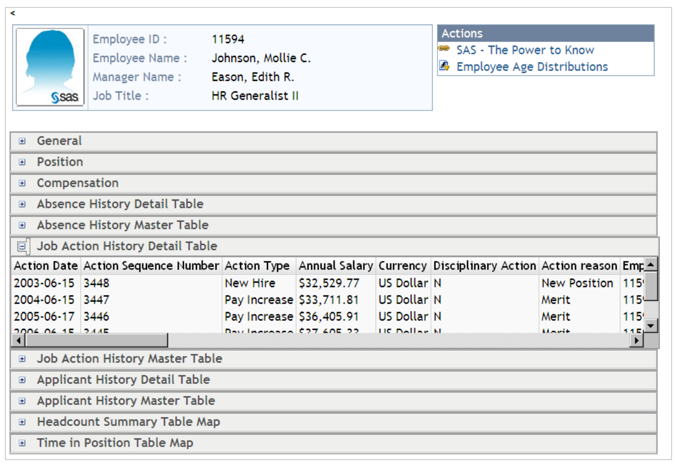
- **details for the selected category.** Notice that the profile details page contains links for several categories, including **General**, **Position**, and **Compensation**. When the user selects a category, the details (such as annual salary, age, and birth date) are displayed.

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.

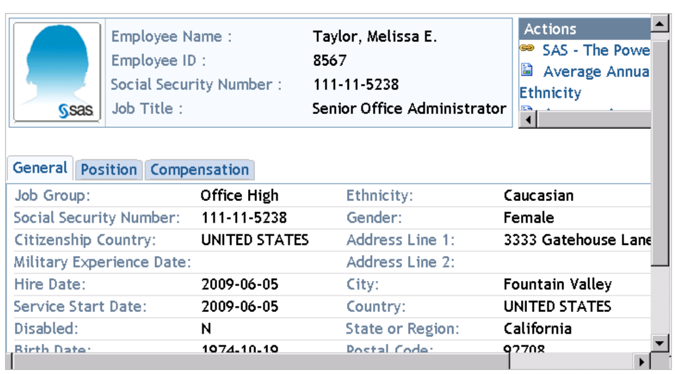
Figure 6.2 *Employee Details with the Section View Template*



- **Tab View template** (HCMProfileTemplateTabView.jsp) displays categories as a set of tabs. A user clicks a tab to display the corresponding information.

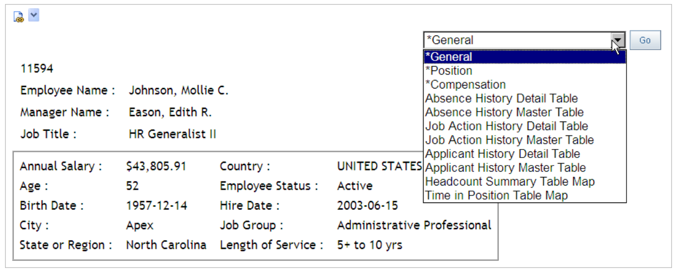
  Note: This template should be used only for a small number of categories.

Figure 6.3 *Employee Details with the Tab View Template*



- **General Profile** (HCMProfileViewerContent.jsp) uses a drop-down list for selecting the category.

Figure 6.4 *Employee Details with the General Profile Template*



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 184.

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. 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.
`public java.util.List<GenericBean> getExternalActions()`	Gets the list of external actions that are attached to this employee profile.
`public java.util.List<GenericBean> getHeaderAttributesList()`	Gets the headerAttributes.

Method	Description
`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 88.

Display the External Actions List

The example JSP gets a list of external actions from the ProfileBean, after first checking to be 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 %>">  
   <%= 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 upon 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.

Method	Description
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.
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.

Method	**Description**
`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 upon 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 Appendix A3, “The Public API,” on page 183.

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 90 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 56. After a template is available, it can be assigned to an employee profile. See "Create an Employee Profile" on page 44.

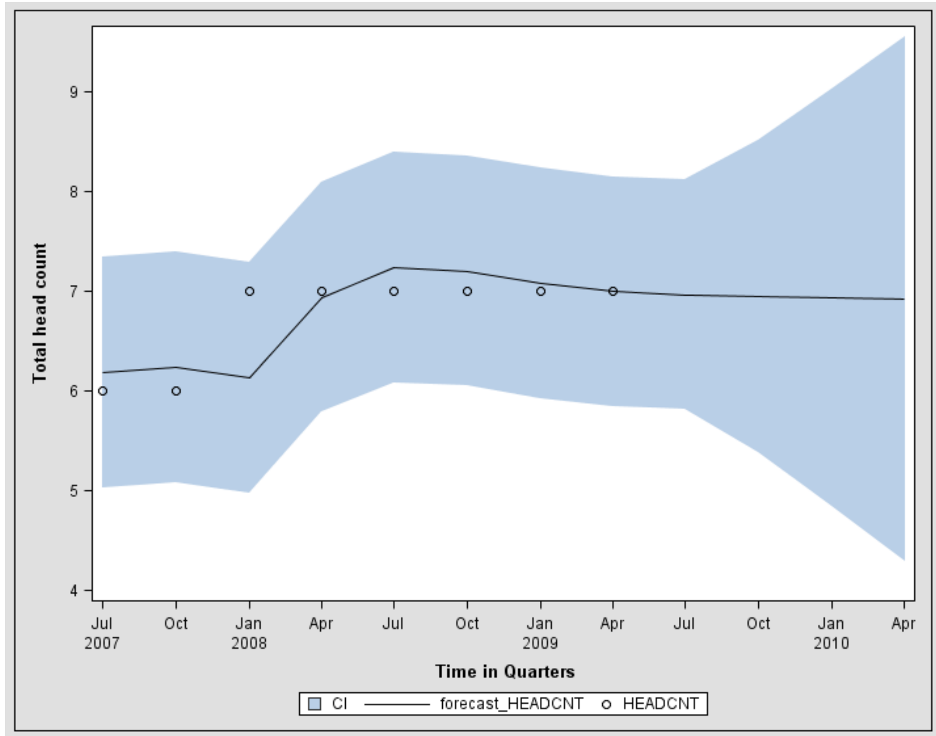
Chapter 7

Forecasting in SAS Human Capital Management

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.

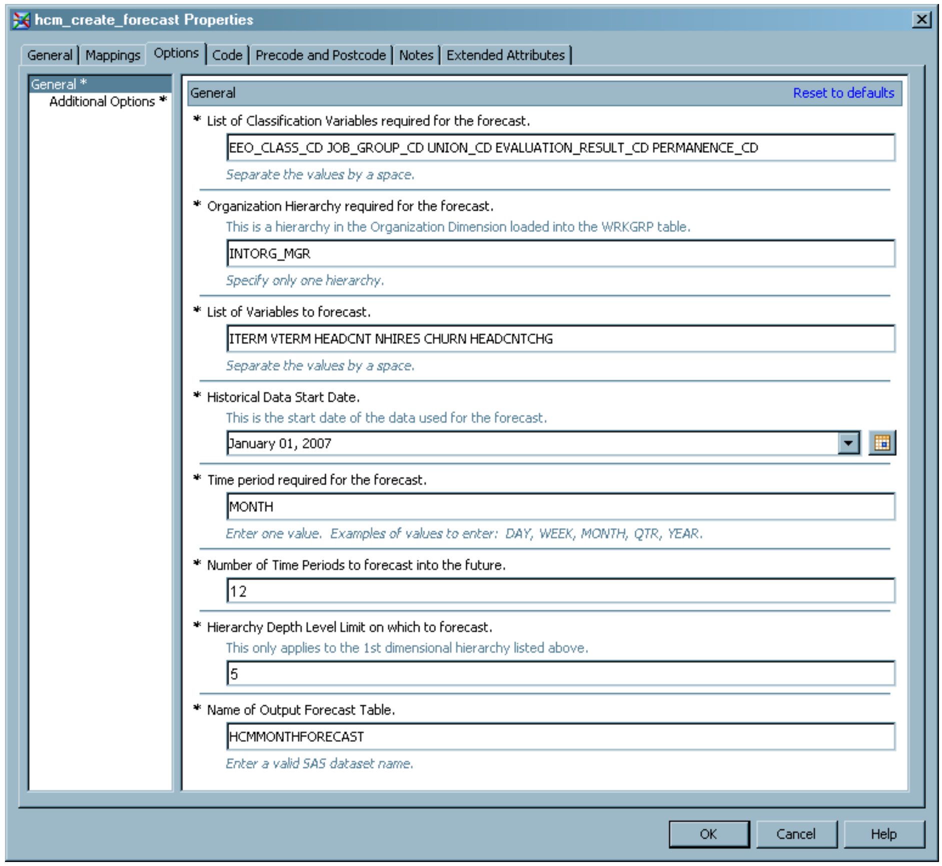
Preparing the Data

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

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.

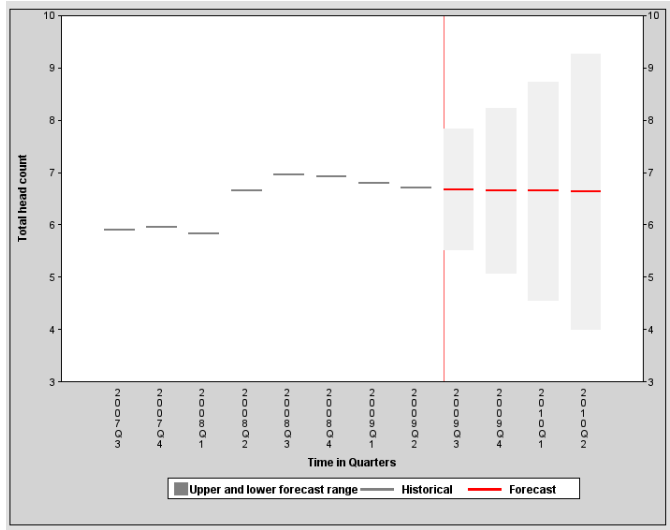
The default parameters are as follows:

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.

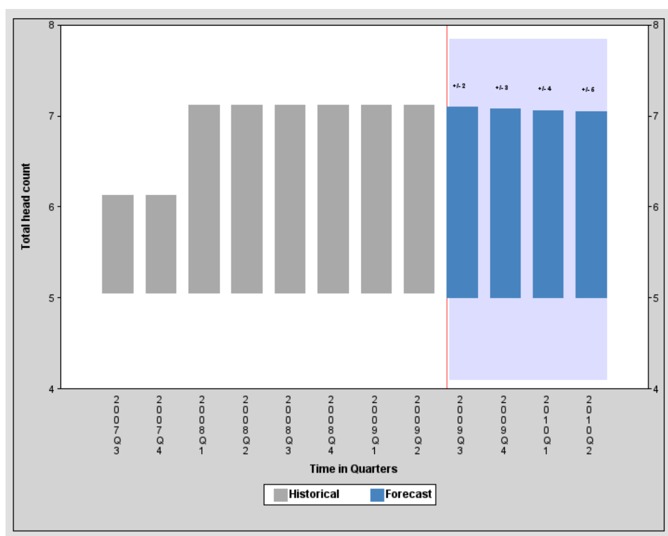
HCMGRAPHTYPE_VAR
specifies the type of graph in which to convey the historical and projected values of the forecast variable. The historical data is displayed to the left of the red line; the forecast data, to the right of the red line. The gray area represents the confidence interval.

By default, these selections are available:

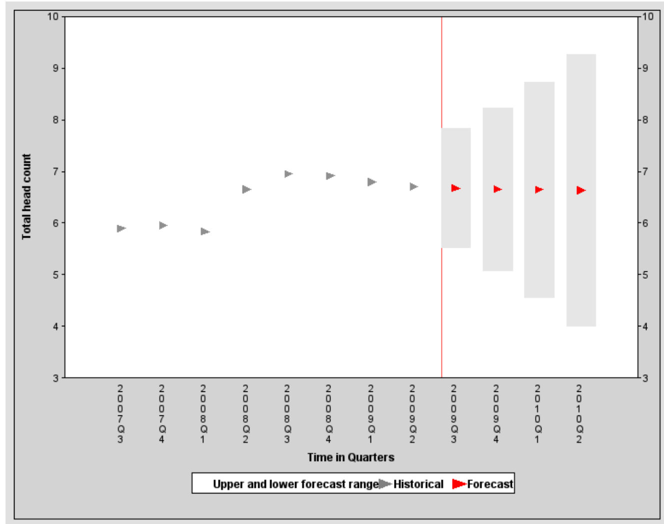
- `dlines` (Floating Lines) specifies a modified line graph. Confidence intervals are displayed as gray bar areas above and below the forecasted line.



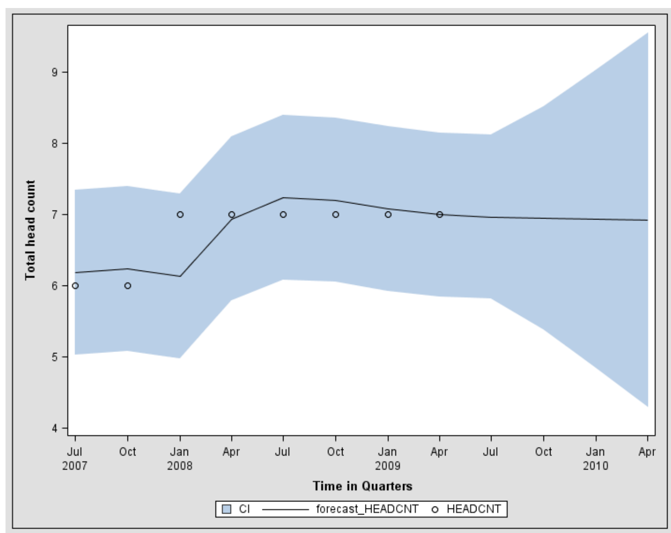
- **`bbars`** (Vertical Bars) specifies a traditional histogram. Confidence intervals can be displayed on the Time axis or on top of the histogram.



- **`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 116.

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.

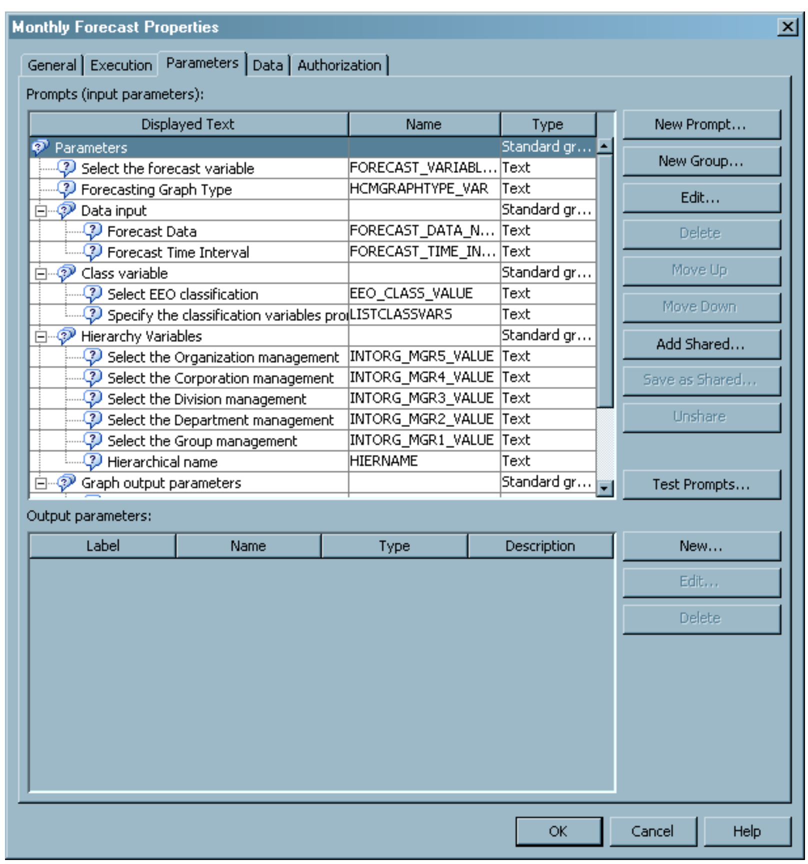
graphheight
specifies the height of the graph in pixels.

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.



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.

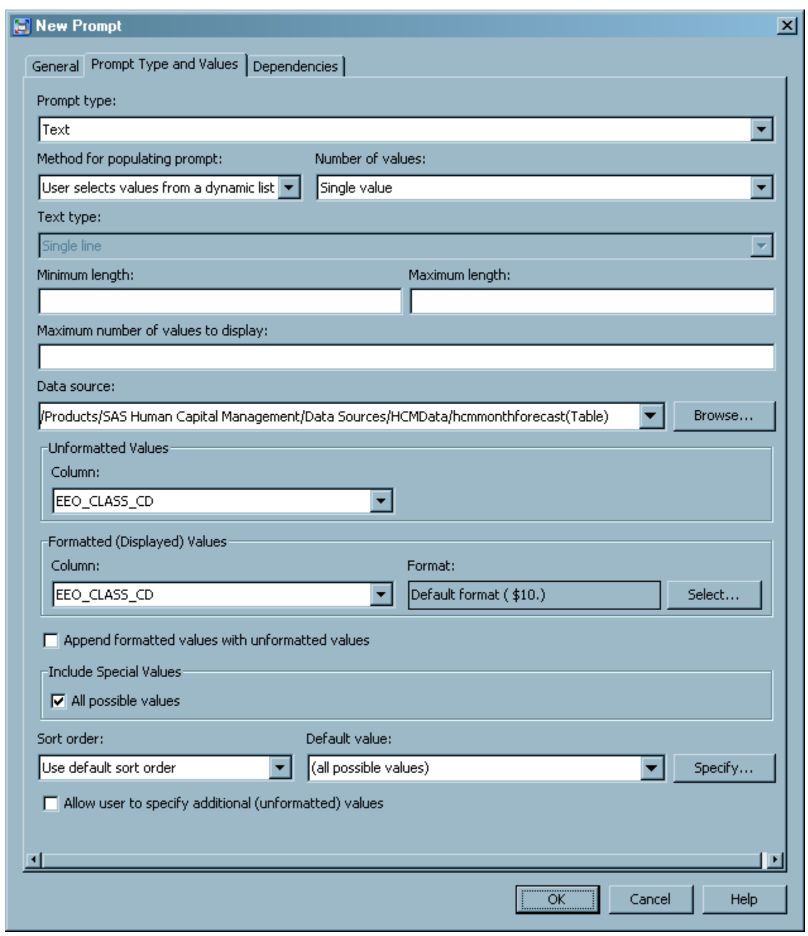
5. Select the **Prompt Type and Values** tab in the Edit Prompt dialog box, and specify available classification variables for a user to select in the **Default value** text box. Be sure to separate each value by a space. The column names of the classification variables that you specify in the **Default 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 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.



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 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 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 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

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 123. For a list of input tables, see "Required Tables" on page 124.

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 129.

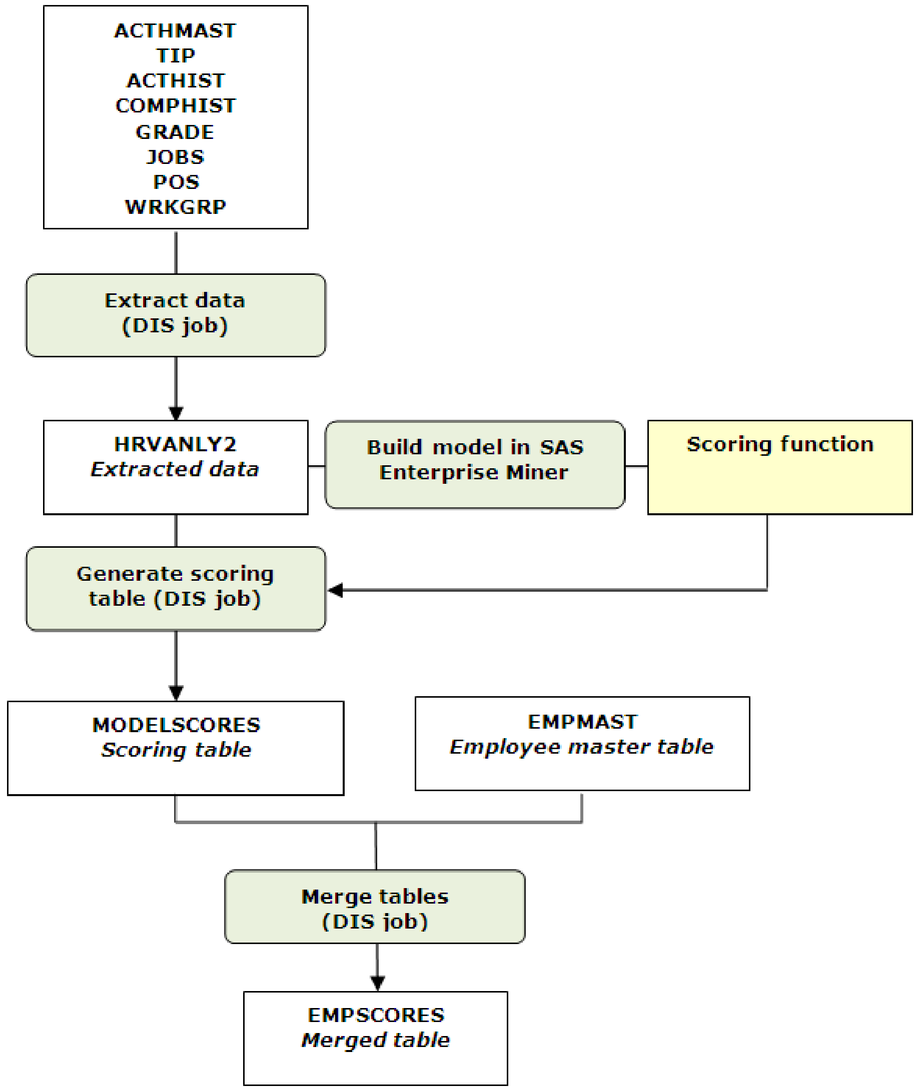
3. The merge stage. You merge the scoring table with the Employee Master table (EMPMAST) to generate a table (EMPSCORES) that combines employee data with predicted termination scores.

   You (or perhaps a power user) create cubes and reports from the merged table.

   See "Working with the Results" on page 137.

The following diagram illustrates the overall retention analysis process:

Figure 8.1 *Retention Analysis Process*



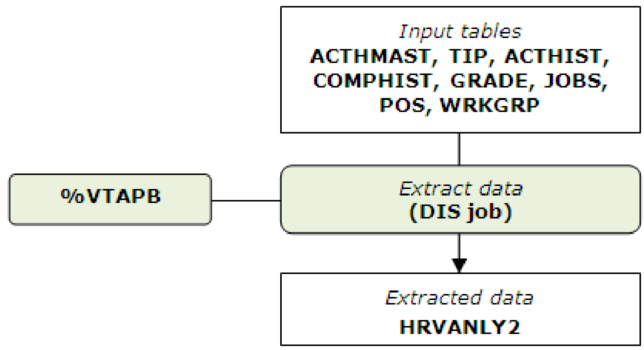
At various times in the future, the site might want to perform a new analysis. With only minor customizations (such as dates) to the macro variables, the site can re-run the entire process and generate a new scoring table, using the existing scoring function.

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 124.
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 127.
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 runtime, 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. Be 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
_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.
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: `%let vt_crit=(%str(put(ACTION_TYPE_CD,$iaction.)='VTERM'));`
NVT_CRIT	Defines the involuntary termination criteria as a valid DATA step statement that can be processed against ACTHMAST. Example: `%let nvt_crit=(%str(put(ACTION_TYPE_CD,$iterm.)='TERMS') and %str(put(ACTION_TYPE_CD,$iaction.) ne 'VTERM'));`

ASUBVARS	(Optional) Defines any non-standard 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 125.
	For example, assume that the site defines voluntary termination as a combination of ACTION_TYPE_CD and MYREASON (a non-standard column):
	`%let vt_crit = %str((put(ACTION_TYPE_CD, $iaction.)='VTERM')) and myreason = 'T');`
	In that case, you would assign asubvars as follows:
	`%let asubvars = myreason;`

PRO_CRIT	(Optional) Defines promotion criteria as a valid DATA step statement that can be processed against ACTHMAST. Example: `%let pro_crit=(%str(ACTION_TYPE_CD='DPRO'));`

Variable	Description
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: `%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: `%let shavars=ANNUAL_SALARY TOTOTHER PAY_LEVEL_STRUCTURE_CD CHGAMT CHGPCT JOB_GROUP_CD;`
AHAVARS	Defines the time-dependent analysis columns from the ACTHMAST table. Example: `%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: `%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: `%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: `%let medvars=CHGAMT RNG_PENE;`

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:
	`%let medvars=ANNUAL_SALARY CHGPCT;`
	`%let medgrp=JOB_GROUP_CD;`
	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. As an 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

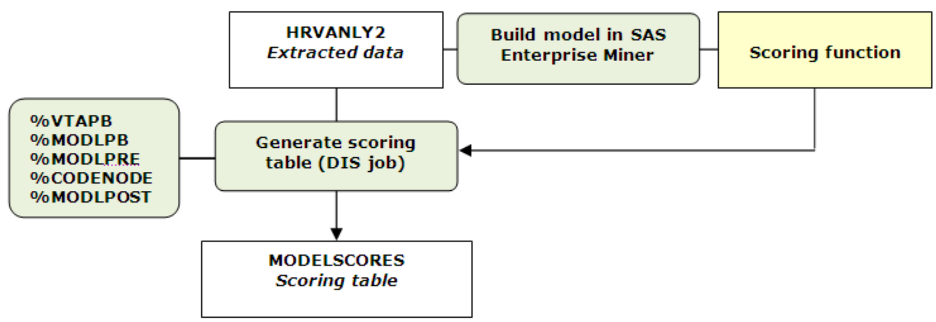
Overview of Transformation

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 132.) 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 131.)

   You will have already modified the %VTAPB macro during the extraction process. For details, see "Modify the %VTAPB Macro" on page 127.

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 135.

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 135.

4. Make any necessary customizations to the %MODLPOST macro.

   For details, see "Customize the %MODLPOST Macro" on page 136.

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 136.

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: `%let hrva2loc=&hcmlib;` 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. 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). Example: `%let beginyr=2004;`

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).
	If you leave the value of ENDYR blank, it defaults to the previous year (**`year(today())-1`**). Example:
	`%let endyr=;`
	Using the default simplifies future maintenance. Otherwise, you must update the ENDYR value each year.
	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.
	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:
	`if vterm_f=1 and lstcondt gt mdy(1,1,&endyr.)`
	`  then vterm_use=1;`
	`  else vterm_use=0;`
	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.

Variable	Description

MODELSUB	Defines a population subset for the analysis. For model creation, the typical assignment is as follows:
	`%let modelsub=where HIRE_DT le mdy(1,1,%eval(%sysfunc(year(%sysfunc(today())))-1));`
	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.
	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:
	`%let modelsub=where HIRE_DT le mdy(1,1, &endyr.);`
	(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: `%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`**).
_SCORE	Defines the input table for the SAS Enterprise Miner program. The default is as follows: `%let _score=&pre_data.;`
_PREDICT	Defines the output table from the SAS Enterprise Miner program. The default is as follows: `%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).

<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td>VTGRP</td><td>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:
• <code>PROB</code> (the default) bases the categorization on the voluntary termination probabilities.
• <code>SCORE</code> bases the categorization on the scores that are derived from the voluntary termination probabilities.
• <code>RANK</code> bases the categorization on the RANK procedure, which divides the voluntary termination probabilities into three equal groups.</td></tr>
<tr><td>VTCUTS</td><td>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).
This example (for the PROB method) defines low risk as less than or equal to 10%, and high risk as greater than 25%:
<code>%let vtcuts = 0.1 0.25;</code>
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:
<code>%let vtcuts = 300 400;</code></td></tr>
<tr><td>TRANSVARS</td><td>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.</td></tr>
<tr><td>MODELVARS</td><td>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:
<code>%let modelvars=CHGAMT_CUR RNG_PENE06 YOS_ACT;</code></td></tr>
</table>

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 file name 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 upon 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 on 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. Check to be 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).
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. 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.

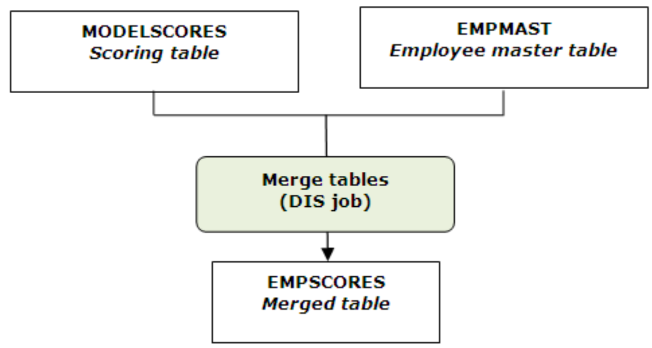
Working with the Results

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 12.

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 68.

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 23.

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

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 kinds 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 kinds 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 runtime, 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 a New 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, imagine 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.

Apply Changes SAS Code

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 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 146.

   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 143.)

   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 146.)

Calculation
The **Calculation** drop-down list defines a calculation that takes place on this measure at runtime. (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 runtime 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 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 147.)

   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, be 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	36,699.95	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.	1/2/2007	27,055.98	7/1/2009	3.50	2.63
Cooper, Nellie C.	9/30/2002	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		268,738.92			

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 Runtime

At runtime, 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**.

Appendix 1

Object Security: List of Objects

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 61.

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. 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. **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.

Object	Description
Generate Symbol	Generate a symbol for a column.
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.
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.

Object	Description
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.
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. The **Advanced Search** object of the General search options must be enabled. 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. 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. 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). 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. 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. 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. 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

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

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 95.

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 192	This is the parent-level package. It contains key public API interfaces.
com.sas.solutions.hcm.publicapi.beans on page 192	Provides Public API classes that act as beans for holding data.
com.sas.solutions.hcm.publicapi.connector on page 192	Provides Public API factory classes.
com.sas.solutions.hcm.publicapi.exceptions on page 192	Provides classes for exceptions in Public API.
com.sas.solutions.hcm.publicapi.models on page 193	Provides Public API model classes.

Hierarchy for all Packages

Package Hierarchies

The following package hierarchies are available:

- com.sas.solutions.hcm.publicapi on page 193
- com.sas.solutions.hcm.publicapi.beans on page 193
- com.sas.solutions.hcm.publicapi.connector on page 193
- com.sas.solutions.hcm.publicapi.exceptions on page 194
- com.sas.solutions.hcm.publicapi.models on page 194

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 194
- “Class GenericBean” on page 196
- “Class GenericResultBean” on page 198
- “Class PublicAPITableModel” on page 199
- “Class ProfileBean” on page 200
- “Class PublicAPIFactory” on page 203
- “Class HCMPublicAPIException” on page 203

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 204

API Index

Overview

This index contains an alphabetical list of all classes, interfaces, constructors, methods, and fields.

C

G

`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 206 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 207 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 208 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 209 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 203 for more information.

`getEmpPhotoSrc()`
: Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the source of employee photograph. See “getEmpPhotoSrc” on page 202 for more information.

`getExternalActions()`
: Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the list of external actions. See “getExternalActions” on page 202 for more information.

`getHeaderAttributesList()`
: Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Gets the headerAttributes. See “getHeaderAttributesList” on page 202 for more information.

`getHeaderDetails(HttpServletRequest)`
: Method in interface com.sas.solutions.hcm.publicapi.PublicAPIInterface. Returns employee profile's header details. See “getHeaderDetails” on page 209 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 210 for more information.

`getLabel()`
: Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the categoryLabel. See “getLabel” on page 196 for more information.

`getLabel()`
: Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See “getLabel” on page 197 for more information.

`getName()`
: Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Gets the categoryName. See “getName” on page 196 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 200 for more information.

ProfileBean()

Constructor for class com.sas.solutions.hcm.publicapi.beans.ProfileBean. See Table A3.18 on page 200 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 203 for more information.

PublicAPIFactory()

Constructor for class com.sas.solutions.hcm.publicapi.connector.PublicAPIFactory. See Table A3.21 on page 203 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 204 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 199 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 199 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 202 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 195 for more information.

setColumnNameList(List)

Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See “setColumnNameList” on page 199 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 196 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 202 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 202 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 202 for more information.

`setHeaderAttributesList(List<GenericBean>)`
Method in class com.sas.solutions.hcm.publicapi.beans.ProfileBean. Sets the headerAttributes. See "setHeaderAttributesList" on page 201 for more information.

`setLabel(String)`
Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the categoryLabel. See "setLabel" on page 196 for more information.

`setLabel(String)`
Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setLabel" on page 198 for more information.

`setName(String)`
Method in class com.sas.solutions.hcm.publicapi.beans.CategoryBean. Sets the categoryName. See "setName" on page 196 for more information.

`setName(String)`
Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setName" on page 197 for more information.

`setRowDataList(List)`
Method in class com.sas.solutions.hcm.publicapi.beans.GenericResultBean. See "setRowDataList" on page 199 for more information.

`setValue(String)`
Method in class com.sas.solutions.hcm.publicapi.beans.GenericBean. See "setValue" on page 197 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 204	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 194	This bean contains category details such as category name, category label, and category attributes for fixed and custom categories.
GenericBean on page 196	This is a generic bean that contains name, value, and label.
GenericResultBean on page 198	This bean contains column name list and row data.
ProfileBean on page 200	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 203	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 203	This is a wrapper exception to hold any other exception that might get generated from HCM code.

Package com.sas.solutions.hcm.publicapi.models

Provides Public API model classes.

Table A3.5 *Class Summary*

PublicAPITableModel on page 199	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()** Gets the list of attributes of the selected fixed category.

javax.swing.table.DefaultTableModel	**getCustomCategoryAttributesTableModel ()** Gets the table model of the selected custom category.
java.lang.String	**getLabel()** Gets the categoryLabel.
java.lang.String	**getName()** Gets the categoryName.
void	**setCategoryAttributesList(java.util.List<GenericBean> categoryAttributesList)** Sets the list of attributes of the selected fixed category.
void	**setCustomCategoryAttributesTableModel (javax.swing.table. DefaultTableModel customCategoryAttributesTableModel)** Sets the table model of the selected custom category.
void	**setLabel(java.lang.String label)** Sets the categoryLabel.
void	**setName(java.lang.String name)** 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 additionally provides column information, such as column labels from the underlying table model.

Table A3.15 *Constructor*

`PublicAPITableModel(HcmDefaultTableModel tableModel)` 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)`** Gets the column label of a column with the passed index.
`java.lang.Object[]`	**`getColumnInfoNames()`** 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()** Gets the categories list with their attributes.

java.lang.String	**getEmpPhotoSrc()** Gets the source of employee photograph.
java.util.List<GenericBean>	**getExternalActions()** Gets the list of external actions.
java.util.List<GenericBean>	**getHeaderAttributesList()** Gets the headerAttributes.
boolean	**isCustomCategorySelected()** Returns whether selected category is fixed or custom.
void	**setCategoriesList(java.util.List<CategoryBean> categoriesList)** Sets the categories list with their attributes.
void	**setCustomCategorySelected(boolean isCustomCategorySelected)** Sets the selected category's type.
void	**setEmpPhotoSrc(java.lang.String empPhotoSrc)** Sets the source of employee photograph.
void	**setExternalActions(java.util.List<GenericBean> externalActions)** Sets the list of external actions.
void	**setHeaderAttributesList(java.util.List<GenericBean> headerAttributesList)** 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()** 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 get generated from HCM code.

See Also: "Serialized Form" on page 211.

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)** 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 configured default table.
GenericResultBean	**getEmployeeDetails(java.lang.String tableName, java.lang.String employeeId, HttpServletRequest request)** 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)** Returns details of an employee for specified columns and table.
GenericResultBean	**getEmployeeList(HttpServletRequest request)** Returns the details of all employees from configured default table.
GenericResultBean	**getEmployeeList(java.lang.String tableName, HttpServletRequest request)** Returns details of all employees from the specified table.

GenericResultBean	**getEmployeeList(java.lang.String tableName, java.util.List<java.lang.String> columnList, HttpServletRequest request)** Returns details of all employees from specified table for a list of columns.
java.lang.String	**getEmployeePhotographSrc(java.lang.String employeeId, HttpServletRequest request)** Returns the relative path to a photograph of the specified employee according to configuration.
java.util.List<GenericBean>	**getHeaderDetails(HttpServletRequest request)** Returns 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	**getProfileDetails(java.lang.String categoryId, HttpServletRequest request)** Returns employee details for specified category of current profile selected by the user.
java.util.List<java.lang.String>	**getTableList(HttpServletRequest request)** 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)** Returns a PublicAPITableModel based upon 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 get to 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
```

Index

U

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.