SAS® Enterprise Case Management 6.2
Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute, Inc. 2013. *SAS® Enterprise Case Management 6.2: Administrator's Guide*. Cary, NC : SAS Institute Inc.

SAS® Enterprise Case Management 6.2: Administrator's Guide

Copyright © 2014, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hardcopy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

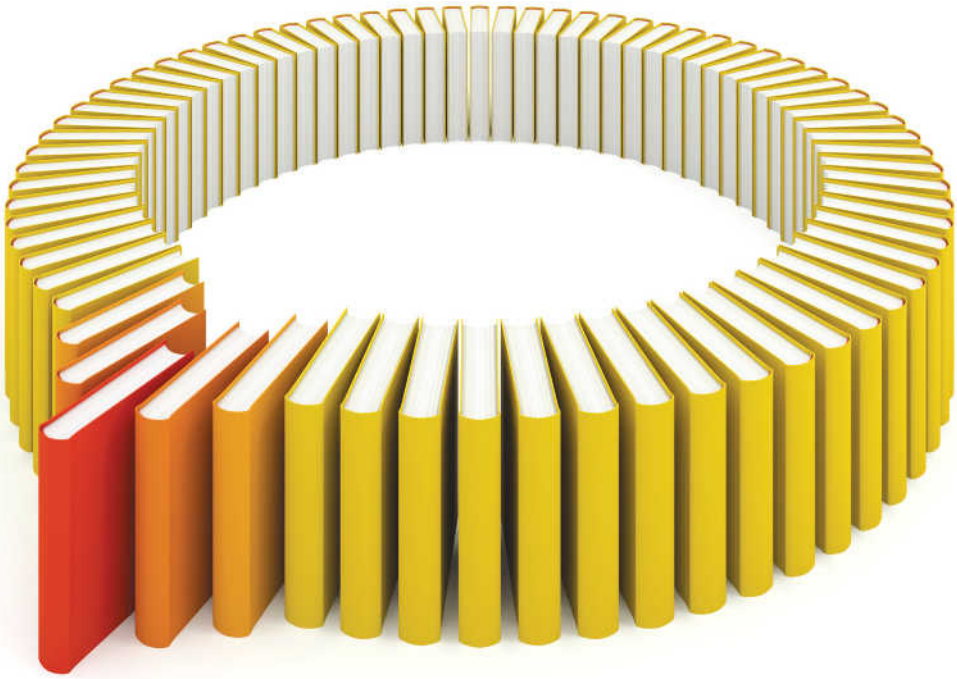
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

Electronic book 1, April 2014

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.

sas
THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

Contents

About this Book . . . . . *ix*

Chapter 1 • Introduction to SAS Enterprise Case Management 6.2 . . . . . **1**
What Is Enterprise Case Management 1
More Information 1

Chapter 2 • Pre-installation Requirements and Tasks **3**
Basic Pre-installation Steps for SAS Enterprise Case Management 4
Pre-installation: Database Information 5
Pre-installation: JDBC Drivers 8
Pre-installation: Oracle Database 9
Pre-installation: SQL Server Database 10
Pre-installation: PostgreSQL Database 11
Pre-installation: DB2 Database 12
Pre-Installation: Global Search 13
Sample Database Creation Scripts 14

Chapter 3 • Installing SAS Enterprise Case Management **15**
Selecting a Single-Tier or Multi-Tier Installation 15
SAS Deployment Wizard Tasks 16
Installed SAS Products 16
Disabling Anonymous Web Access 17
Specifying DBMS Credentials 17
Reviewing the Instructions.html File 28
Default File Locations 28
Updating the SAS SID File 30
Apply All SAS Hotfix Updates 30

Chapter 4 • Post-installation Requirements and Tasks **31**
Post-installation Overview 32
SAS Social Network Analysis Configuration 32
Specify the Isolation Level for DB2 Databases 33
Sample Scripts Checklist 33
Clustering Support 34
Defining Users, Groups, and Roles 34
Customizing Your SAS Enterprise Case Management Installation 39
Uploading Definitions and Properties 40
Capabilities in SAS Enterprise Case Management 43
Subscriptions and Notifications 51
Notifications and the SAS Information Delivery Portal 55
Configuring the Web Service 56
Deploy SAS Spelling Correction 56

Chapter 5 • Customizing SAS Enterprise Case Management **61**
Introduction to Customizing 62
User-Defined Fields 66
User-Defined Generic Data Tables 69
User Interface Definitions 70
Configurations 71

Data Security 76
Resource Bundles 77
Workflows 78
Reference Tables 82
Search Panels 87
Adding Data to Relationships 93

Chapter 6 • Using the Custom Page Builder **97**
Overview of the Custom Page Builder 98
Customizable User Interfaces 99
Assigning the Custom Page Builder Permission 100
Working with User Interface Definitions 100
Valid XML Elements and Descriptions for User Interface Definitions 101
Customizable Menus 116
Working with Menu Definitions 120
Custom Page Builder: Creating Custom Help 122
Expressions and Functions 123
Lazy Initialization for Data Displayed on a Tab 123
Customization Examples 125
DataStores and DataGrids 135
Custom Page Builder Components 148
Custom Page Builder Directives 152

Chapter 7 • Using and Configuring Global Search **157**
Introduction 157
Indexing Data 157
Global Search Configuration XML Format 158
Valid XML Elements and Descriptions for Global Search Configuration 160
Global Search Configuration Administration 165
Managing Multiple Global Search Configurations 165
Global Search Directive 166
Bulk Load Capability 166
Global Search Re-Indexing 166
ECM.Search.URL Parameter 167
Enabling or Disabling Global Search 167
Manually Installing Global Search 167

Chapter 8 • Regulatory Reports and E-Filing **169**
Introduction 170
FINCEN SAR Implementation 170
FINCEN CTR Implementation 171
E-Filing Process 172
Configuring E-Filing 173
Regulatory Report Form Configuration 177
Support of Multiple Versions of Report Forms 183
FINCEN 183
SAR 186
CTR 187

Chapter 9 • Related Items **189**
Overview 189
Configuring Match Criteria 189

Chapter 10 • Financial Items **193**
Overview 193
Defining Financial Item Types in a Reference Table 193

Defining the UDF for Financial Transactions 194
Defining the UDF for Financial Summaries 194
Adding a FinancialItemsTable Component to a Case, Incident, or Subject User Interface 194
Defining the Financial Items User Interface 194
Customizing a SAS Stored Process to Compute Financial Summaries 195

Chapter 11 • Case Network Analysis **197**
Overview 197
Case Network Analysis Process 197
Configuring Link Criteria 198
Configuring the Data Source 200
Configuring Displayed Data Fields and Link Filters 200
Configuring Display Labels 201
Case Network Analysis Logic 202

Chapter 12 • Configuring Subject Search **203**
Overview 203
Subject Search Process 203
Configuring Match Criteria for Subject Search 203
Subject Search Logic 204

Chapter 13 • Report Mart **207**
SAS Enterprise Case Management Report Mart 207

Chapter 14 • Internationalization **211**
Overview 211
Specify the Database Character Encoding 211
SAS Session Encoding Consideration and DBCS Support 212
Default Encoding for Databases Supported by SAS Enterprise Case Management . . 213
Restricting the Maximum Length of VARCHAR Fields 213
Naming Conventions for Locales 214
Create and Use Custom Translated Messages 214
Localizing Custom Table Labels and Column Labels 215
Localizing Reference Tables 216
Localizing Workflow Activities and Statuses 216

Chapter 15 • Adding Custom SAS Code **217**
Adding Custom SAS Code 217

Chapter 16 • Event Logging **219**
Overview 219
Currently Supported Events 219
Creating a Batch Load Event 222

Chapter 17 • Additional Tasks **225**
Case Routing Configurations for SAS Enterprise Case Management: Regional Manager Setup 225
Setting Up Data Management Jobs 230
SAS Enterprise Case Management – Backup Requirements 230
Metadata Clustering 231

Appendix 1 • SAS Enterprise Case Management Web Service **233**
Introduction 233
A Sample Request 234

Appendix 2 • Troubleshooting 235
Workflow Status Updates 236
Report Workflow Transition Problems 236
Unable to Open SAS Workflow Studio 236
Database Error Warnings and SAS Deployment Wizard 237
Post-installation Database Steps Required after Unsuccessful SAS Deployment Wizard Database Installation 237
ODBC Database Transcode Error 238
Case Network Graph Stops Working 238
Case Network Analysis Web Service Not Created 239
Special Characters Are Missing in Case Network Analysis and Report Mart 240
Specifying the Version Number for SAS Enterprise Case Management 241
DBMS JAR File and Multiple Machine Installations 241
Assigning the Primary Owner to a Case 241
Adding the Custom Column Type VARCHAR 241
Locking and Unlocking an Entity 242
Using the Search Functionality in SAS Enterprise Case Management 242
Search Box Error Messages 243
SAS Enterprise Case Management Time-Out 244
Returning to SAS Workflow Administrator after Timing Out 244
Financial Items Warning Message 245
Transaction Sequential Number Is Not Assigned After an E-File Is Generated 245
Incorrect or Missing Translations 245
Field Definition Changes and Search Page Errors 246
Warning While Uploading a UI Definition with Required Fields 246
SAS Lock Down State Not Supported 246
Uploading Large Attachments 247
Main Navigation Menu Errors When Logging in 247
Application Is Not Finding Data in the Database 247
Notifications are Not Being Received 248
Maintaining Database Statistics 248
Browser Not Supported Message in Internet Explorer 9 249
Enabling Right-to-Left Page Layout 249

Appendix 3 • SASMSG and %SMD2DS 251
How Does SAS Enterprise Case Management Use %SMD2DS and SASMSG? 251
About the SASMSG Function 251
Formatting 252
Open Code Macro Statements 252
The %SMD2DS Macro 252
Example: Add a Message and Its Translation to Be Used by SASMSG 253
Example: Message Substitutions 254

Index 255

About this Book

Audience

This documentation is intended primarily for those users who are responsible for the installation and configuration of SAS Enterprise Case Management. A secondary audience includes those users who are responsible for managing data, creating workflows and user interfaces, and overseeing case management. Examples of such users include systems administrators, database administrators, and high-level case management personnel who are interested in implementing a specific configuration of SAS Enterprise Case Management in order to meet specific organizational case management goals. Therefore, the scope of this documentation is limited primarily to the administrative tasks that these users are likely to perform. Moreover, this documentation assumes familiarity with the technical terminology and concepts that are required to perform these tasks. For information about the functionality of the SAS Enterprise Case Management user interface, see the *SAS Enterprise Case Management: User's Guide*.

Chapter 1

Introduction to SAS Enterprise Case Management 6.2

What Is Enterprise Case Management 1

More Information 1

What Is Enterprise Case Management

Case management is a business process that involves coordinating, researching, and tracking information about incidents that might pose a risk to an organization. Case management can span organizations and include various business users.

Because financial and banking institutions are required to report suspicious financial activity, case management includes the process of electronically filing regulatory reports with government agencies.

By providing a structured environment for defining and managing workflows, SAS Enterprise Case Management enables business users to streamline processes and conduct more efficient, effective, and consistent investigations. Customized workflows can be created for various types of cases and reports. Workflows are classified by type, category, and subcategory, and automatically route cases or reports to the appropriate individuals or groups, as defined by your organization. Workflows can require users to complete specific actions before moving a case or report to the next step in the business process.

SAS Enterprise Case Management creates auditable records for management, examiners, and regulatory agencies. Each audit record contains user identification, a time stamp, and the dates when actions were performed.

More Information

For information about support fixes, see the SAS Notes that are available on the SAS Technical Support website. Search for available SAS Notes for SAS Enterprise Case Management at **`http://support.sas.com`**.

For information about the hardware, software, and database requirements of SAS Enterprise Case Management, and for links to other sources of related information, see **`http://support.sas.com/resources/sysreq/index.html`**.

Chapter 2

Pre-installation Requirements and Tasks

Basic Pre-installation Steps for SAS Enterprise Case Management 4
Completing Pre-installation Tasks 4
Verifying Your Operating System Requirements 4
Creating the SAS Enterprise Case Management User Accounts 4
Obtaining a Deployment Plan and SID File 5
Download Your Software with the SAS Download Manager 5

Pre-installation: Database Information 5
Determining the Required Database Information 5
SAS Web Infrastructure Platform Database Requirement 7
Creating the Social Network Analysis Database 8

Pre-installation: JDBC Drivers 8

Pre-installation: Oracle Database 9
Installing the Oracle Database 9
Create the Oracle User for Enterprise Case Management 9
Test Access to the Database 9

Pre-installation: SQL Server Database 10
Installing the SQL Server Database 10
Create the SQL Server User for Enterprise Case Management 10
Verify SAS/ACCESS for ODBC or SAS/ACCESS for SQL Server 10
Configuring the SQL Server ODBC Connection 10
Test Access to the Database 11

Pre-installation: PostgreSQL Database 11
Installing the PostgreSQL Database 11
Configuring the PostgreSQL Database for a Multi-Tier Installation 11
Creating the PostgreSQL User for SAS Enterprise Case Management 11
Test Access to the Database 12

Pre-installation: DB2 Database 12
Installing the DB2 Database 12
Create the DB2 User for Enterprise Case Management 12
Test Access to the Database 12

Pre-Installation: Global Search 13

Sample Database Creation Scripts 14

Basic Pre-installation Steps for SAS Enterprise Case Management

Completing Pre-installation Tasks

Before you begin to install the SAS Intelligence Platform and SAS Enterprise Case Management, you must complete a set of pre-installation tasks. You must install various third-party components, verify your operating system requirements, create the needed user accounts, address database requirements, and obtain your SAS software. Specifically, you must complete the following tasks:

Verifying Your Operating System Requirements

Before you install SAS Enterprise Case Management, make sure that you meet the minimum system requirements that are described in the system requirements documentation. System requirements are unique for each operating system. Items that are addressed as system requirements include software requirements, hardware requirements, space requirements, specific product requirements, and graphics hardware and software compatibility.

Some specific items that you should check include the following settings:

- Set the screen resolution for SAS Enterprise Case Management no lower than 1024 x 768.
- Set your browser's pop-up blocker to allow pop-ups for your applications.

For more requirements information, see “SAS System Requirements” at **`http://support.sas.com/resources/sysreq/index.html`**.

Creating the SAS Enterprise Case Management User Accounts

As a pre-installation task, you must have two operating system accounts, one for installing and configuring the SAS software and one for running the spawned servers. For details about these user accounts, refer to the pre-installation checklist or the *SAS Intelligence Platform: Installation and Configuration Guide* for your deployment. You will also need at least one operating system account to serve as an administrator for SAS Enterprise Case Management. Details about preparing a user account are discussed in Chapter 4, “Post-installation Requirements and Tasks,” on page 31. You can use an existing account if one already exists. A SAS Enterprise Case Management administrative user is specific to SAS Enterprise Case Management. This user must have a valid host operating system account, and as a post-installation task, you must associate that account with a metadata user in SAS Management Console. Product administrators have access to perform any action on any data in SAS Enterprise Case Management.

The SAS Spawned Servers account (sassrv) needs to be separate, but from the same user group, as the installation user. In a Windows environment, it must be included in the administrator's group to ensure stored processes can write to the SAS Enterprise Case Management config directories. Refer to *SAS Intelligence Platform: Installation and Configuration Guide* for guidance in setting up the SAS Spawned Servers account (sassrv).

It is often necessary to change the name of the administrative user from admin to match an existing user name in your environment. For example, if you configure your web application server so that the SAS Enterprise Case Management web application authenticates users against an LDAP server, then you must change the name of the administrative user to the user name found in the LDAP user directory. That user can then log on as the administrator in SAS Enterprise Case Management. Be aware that a SAS Enterprise Case Management product administrator account is not the same as a general administrator account, such as the SAS Administrator (sasadm@saspw).

See "Setting Up Users, Groups, and Ports" in the *SAS Intelligence Platform: Installation and Configuration Guide*.

Note: SAS Enterprise Case Management uses both regular user accounts and a product administrative user account. You can create regular user accounts for SAS Enterprise Case Management as a post-installation task. For more information, see **`http://support.sas.com/documentation/`**.

Obtaining a Deployment Plan and SID File

Before you can install your SAS software, you must obtain a deployment plan and SID file. The deployment plan is a summary of the software that is installed and configured during your installation. A deployment plan file, named plan.xml, contains information about what software should be installed and configured on each machine in your environment. This plan serves as input to the SAS installation and configuration tools. A deployment plan can be a custom plan for your specific software installation or it can be a standard, predefined deployment plan that describes a common configuration. The SID file is used by the SAS system to install and license SAS software. It is a control file that contains license information that is required in order to install SAS. For more information about deployment plans and the SID file, see "SAS Deployment Wizard Options" and "About Deployment Plans" in the *SAS Intelligence Platform: Installation and Configuration Guide*.

Download Your Software with the SAS Download Manager

Download the software that is listed in your SAS Software Order with the SAS Download Manager. You can then use the SAS Deployment Wizard to install your software.

Pre-installation: Database Information

Determining the Required Database Information

During the installation and configuration of SAS Enterprise Case Management, the SAS Deployment Wizard requires information about the database that SAS Enterprise Case Management uses. The following table provides information that you must have to complete the steps in the SAS Deployment Wizard.

Table 2.1 *Database Information*

Property	Description
Database Type	Specifies the database vendor to use with SAS Enterprise Case Management. SAS Enterprise Case Management supports the Oracle, SQL Server, and DB2, and PostgreSQL databases.
User Name or Schema	Specifies the user name for the database used with your SAS Enterprise Case Management installation. *Note:* The schema user requires adequate permissions to create all objects required for the schema initialization. For Oracle, these include sequences, tables, indexes and views. *Note:* The configuration script assumes that the user name and schema name are the same. However, you can change the schema name that SAS Enterprise Case Management uses after the configuration has completed. See "Application Is Not Finding Data in the Database" on page 247 for more information.
Password	Specifies a valid password for the user name associated with the database account.
Port	Specifies the port used by the database. The default ports for the databases supported by SAS Enterprise Case Management are as follows: • DB2:50000 • Oracle:1521 • Microsoft SQL Server:1433 • PostgreSQL:5432
Host Name	Specifies the host name of the machine where the database is installed.

Property	Description

Database Name	Specifies the database name. For SQL Server and PostgreSQL, there must be an ODBC connection with the same name as the database name.
	
	For Oracle databases, the Net Service Name and the Service Name fields that are configured in the tnsnames.ora file must be the same. You must use this value for the **Database Name** field in the SAS Deployment Wizard.
	
	For example, if you had the following entry in the tnsnames.ora file, you would enter **`datahaus`** in the **Database Name** field in the SAS Deployment Wizard:
	
	`datahaus =`
	`(DESCRIPTION =`
	`(ADDRESS_LIST =`
	`(ADDRESS =`
	`(COMMUNITY = TCP_COMM)`
	`(PROTOCOL = TCP)`
	`(HOST = hostname.your.company.com)`
	`(PORT = 1521)`
	`)`
	`)`
	`(CONNECT_DATA =`
	`(SERVICE_NAME = datahaus)`
	`)`
	
	Note: The Net Service Name and Service Name in the above example are the same. This is required to build the correct LIBNAME statement in the SAS Enterprise Case Management stored processes.

DBMS JDBC JAR File	Specifies the location of the database vendor's JDBC JAR file to facilitate Java access. You must have this file available on the middle tier.
Path to DB2 License JAR File (DB2 only)	Specifies the location of the DB2 license JAR file.

SAS Web Infrastructure Platform Database Requirement

SAS Enterprise Case Management uses SAS Web Infrastructure Platform. SAS Web Infrastructure Platform must be installed with a platform-supported database, such as the SAS Web Infrastructure Platform Data Server or a third party database such as Oracle, SQL Server, or DB2. See the SAS Web Infrastructure Platform documentation for supported databases.

Note: When you install SAS Web Infrastructure Platform with SQL Server, the SAS Deployment Wizard creates the database during configuration. Therefore, you should create a SQL Server login with the dbcreator role, but do not create a database. When asked by SAS Deployment Wizard for a SAS Web Infrastructure Platform database name, provide the name of the database that you want to be created. It should not be the name of an existing database.

Creating the Social Network Analysis Database

SAS Enterprise Case Management uses SAS Social Network Analysis. Because SAS Social Network Analysis stores information in a database, you are prompted while configuring SAS Enterprise Case Management for information about a database connection for SAS Social Network Analysis. Before you configure SAS Enterprise Case Management, you need to create a SAS Social Network Analysis database. SAS Social Network Analysis supports the following databases:

- Oracle
- DB2
- PostgreSQL
- SQL Server

To create the database, you need to create a script. For sample scripts, see "Sample Database Creation Scripts" on page 14.

To prepare the SAS Social Network Analysis database and create the tables, see the "Preparing the Configuring the Database" chapter of the *SAS Social Network Analysis Server: Installation and Configuration Guide* found at **`http://support.sas.com/documentation/solutions/sna/index.html`**.

Pre-installation: JDBC Drivers

Note: The following JDBC drivers must be placed in a separate directory without any other files to ensure proper installation and configuration of SAS Enterprise Case Management.

- Oracle: SAS Enterprise Case Management uses the ojdbc6.jar file. You can download a copy of the Oracle driver from **`http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html`**. Select the latest Oracle driver. The JDBC driver version must match the database version.
- PostgreSQL 9.0: SAS Enterprise Case Management uses the postgresql-9.2-1002.jdbc4.jar PostgreSQL driver. It is located in the **`jdbc`** subdirectory of your PostgreSQL installation.
- SQL Server 2008: SAS Enterprise Case Management uses the sqljbc4.jar Microsoft SQL Server JDBC Driver SQL Server Native Client 10.0. Visit the official Microsoft website to download this driver.
- DB2: SAS Enterprise Case Management uses two JDBC files: db2jcc.jar and db2jcc_license_cu.jar. These files are located in the **`\sqllib\java`** directory of your DB2 installation.

Pre-installation: Oracle Database

Installing the Oracle Database

SAS Enterprise Case Management requires a database. You must install this third-party software before installing SAS Enterprise Case Management. SAS Enterprise Case Management supports the Oracle database.

Create the Oracle User for Enterprise Case Management

SAS Enterprise Case Management will store transactional data in the Oracle database. Before installing SAS Enterprise Case Management, create a user in Oracle with the following privileges:

- CREATE SESSION
- CREATE SEQUENCE
- CREATE TABLE
- CREATE VIEW

In addition, that user will need adequate table space quota for its default and temporary table spaces.

Note: The schema user requires adequate permissions to create all objects required for the schema initialization. For Oracle, these include sequences, tables, indexes and views.

Test Access to the Database

As a post-installation task, you must run several SAS scripts provided by SAS Enterprise Case Management. These database scripts assume that the SAS environment can already access the Oracle database.

To test the ability of SAS to access the Oracle database, after SAS Enterprise Case Management is installed but before it is configured, open an interactive SAS session and try to create a libref, for example:

```
libname ecmtest oracle path='casemgmt' user='ecmdata' password='ecmdata'
   connection=global;
libname tmptest oracle path='casemgmt' user='ecmdata' password='ecmdata'
   dbmstemp=yes connection=global;
```

Pre-installation: SQL Server Database

Installing the SQL Server Database

SAS Enterprise Case Management requires a database. You must install this third-party software before installing SAS Enterprise Case Management. SQL Server is one of the databases supported by SAS Enterprise Case Management.

Create the SQL Server User for Enterprise Case Management

SAS Enterprise Case Management stores transactional data in the SQL Server database. Before installing SAS Enterprise Case Management, create a user in SQL Server with access to the target database. Create a schema and make the new user the owner of the schema. Set that as the default schema for the user for that database.

Verify SAS/ACCESS for ODBC or SAS/ACCESS for SQL Server

SAS Enterprise Case Management supports the SQL Server database. As a post-installation task, you must run several SAS scripts provided by SAS Enterprise Case Management. These database scripts assume that the SAS environment can already access the SQL Server database. Follow the instructions in the section "Configuring the SQL Server ODBC Connection" on page 10, to verify that your SAS/ACCESS engine can access your SQL Server database.

Configuring the SQL Server ODBC Connection

If you are using SAS/ACCESS for ODBC in Windows, you need to create a System DSN (Data Source Name).

1. From the Windows **Start** menu, select **Settings** ⇨ **Control Panel** ⇨ **Administrative Tools** ⇨ **Data Sources (ODBC)**. The ODBC Data Source Administrator window appears.
2. Select the **System DSN** tab and then click **Add**. The Create New Data Source window appears.
3. Select the SQL Server Native Client 10.0 driver from the list and then click **Finish**. The SQL Server ODBC Driver Setup window appears.
4. Enter the driver information in the ODBC Driver Setup window. For example, complete the following:
   a. Enter `casemgmt` in the **Name** box. The DSN name must be the same as the name of the database in SQL Server.
   b. (Optional) Enter ***SAS Enterprise Case Management Transactional Schema*** in the **Description** box.
   c. Enter the database server host name in the **Server** box.
   d. Click **Next**.
   e. Enter the appropriate authentication setting.

f. Enter user name and password information to obtain the default settings. Enter a user name (for example, ecmdata) in the **User Name** box. Enter a password in the **Password** box.

g. Click **Next**.

h. Click **Finish**.

i. Click **Test Data Source** to verify the data source information.

j. Click **OK** to save the driver information and to close the SQL Server ODBC Driver Setup window.

k. Click **OK** to close the ODBC Data Source Administrator window.

Test Access to the Database

To test the ability of SAS to access the SQL Server database, after SAS Enterprise Case Management is installed, but before it is configured, open an interactive SAS session and try to create a libref. For example, using the sample information in the previous section, the LIBNAME statement would look like this:

```
libname ecmtest odbc dsn=casemgmt user=ecmdata password=ecmdata connection=global;
libname tmptest odbc dsn=casemgmt user=ecmdata password=ecmdata connection=global
   dbmstemp=yes;
```

Pre-installation: PostgreSQL Database

Installing the PostgreSQL Database

SAS Enterprise Case Management requires a database. You must install this third-party software before installing SAS Enterprise Case Management. PostgreSQL is one of the databases supported by SAS Enterprise Case Management.

Configuring the PostgreSQL Database for a Multi-Tier Installation

For security reasons, PostgreSQL does not listen on all available IP addresses on the server machine initially. In order to access the server over the network, you must enable listening on the address first.

For PostgreSQL servers version 8.0 and later, this is controlled using the listen_address parameter in the postgresql.conf file. Here, you can enter a list of IP addresses the server should listen on, or simply use * to listen on all available IP addresses.

Creating the PostgreSQL User for SAS Enterprise Case Management

SAS Enterprise Case Management will store transactional data in the PostgreSQL database. Before installing SAS Enterprise Case Management, create a user in PostgreSQL and then create a database owned by that user.

Note: The SAS Enterprise Case Management configuration script assumes that the user's default schema is the same as the user name. In PostgreSQL, the default

schema is typically “public.” After creating the SAS Enterprise Case Management user, log on as that user, create a new schema, and then set the account search_path to that schema. For example:

```
psql -U postgres -c "CREATE USER ecmdbuser WITH PASSWORD foo CREATEDB
NOCREATEUSER"
createdb -U ecmdbuser -E UNICODE ecmdb
psql -U ecmdbuser -c "CREATE SCHEMA ecmdbuser AUTHORIZATION foo"
psql -U ecmdbuser -c "SET search_path TO ecmdbuser"
```

Test Access to the Database

To test the ability of SAS to access the PostgreSQL database, after SAS Enterprise Case Management is installed, but before it is configured, open an interactive SAS session and try to create a libref. For example, using the sample information in the previous section, the LIBNAME statement would look like this:

```
libname ecmtest POSTGRES server=’mydbserver.com’ port=5432 database=casemgmt
user=ecmdata password=ecmdata connection=global;

libname tmptest POSTGRES server=’mydbserver.com’ port=5432 database=casemgmt
user=ecmdata password=ecmdata connection=global dbmstemp=yes;
```

Pre-installation: DB2 Database

Installing the DB2 Database

SAS Enterprise Case Management requires a database. You must install this third-party software before installing SAS Enterprise Case Management. DB2 is supported by SAS Enterprise Case Management.

Create the DB2 User for Enterprise Case Management

SAS Enterprise Case Management stores transactional data in the DB2 database. Before installing SAS Enterprise Case Management, create a user in DB2 with access to the target database. Create a schema and make the new user the owner of the schema. Set that as the default schema for the user, for that database.

Test Access to the Database

SAS Enterprise Case Management supports the DB2 database. As a post-installation task, you must run several SAS scripts provided by SAS Enterprise Case Management. These database scripts assume that the SAS environment can already access the DB2 database.

To test the ability of SAS to access the DB2 database, after SAS Enterprise Case Management is installed, but before it is configured, open an interactive SAS session and try to create a libref. For example, using the sample information in the previous section, the LIBNAME statement would look like this:

```
libname ecmtest db2 database=casemgmt user=ecmdata password=ecmdata
   connection=global;
```

```
libname tmptest db2 database=casemgmt user=ecmdata password=ecmdata
   connection=global dbmstemp=yes;
```

Pre-Installation: Global Search

SAS Enterprise Case Management 6.2 integrates with the Solr open source search engine to provide text-based searching of any data visible in SAS Enterprise Case Management. Solr is a Java-based program that maintains a search index and responds to search queries using sockets. Like a database, Solr stores the search index on the local file system. The example Solr web application runs within the Jetty web application container on port 8983. Solr can be configured to run in other web application containers and on other ports. However, these instructions assume the use of the provided Jetty container and port.

Note: Depending on the size of the data you are indexing, you might choose to deploy Solr on one of your SAS middle-tier servers or on a separate search server.

SAS Enterprise Case Management includes a script to simplify the configuration of a Solr server. The script and other related configuration files are bundled into a single ZIP file named PrepareSolr.zip. If you choose to install Solr on a separate host machine, you will need to copy the PrepareSolr.zip file from your middle-tier machine to your destination machine.

1. Download solr-4.2.1.zip from **http://archive.apache.org/dist/lucene/solr/4.2.1**.
2. Unzip solr-4.2.1.zip in its destination directory. For Windows, a common path might be **C:\solr-4.2.1**. For UNIX, a common path might be **/usr/local/solr-4.2.1**.
3. Open ***SAS-middle-tier-installation-directory*/deploy/sample/Solr/PrepareSolr.zip** and unzip it into the solr-4.2.1 directory. This creates a folder under **solr-4.2.1** named **ecm**.

To run the PrepareSolr script on Windows, issue the following commands:

1. **cd \solr-4.2.1\ecm**
2. **PrepareSolr.cmd ..**

To run the PrepareSolr script on UNIX, issue the following commands:

1. **cd /usr/local/solr-4.2.1/ecm**
2. **chmod +x prepareSolr.sh**
3. **./prepareSolr.sh ..**

This creates a new Solr server directory named **ecm** under your Solr installation, with all required files copied into place. You can start the new Solr server by issuing the following commands:

For Windows:

1. **cd \solr-4.2.1\ecm**
2. **start-solr.cmd**

For UNIX:

1. **cd /usr/local/solr-4.2.1/ecm**
2. **./start-solr.sh**

When the server starts up, you will see log file information. If it starts up correctly, you should see a message in the log file that looks like the following:

```
4914 [main] INFO org.eclipse.jetty.server.AbstractConnector –
Started SocketConnector@0.0.0.0:8983
```

Note: You can gracefully stop Solr by pressing Ctrl-C in the terminal window and waiting for the process to terminate.

The SAS Enterprise Case Management configuration script will ask if you want to use the new search functionality. If you choose yes, then it will prompt for the Solr URL. If you are running Solr using the PrepareSolr script, the URL will be **http://<*hostname*>:8983/solr/EntCaseMgmtCollection**.

Instructions for setting up Solr as an operating system service are provided in the PrepareSolr directory.

You can verify that your Solr server has started up properly by opening **http://hostname:8983/solr** in your browser. This should open the Solr dashboard. On the left side is a drop-down menu with two cores, EntCaseMgmtCollection and SearchMetadata.

Note: If you attempt to open the Solr URL you need for configuration (**http://hostname:8983/solr/EntCaseMgmtCollection**), it is normal for that link to return a 404 error.

Sample Database Creation Scripts

Sample scripts are provided for creating users, schemas, and databases. The scripts are in one of the following locations, depending on the platform:

- Windows platforms: ***SAS-installation-directory*\SASFoundation\9.4\casemgmtmva\sasmisc\sample\dbscript\<*dbname*>**
- UNIX platforms: ***SAS-installation-directory*/SASFoundation/9.4/misc/casemgmtmva/sample/dbscript/<*dbname*>**

Documentation for running the scripts is provided within the script files in the form of comments.

Chapter 3

Installing SAS Enterprise Case Management

Selecting a Single-Tier or Multi-Tier Installation 15
SAS Deployment Wizard Tasks 16
Installed SAS Products 16
Disabling Anonymous Web Access 17
Specifying DBMS Credentials 17
Reviewing the Instructions.html File 28
Default File Locations 28
Updating the SAS SID File 30
Apply All SAS Hotfix Updates 30

Selecting a Single-Tier or Multi-Tier Installation

You can install SAS Enterprise Case Management on one or several machines. This choice is determined at the time you order SAS Enterprise Case Management and is detailed in the order plan (plan.xml) file. You must first install SAS Enterprise Case Management on the server-tier machine. You can then install SAS Enterprise Case Management on other additional machines that are part of a middle tier in your configuration. For guidelines on installing SAS on multiple machines, see “Installation Order Rules for Multiple Machine Deployments” in the *SAS Intelligence Platform: Installation and Configuration Guide*.

The server tier consists of a set of SAS servers that are installed as a part of the SAS Intelligence Platform. These servers host (and can be used to load) the reporting data. In addition, they execute SAS analytical and reporting processes. The SAS Workspace Server, SAS Stored Process Server, and SAS Metadata Server enable this capability.

The middle tier hosts the web application, which is deployed on a Java web application server. The web application sends data to and receives data from the web browsers on the client tier. It then organizes the data for storage on the data tier and for use on the server tier.

The client tier is also part of the SAS Enterprise Case Management configuration. On the client tier, users collect and load data and perform day-to-day operational risk tasks through the web application. In addition, although reports are configured on the server tier, they are visible in the user interface to users who have access only to the machines on the client tier.

SAS Deployment Wizard Tasks

The SAS Deployment Wizard is used to install and configure the SAS software and related products that are included in your deployment plan file. When you execute the SAS Deployment Wizard, you select the deployment type that you are performing. You can choose to install and configure the software in the same instance, or you can configure the software at a later point. The latter is recommended because it gives you the opportunity to test the SAS license and the database connection in SAS before the configuration step.

Depending on your specific deployment plan and the SAS products that you are installing, the SAS Deployment Wizard can prompt you to perform a variety of tasks, including the following items:

- specify your order plan and SAS software products that you are installing and configuring
- specify third-party products that you have installed
- specify any required machine information
- specify server information for any SAS servers that you are installing
- specify user account information
- specify database connection information
- specify whether to install global search
- install sample UI definitions, custom properties, menu definitions, workflows, and search configurations
- install the server tier for SAS Enterprise Case Management on the server machine in your configuration
- install the middle tier for SAS Enterprise Case Management on other machines in your configuration

For further information, see "Preparing to Install and to Configure" in the *SAS Intelligence Platform: Installation and Configuration Guide*. In addition, see the *SAS Deployment Wizard User's Guide* at **`http://support.sas.com/documentation/installcenter/`**.

Installed SAS Products

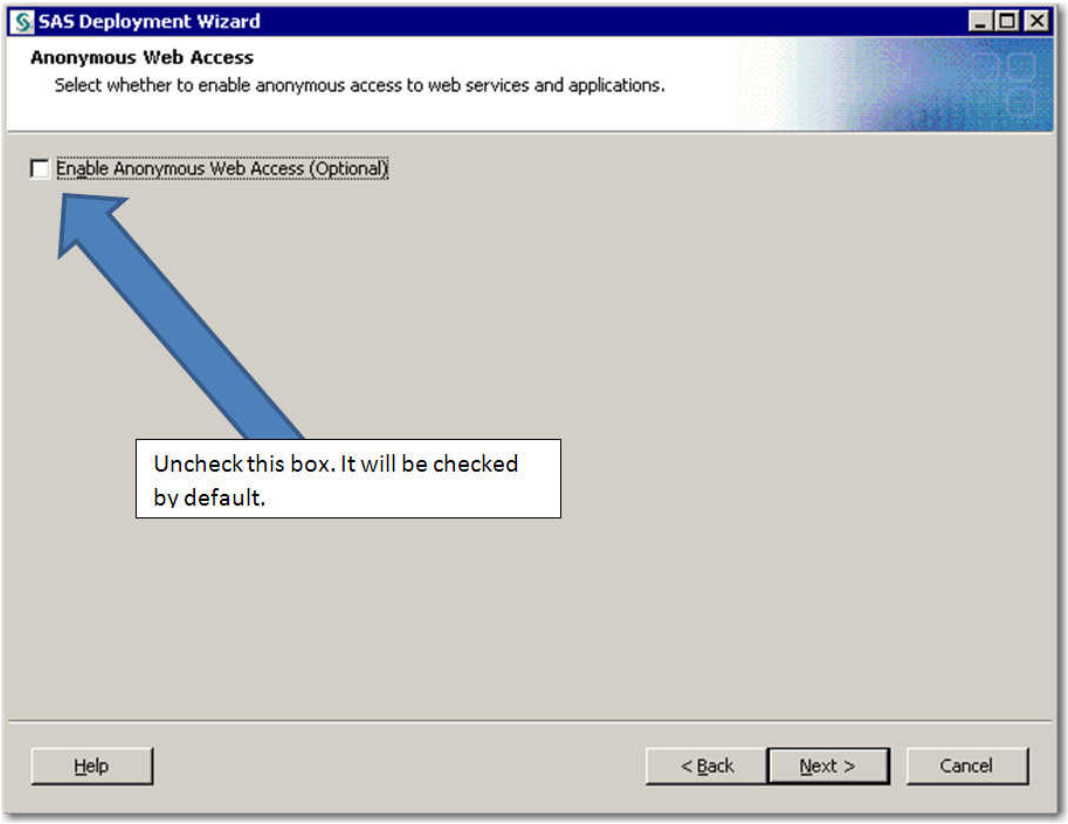
SAS Enterprise Case Management installation includes the installation of various SAS products. During installation, the SAS Deployment Wizard prompts you for the installation and possibly the configuration of each of these SAS products. Some of the products that are installed as part of the SAS Enterprise Case Management installation include the following:

- SAS Foundation 9.4
- SAS Management Console
- SAS Web Infrastructure Platform
- SAS Web Infrastructure Data Server

Disabling Anonymous Web Access

During the installation of SAS components, there is an option to enable anonymous web access. Enabling anonymous web access might give unknown visitors access to information that they would not otherwise be able to see. It is suggested that you do not enable anonymous web access.

Display 3.1 *Disable Anonymous Web Access*

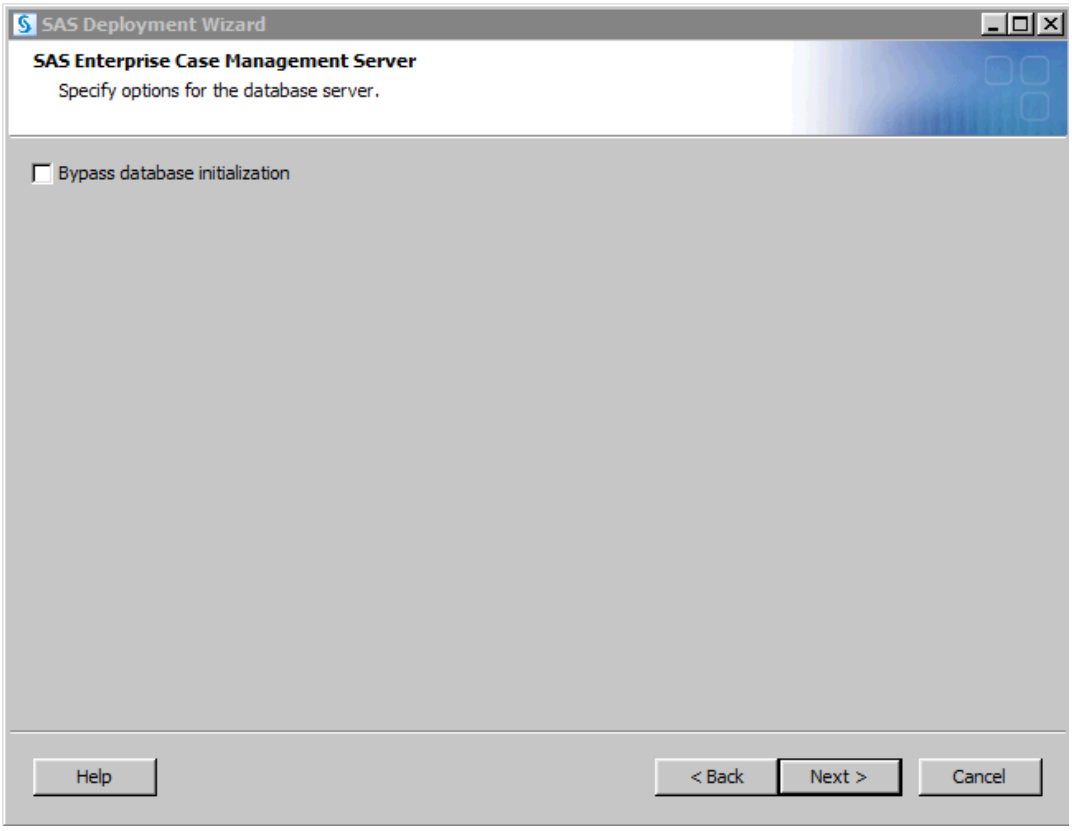


Specifying DBMS Credentials

During your SAS Enterprise Case Management installation, you must enter database management system (DBMS) information for the server tier and middle tier of your configuration. You must enter specific information for the database that is used with SAS Enterprise Case Management.

The following page prompts you to specify options for the database server.

Display 3.2 *SAS Enterprise Case Management Server-Tier Configuration — Database Server Options*

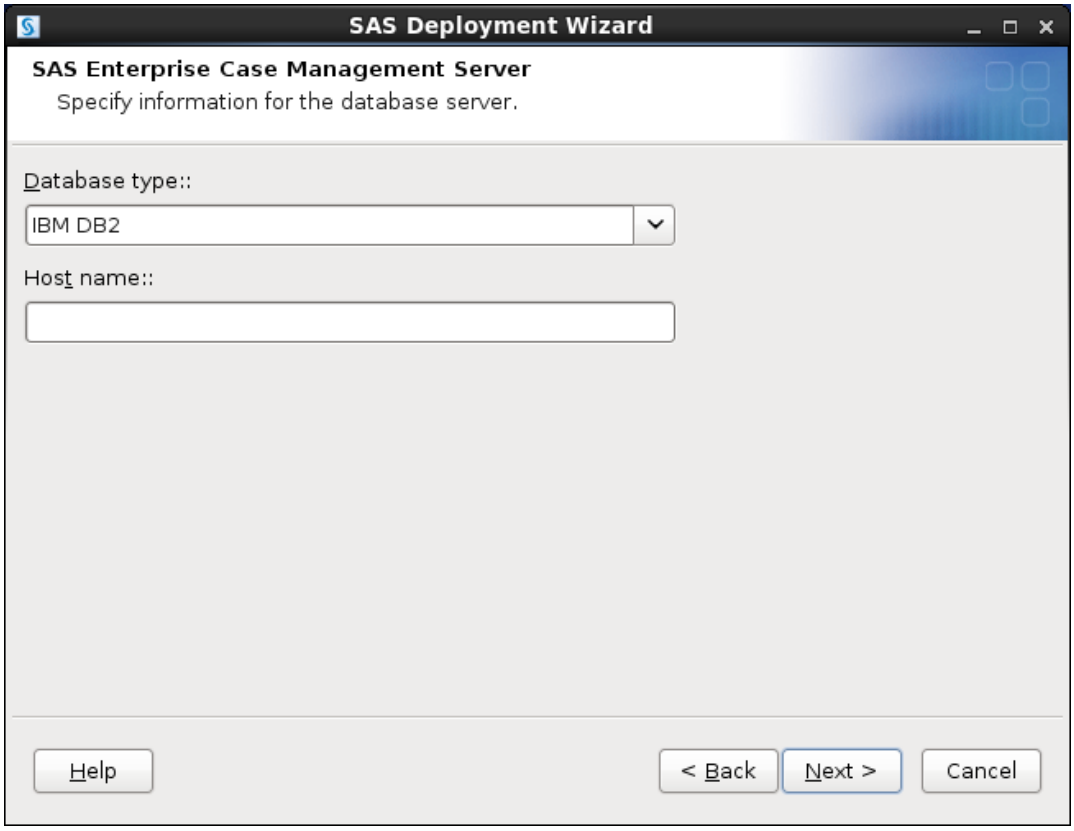


Click the following check box, depending on your need:

Bypass database initialization
specifies whether or not to bypass the database table creation and initialization during SAS Deployment Wizard configuration.

The following page prompts you to provide information for the database server.

Display 3.3 *SAS Enterprise Case Management Server-Tier Configuration — Database Server Information*



You must enter information for the following text boxes:

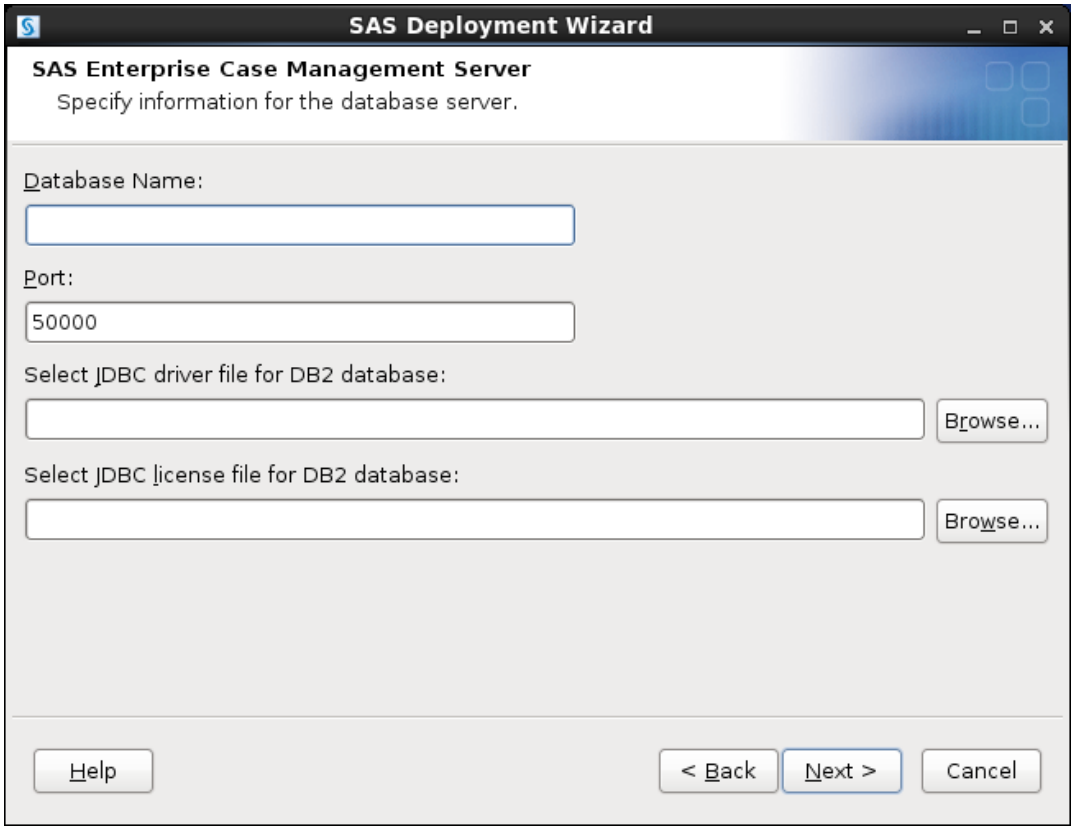
Database type
specifies the database vendor to use with SAS Enterprise Case Management.

Host name
specifies the fully qualified host name of the SAS Enterprise Case Management database.

The following page prompts you for the connection information for the database server you have chosen.

Note: You will only see the page prompt for your chosen database.

Display 3.4 *SAS Enterprise Case Management Server-Tier Configuration – DB2 Database Connection Information*



You must enter information for the following text boxes:

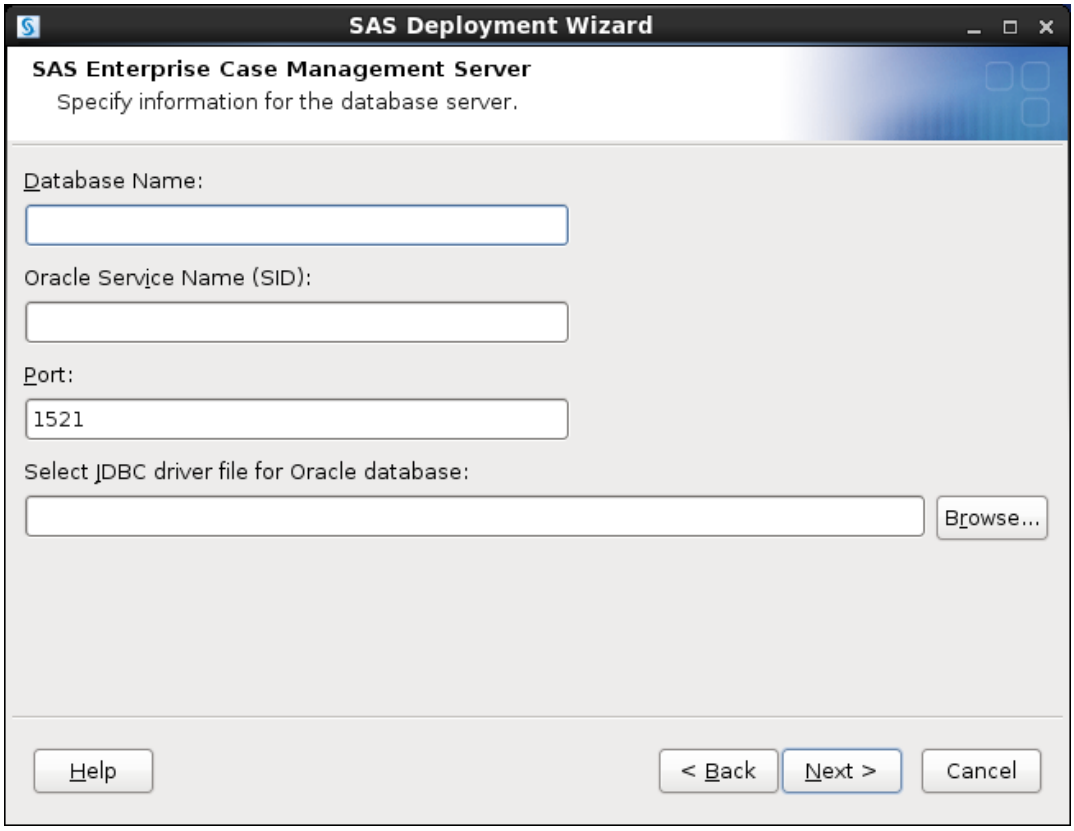
Database Name
specifies the database name.

Port
specifies the port used by the database. The default port for DB2 is 50000.

Select JDBC driver file for DB2 database
specifies the DB2-provided JDBC JAR file that facilitates Java access (for example db2jcc.jar). It just be available on this host for configuration to take place.

Select JDBC license file for DB2 database
specifies the DB2-provided JDBC JAR file for licensed Java access (for example, db2jcc_license_cu.jar). It must be available on this host for configuration to take place.

Display 3.5 *SAS Enterprise Case Management Server-Tier Configuration – Oracle Database Connection Information*



You must enter information for the following text boxes:

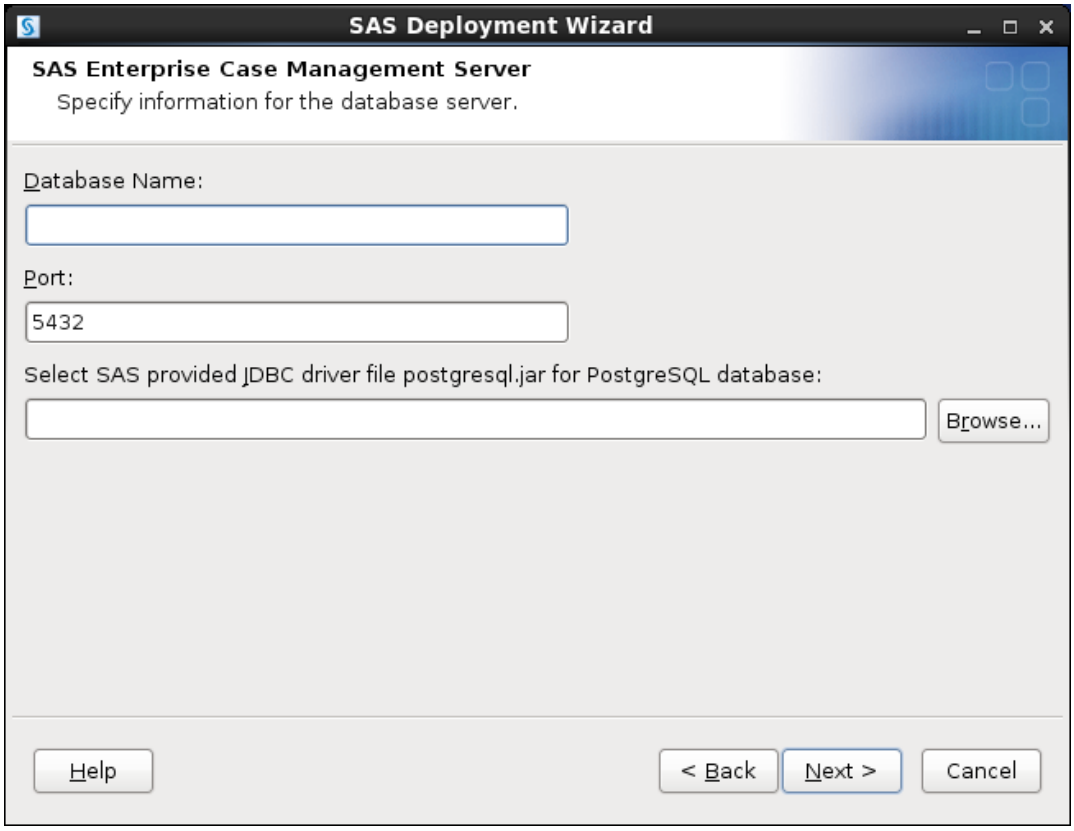
Database Name
specifies the local name of the database from tnsnames.ora.

Oracle Service Name (SID)
specifies the database name on the server.

Port
specifies the port used by the database. The default port for Oracle is 1521.

Select JDBC driver file for Oracle database
specifies the Oracle-provided JDBC JAR file that facilitates Java access. It just be available on this host for configuration to take place.

Display 3.6 *SAS Enterprise Case Management Server-Tier Configuration – PostgreSQL Database Connection Information*



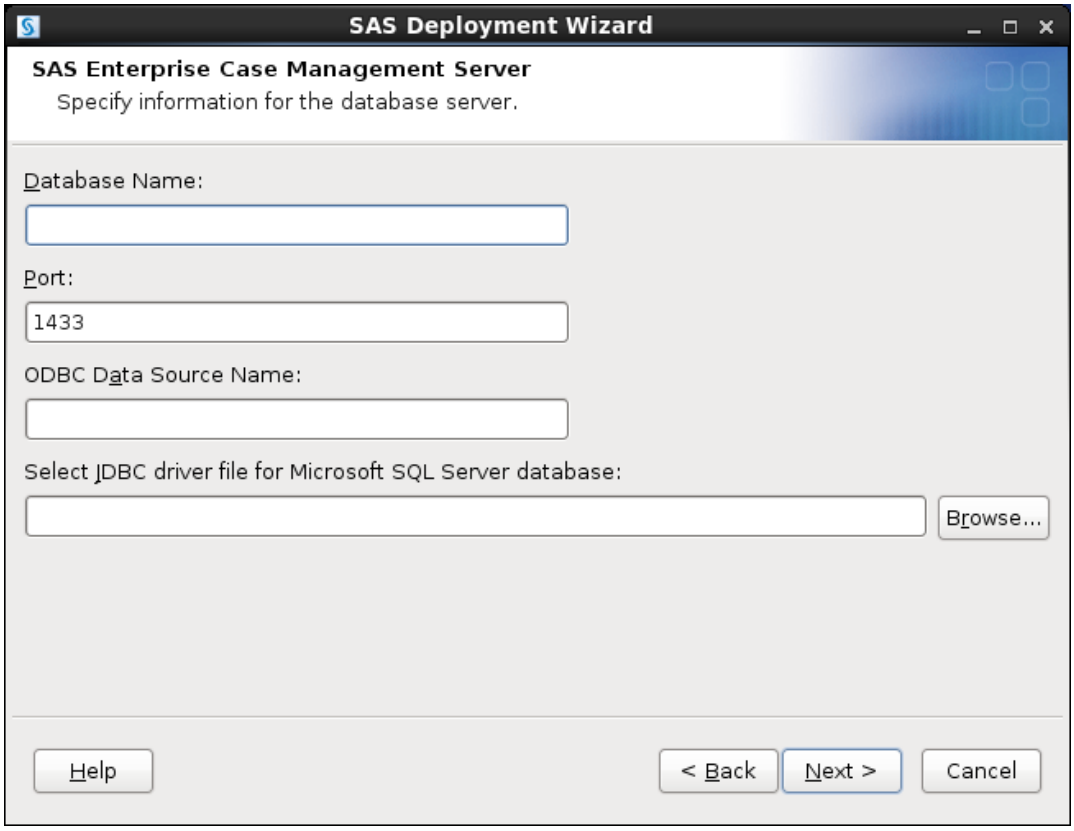
You must enter information for the following text boxes:

Database Name
specifies the database name.

Port
specifies the port used by the database. The default port for PostgreSQL is 5432.

Select SAS provided JDBC driver file postgresql.jar for PostgreSQL database
specifies the SAS-provided JDBC JAR file that facilitates Java access. It just be available on this host for configuration to take place.

Display 3.7 *SAS Enterprise Case Management Server-Tier Configuration – SQL Server Database Connection Information*



You must enter information for the following text boxes:

Database Name
specifies the database name.

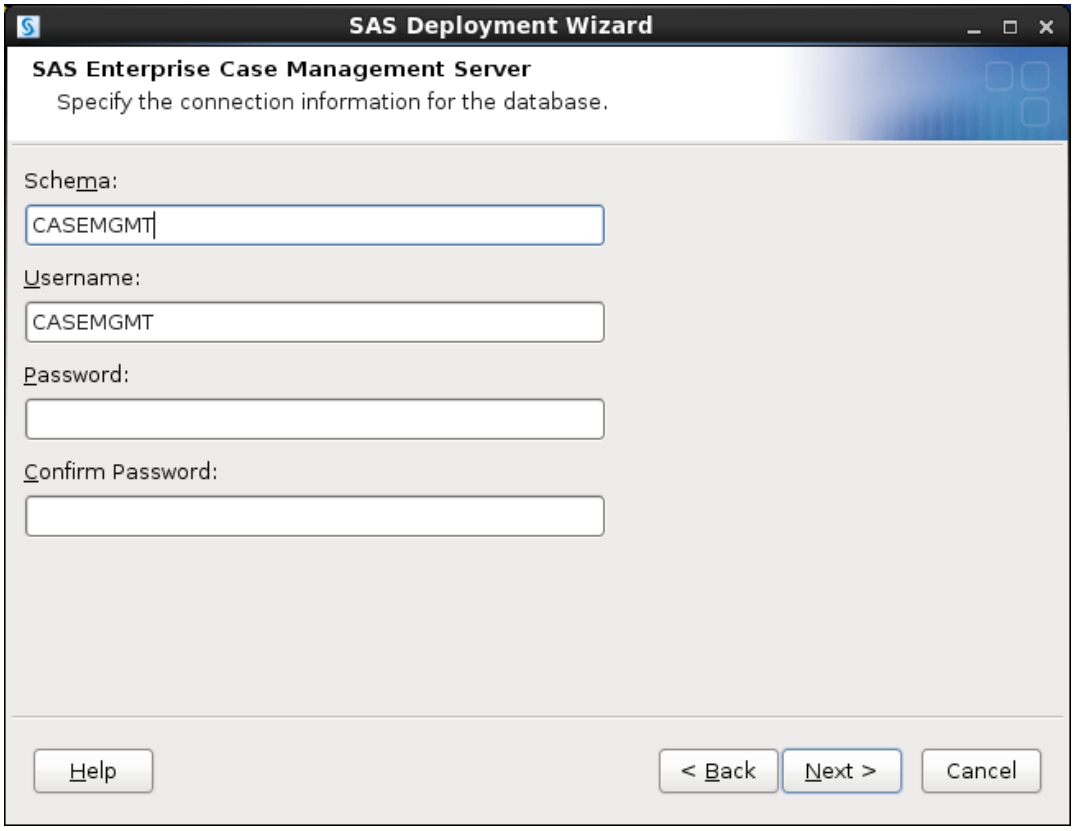
Port
specifies the port used by the database. The default port for SQL Server is 1433.

ODBC Data Source Name
specifies the name of the system DSN for the SAS Enterprise Case Management database defined in your Windows ODBC data source administrator or odbc.ini file.

Select the JDBC driver file Microsoft SQL Server database
specifies the Microsoft-provided JDBC JAR file that facilitates Java access. It just be available on this host for configuration to take place.

The following page prompts you to provide database connection information for the server machine.

Display 3.8 *SAS Enterprise Case Management Server-Tier Configuration – Server Machine Connection Information*



You must enter information for the following text boxes:

Schema
specifies the schema name that will be associated with the user for the database used with your SAS Enterprise Case Management installation.

Username
specifies the user name for the database used with your SAS Enterprise Case Management installation.

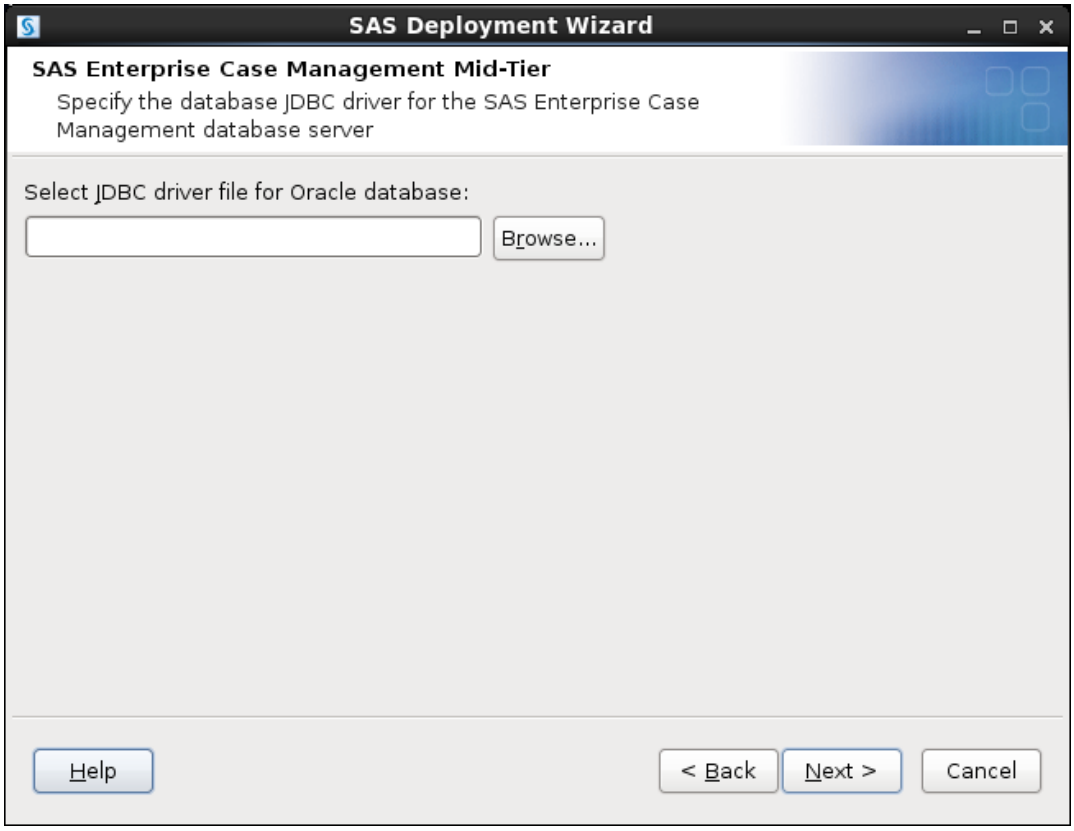
Password
specifies a valid password for the user name associated with the database account.

Note: The installation process attempts to connect to the database, to make sure that all of the information provided is accurate. If it cannot connect, you will receive a warning asking you to verify the information you have provided.

The following page prompts you to provide JDBC driver information for the middle tier. You will only see this prompt if your middle tier is not located on the same server as your server tier.

Note: You will see the page for the database you have selected. The image below is only an example.

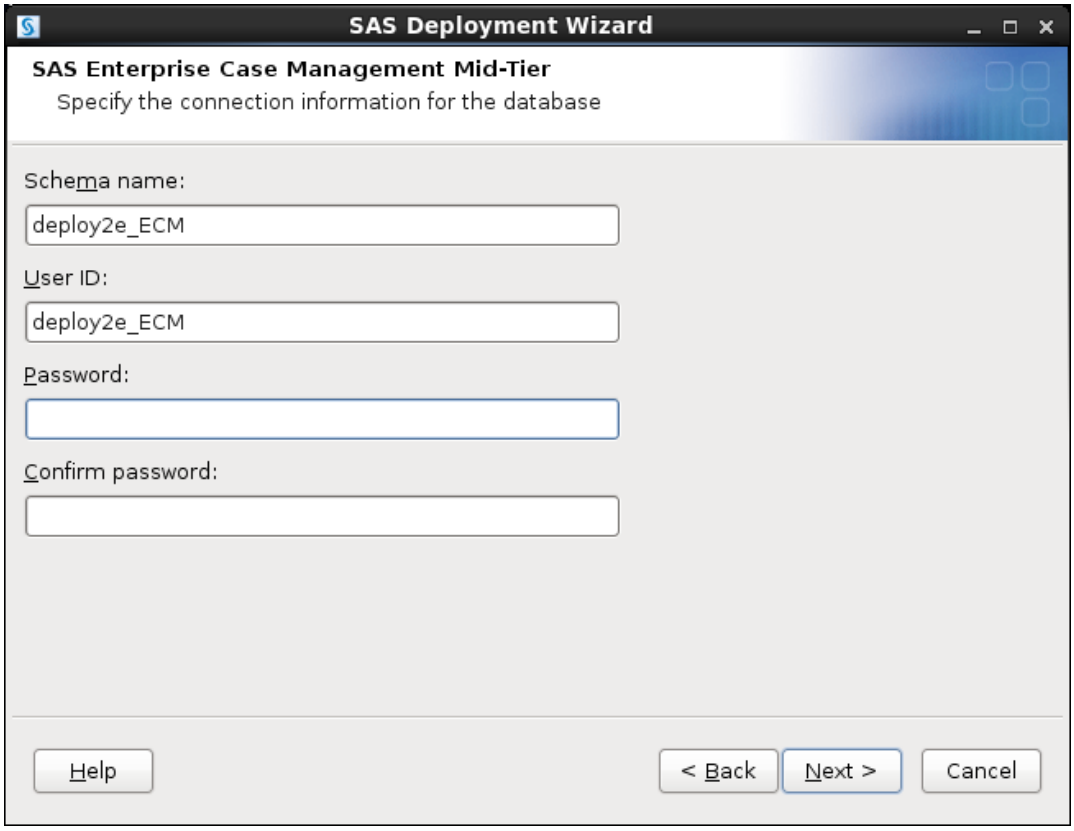
Display 3.9 *SAS Enterprise Case Management Middle-Tier Configuration – JDBC Driver Information*



You must provide the JDBC JAR file to facilitate Java access. It must be available on this host for configuration to take place.

The following page prompts you to provide database connection information for the middle tier. You will only see this prompt if your middle tier is not located on the same server as your server tier.

Display 3.10 *SAS Enterprise Case Management Middle-Tier Configuration – Database Connection Information*



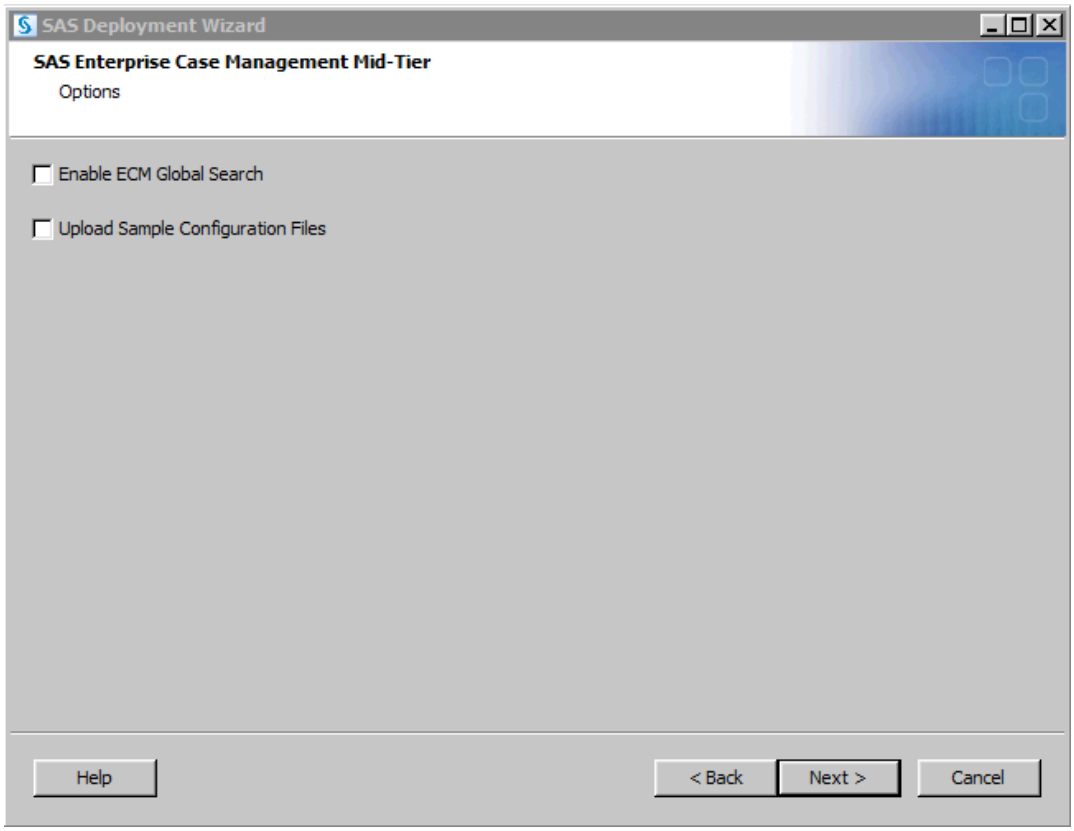
Schema name
specifies the schema name that was entered during the SAS Enterprise Case Management server configuration.

User ID
specifies the user name that was entered during the SAS Enterprise Case Management server configuration.

Password
specifies a valid password for the user name associated with the displayed database account.

The following page prompts you to select whether you want to enable the global search feature as well as run the sample definitions and properties and load the database with the samples.

Display 3.11 *SAS Enterprise Case Management Middle-Tier Configuration – Search, Definitions, Properties, and Samples Options*



Click the following check boxes, depending on your needs:

Enable ECM Global Search
enables the global search feature.

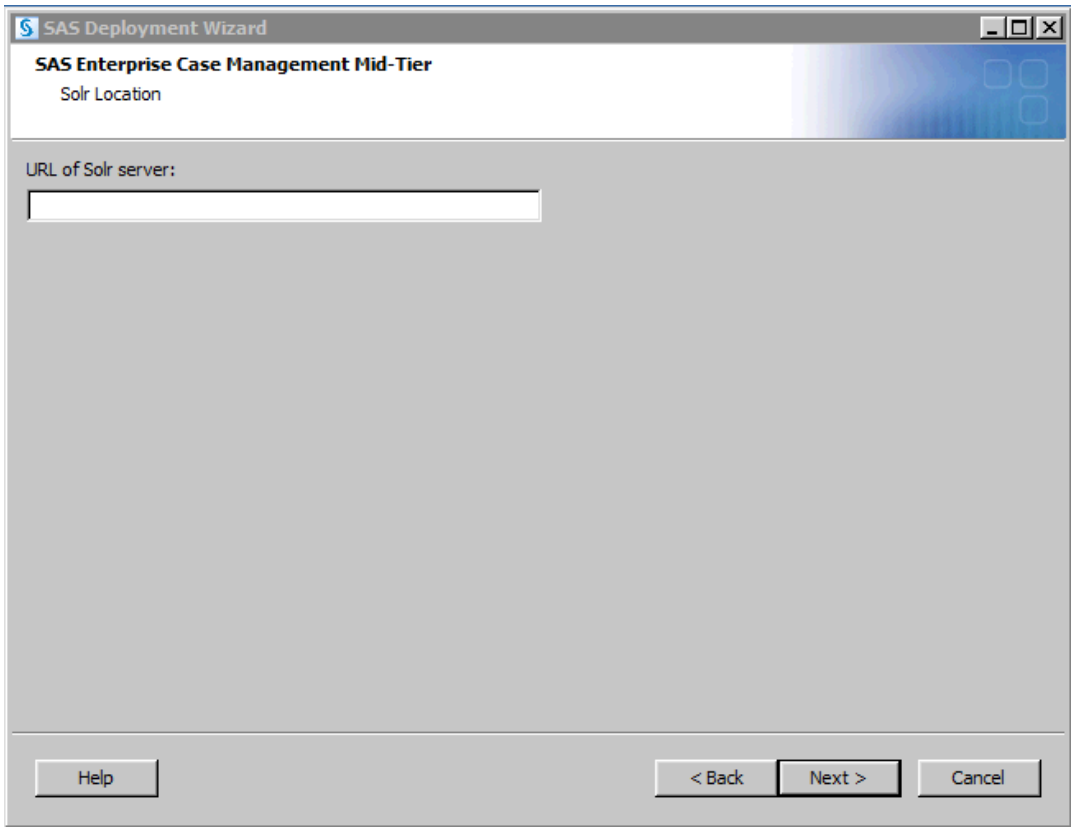
Note: Global search can be enabled later, if the box is not checked during configuration. For more information, see "Enabling or Disabling Global Search" on page 167.

Upload Sample Configuration Files
specifies whether or not to run the sample definitions and properties, as well as load the database with the samples.

Note: If you do not want the sample definitions and properties to be installed automatically, see "Uploading Definitions and Properties" on page 40, for manual instructions.

If you choose to enable the global search feature, the following prompt is shown asking for the location of the global search server. See "Pre-Installation: Global Search" on page 13, for more information about setting up a search server. If the search server uses the sample configuration, the URL for the search server will be **`http://<hostname>:8983/solr/EntCaseMgmtCollection`**.

Display 3.12 *SAS Enterprise Case Management Middle-Tier Configuration – Location of Global Search Server*



Follow the remaining page prompts for configuration until you reach the installation summary.

Note: If you are configuring the middle tier, you will also be prompted for SAS Social Network Analysis database credentials. For information on using SAS Social Network Analysis with a supported database, refer to "Creating the Social Network Analysis Database" on page 8.

Reviewing the Instructions.html File

After you have installed and configured your SAS software, the SAS Deployment Wizard writes an instructions file called Instructions.html to the **`Documents`** directory in your SAS configuration directory. The Instructions.html file contains additional information and details for configuring your installation. You can review this file for any additional steps to your installation.

Default File Locations

The following table shows the default locations of the directories and files that are installed with SAS Enterprise Case Management.

Table 3.1 *Default File Locations*

Directory/File	Windows Path	UNIX Path
SAS-installation-directory	**C:\Program Files*SASHome***	**/usr/local/*SASHome***
SAS-middle-tier-installation-directory	**C:\Program Files*SASHome*\SASEnterpriseCaseManagementMidTier\6.2**	**/usr/local/*SASHome*/**SASEnterpriseCaseManagementMidTier/6.2
!SASROOT	**C:\Program Files*SASHome*\SASFoundation\9.4**	*SAS-installation-directory*/**SASFoundation/9.4**
SAS-configuration-directory	**C:\SAS\Config**	**/usr/local/config**
SAS Enterprise Case Management Server <casemgmtmva>	*SAS-configuration-directory***Lev<num>\Applications\SASEnterpriseCaseManagement\6.2**	*SAS-configuration-directory*/**Lev<num>/Applications/SASEnterpriseCaseManagement/6.2**
SAS Enterprise Case Management Middle-Tier Staging Directory	*SAS-configuration-directory***Lev<num>\Web\Staging**	*SAS-configuration-directory*/**Lev<num>/Web/Staging**
SAS Enterprise Case Management Stored Processes	*SAS-configuration-directory***Lev<num>\Applications\SASEnterpriseCaseManagement\6.2\sasstp**	*SAS-configuration-directory*/**Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/sasstp**
SAS Enterprise Case Management Macro Definitions	*!SASROOT***casemgmtmva\ucmacros** and *SAS-configuration-directory* **\Lev<num>\Applications\SASEnterpriseCaseManagement\6.2\Source\ucmacros**	*!SASROOT*/**ucmacros/casemgmtmva** and *SAS-configuration-directory*/**Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros**
SAS Deployment Wizard Summary	*SAS-configuration-directory***Lev<num>\Documents\DeploymentSummary.html**	*SAS-configuration-directory*/**Lev<num>/Documents/DeploymentSummary.html**
Configuration Logs	*SAS-configuration-directory***Lev<num>\Logs\Configure**	*SAS-configuration-directory*/**Lev<num>/Logs/Configure**
SAS Enterprise Case Management Middle-Tier Web Log	*SAS-configuration-directory***Lev<num>\Web\Logs\SASEntCaseManagement6.2.log**	*SAS-configuration-directory*/**Lev<num>/Web/Logs/SASEntCaseManagement6.2.log**

Note: Depending on the architecture of your implementation, the *SAS-installation-directory* and *SAS-configuration-directory* location on the server tier might not be the same as those directories on the middle tier.

Updating the SAS SID File

The SAS installation data (SID) file will need to be updated at the appropriate SAS renewal period. The correct method to update an SID for Enterprise Case Management is to use the SAS Deployment Manager to apply the SID. Updating the SID file using the SAS License and Renewal feature for Base SAS installation will not update the system correctly. The SNA graph will not display and the SNA application log files will log a license error.

Apply All SAS Hotfix Updates

After completing the post-installation steps, apply any hotfixes required for all SAS components listed in your order. The SAS 9.4 hotfix download is found at: **`http://ftp.sas.com/techsup/download/hotfix/hotfix.html`**.

Chapter 4

Post-installation Requirements and Tasks

Post-installation Overview 32

SAS Social Network Analysis Configuration 32

Specify the Isolation Level for DB2 Databases 33

Sample Scripts Checklist 33

Clustering Support 34

Defining Users, Groups, and Roles 34
- Overview 34
- Defining Users in SAS Management Console 37
- Defining Groups in SAS Management Console 38
- Defining an Administrative User in the SAS Metadata Repository 38
- Defining Roles for SAS Enterprise Case Management Access 39

Customizing Your SAS Enterprise Case Management Installation 39
- Setting User Permission to the E-File Directory 39
- Loading the SAS Enterprise Case Management Configuration Tables 40

Uploading Definitions and Properties 40
- Clearing the Cache 40
- Uploading User Interface Definitions 41
- Uploading Custom Properties 41
- Uploading Workflow Definitions 42
- Uploading Menu Definitions 43

Capabilities in SAS Enterprise Case Management 43
- Associating Capabilities 43
- SAS Enterprise Case Management Capabilities – User Interface Impact 47

Subscriptions and Notifications 51
- Introduction to Subscriptions and Notifications 51
- Event Alert Notifications 52
- Case Report Notifications 54
- Task List Notifications 54
- Adjusting the Reminder Interval for the Task List 55

Notifications and the SAS Information Delivery Portal 55
- Configuring the SAS Information Delivery Portal for SAS Enterprise Case Management 55
- What to Expect from the Portal 56
- Controlling Alert Notifications from the SAS Preferences Manager 56

Configuring the Web Service 56

Deploy SAS Spelling Correction 56
Installation 57
Configuration of SAS Spelling Correction 59
Installing and Registering the SAS Spelling Server as a Windows Service 60

Post-installation Overview

At the end of the installation process, the SAS Deployment Wizard produces an HTML document named Instructions.html. If your server tier and middle tier are hosted on separate machines, you will have an Instructions.html file for each machine. To complete your installation, you will need the information that is provided in Instructions.html and the information specific to SAS Enterprise Case Management that is documented in this chapter.

SAS Social Network Analysis Configuration

SAS Enterprise Case Management uses the SAS Social Network Analysis web application to display Case Network Analysis graphs. To enable Case Network Analysis, SAS Social Network Analysis server tables and sequences must be created and initialized. If you have not already initialized these tables and sequences, see the "Creating Database Tables" section of the *SAS Social Network Analysis Server: Installation and Configuration Guide*, located at **`http://support.sas.com/documentation/solutions/sna/index.html`**.

After the DDLs are executed, the SAS Social Network Analysis Server is configured with a default set of database settings that control the interface. You can update the following settings in the SNA_CONTEXT_PREFERENCE table to deliver an optimized interface specific to Case Network Analysis. For the complete list of parameters, refer to the *SAS Social Network Analysis Server: Administration Guide*.

Parameter	Description	Default Setting
comment_manager_flg	Enables the Comment Manager.	Enabled
logout_warning_time	Determines how many minutes before the session time-out to present the user with a logout warning message. The default session time-out interval is 30 minutes.	Five minutes before session time-out.

To view Case Network Analysis graphs, you need to have proper access. A folder called ECM should be added in SAS Management Console under **System** ⇨ **Applications** ⇨ **SAS Social Network Analysis** ⇨ **Social Network Analysis 6.2**. SAS Enterprise Case Management Users should have read metadata access to the folder.

To set the user permission, edit the properties of the ECM folder. On the **Authorization** tab, click **Add**. Then move the SAS Enterprise Case Management Users group from

Available Identities to the **Selected Identities** and click **OK**. Accept the default permissions and click **OK** to save the folder properties.

Note: The ECM folder is used by the SAS Social Network Analysis component. It must be called ECM to match the prefix of the ECMSocialNetworkAnalysis web service.

Specify the Isolation Level for DB2 Databases

If SAS Enterprise Case Management is connected to a DB2 database, some database connections might become locked and cause usage problems. This is caused by the default isolation level for DB2, which is set to Repeatable Reads. When SAS Enterprise Case Management performs SELECT and UPDATE operations for the same record in one transaction, the UPDATE operation will be locked as it waits for the SELECT operation to release the lock.

In order to avoid these problems, you should set the isolation level to Read Committed, so the SELECT operation will not lock the record, and the UPDATE operation can be performed.

Sample Scripts Checklist

SAS Enterprise Case Management is shipped with sample scripts to demonstrate its functionalities and show how it can be customized for customer-specific implementation. The following checklist provides an overview of the post-installation steps to implement the samples. For implementation of the FINCEN Suspicious Activity Report (SAR) and the Currency Transaction Report (CTR), refer to "Regulatory Reports and E-Filing" on page 170.

1. Create an installer SAS Enterprise Case Management account.

   Note: In SAS Enterprise Case Management 6.2, the installer account is no longer automatically defined as a valid Enterprise Case Management user. Refer to "Defining Users, Groups, and Roles" on page 34 to create the account before moving to step 2.

2. Run the post installation script load_post_install_data.sas. To do this, follow the instructions in "Loading the SAS Enterprise Case Management Configuration Tables" on page 40.

3. Clear the cache. Follow the instructions in "Clearing the Cache" on page 40 to clear the old configuration data in memory.

4. Upload the user interface definitions. Follow the instructions in "Uploading User Interface Definitions" on page 41 to upload all the user interface definition sample files except for FINCEN report-related samples, which are rr-fincen-ctr-02.xml, rr-fincen-sar-01.xml, and efile-fincen-new-01.xml.

   Note: This step can be skipped if the samples were uploaded during configuration.

5. Upload the custom properties. Follow the instructions in "Uploading Custom Properties" on page 41.

   Note: This step can be skipped if the samples were uploaded during configuration.

6. Upload the workflow definitions. Follow the instructions in "Uploading Workflow Definitions" on page 42 to upload CaseManagementFinancialFraud.xml and CaseManagementGeneric.xml.

   Note: This step can be skipped if the samples were uploaded during configuration.

7. Configure the web services. Follow the instructions in "Configuring the Web Service" on page 56 to allow BI web services to accept warnings.

8. Deploy SAS Spelling Correction. Follow the instructions in "Deploy SAS Spelling Correction" on page 56 to enable spell checking in SAS Enterprise Case Management.

Clustering Support

You can deploy SAS Enterprise Case Management in a cluster, providing fault tolerance and scalability. SAS Enterprise Case Management has the same limitations as the SAS Intelligence Platform, whereby each user's session must be tied to a single server. For more information about how to configure SAS Web applications in a cluster, see the *SAS Intelligence Platform: Middle-Tier Administration Guide* at **`http://support.sas.com/documentation/onlinedoc/intellplatform`**.

Defining Users, Groups, and Roles

Overview

You must configure users, groups, and roles to use SAS Enterprise Case Management. The deployment process created several groups and roles for SAS Enterprise Case Management that you can use to get started.

Users
: Every user who needs to log on to the SAS Enterprise Case Management web application must be defined in the SAS Metadata Repository and be associated with one or more groups and one or more roles that have one or more capabilities within SAS Enterprise Case Management. Every SAS Enterprise Case Management user should be a member of the Enterprise Case Management Users group.

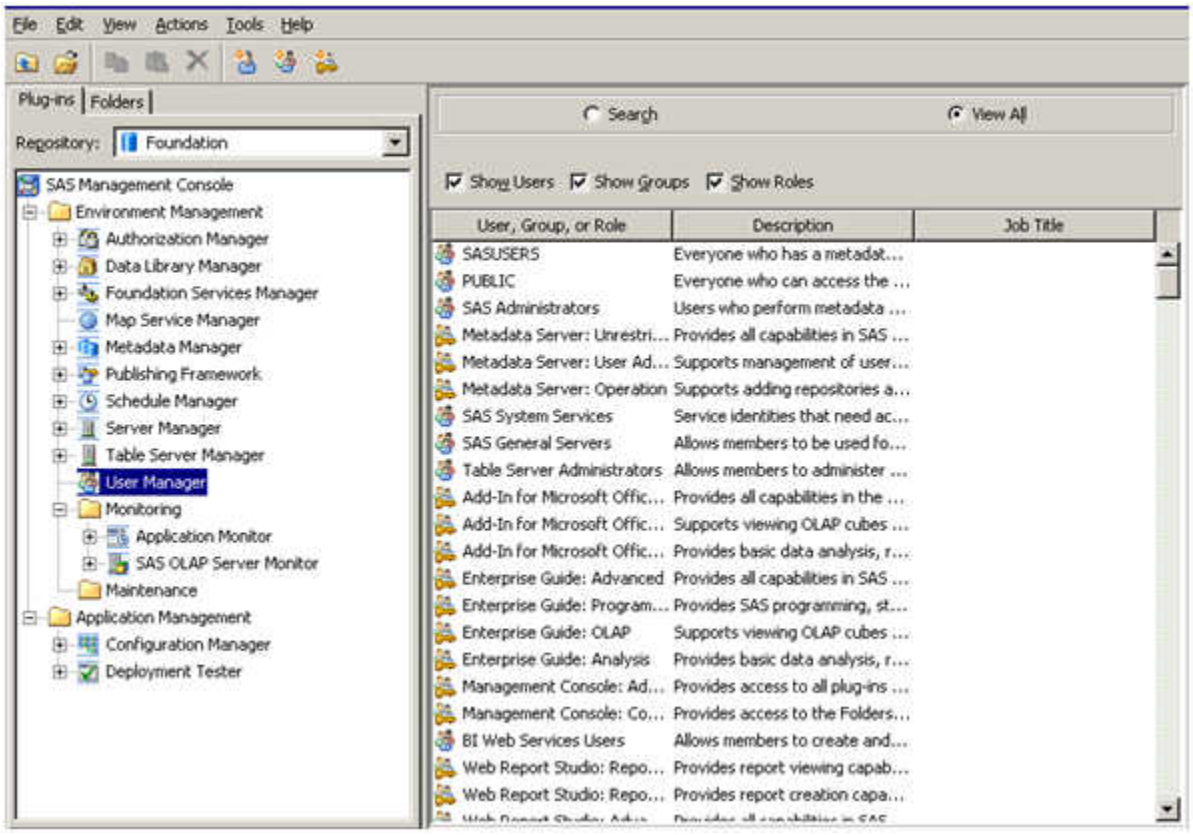
Groups
: A group is a group of users classified by common traits or common data access levels. Groups are typically used for giving users access to data. Groups can also be used within workflows to allow a restricted set of users to perform an activity. The Enterprise Case Management Users group is pre-loaded during installation. It enables members to access SAS Enterprise Case Management. You must define all other groups.

Roles
: A role provides a grouping functionality. Roles determine what a user can do within the application. Roles can also be used within workflows to allow a restricted set of users to perform an activity. The Case Management: Advanced role is pre-loaded

during installation. It provides all capabilities in SAS Enterprise Case Management. You must define all other roles.

Groups, roles, and users are defined with the User Manager function in SAS Management Console, as shown in the following display.

Display 4.1 *User Manager – SAS Management Console*



Note: For specific information about defining users, groups, and roles, see the *SAS Management Console: Guide to Users and Permissions*.

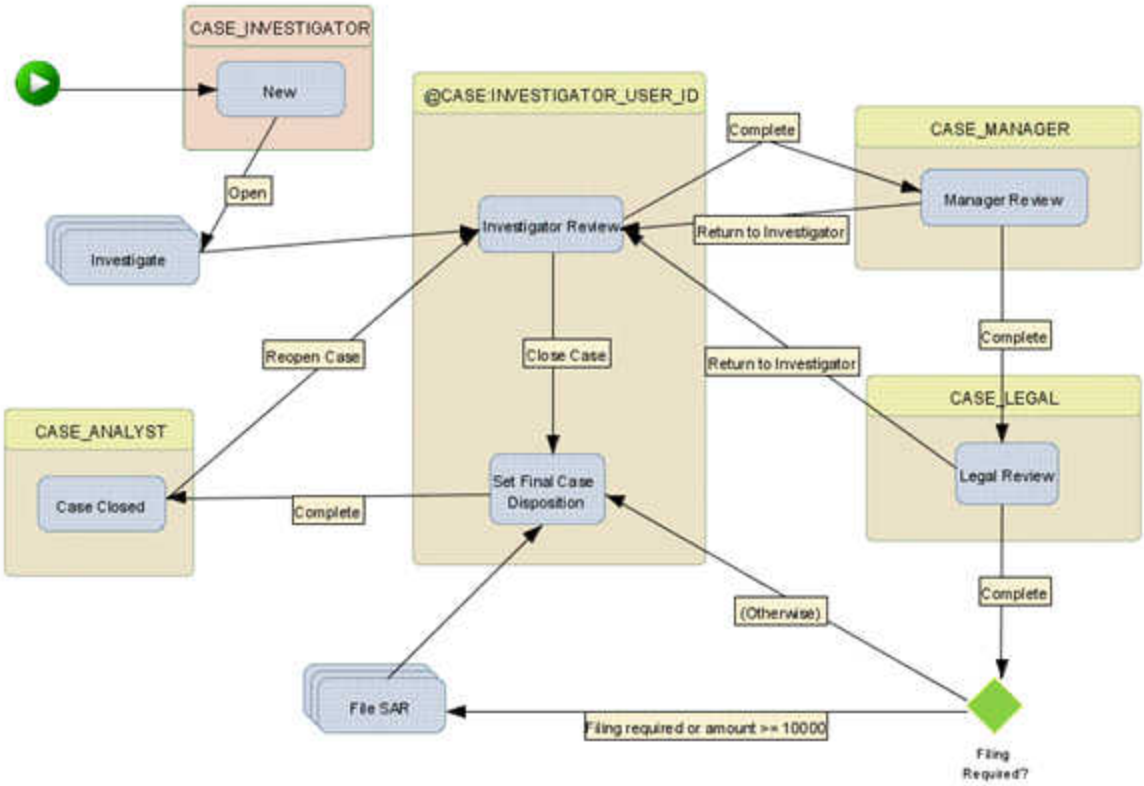
Groups and roles can be used as drop-down lists when configured as reference tables on the search panel. Preparation of reference tables and search panels are discussed in Customizing SAS Enterprise Case Management on page 62.

Groups, roles and users can be used in workflow definitions to determine who can perform activities in the investigative process. In the following display, the following roles are referenced in the workflow definition:

- CASE_INVESTIGATOR
- CASE_ANALYST
- CASE_MANAGER
- CASE_LEGAL

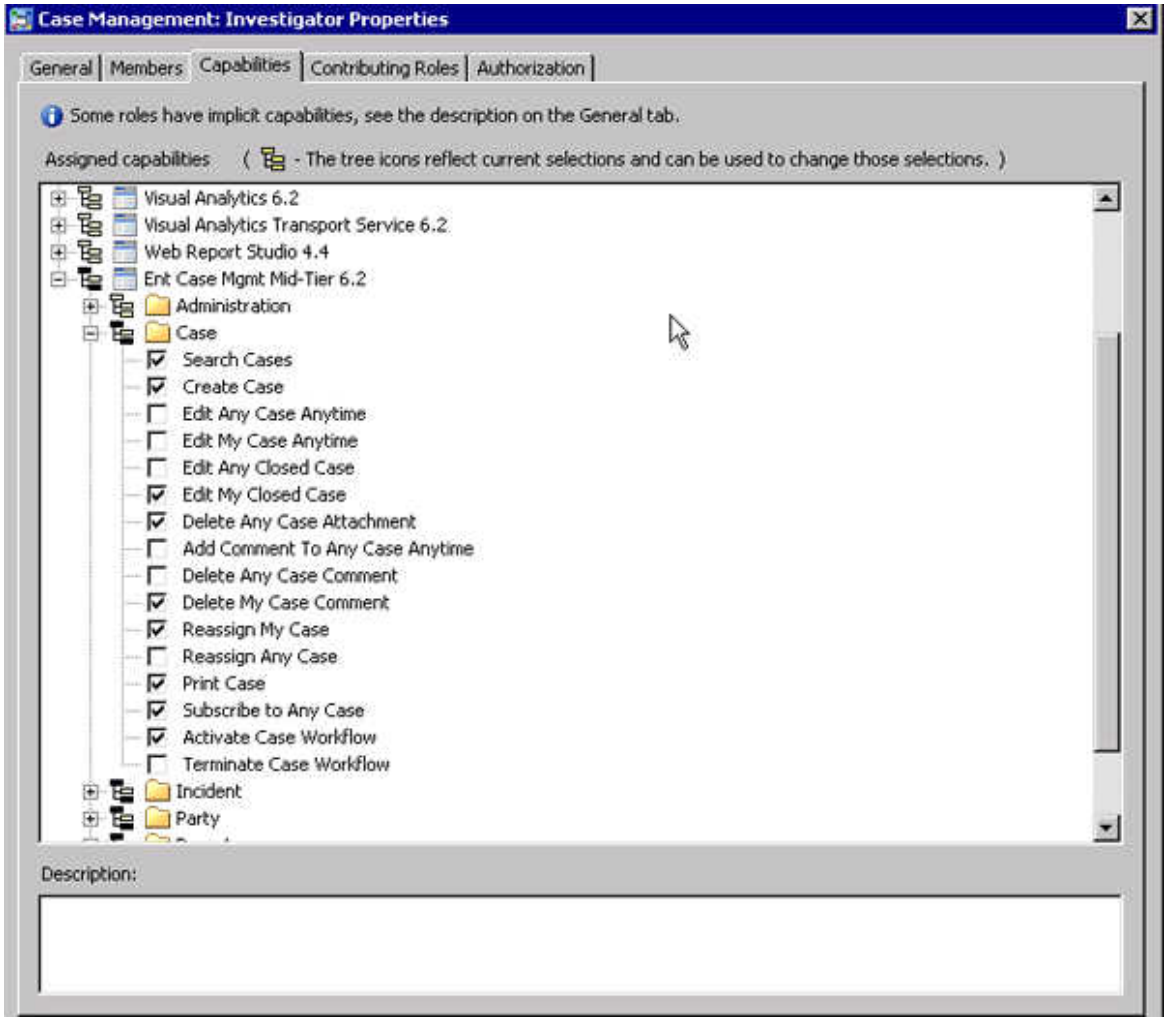
The display also shows the data object @CASE_INVESTIGATOR_USER_ID, which refers to the primary owner of the case. For more information about how the primary owner is set, see "Assigning the Primary Owner to a Case" on page 241.

Display 4.2 *Roles – Workflow Definition*



Capabilities can be associated with roles in the SAS Management Console as shown in the following display.

Display 4.3 *Roles – Capabilities*



Groups, roles and users can be referenced in user interface definitions. The following user interface definition shows how to get the display name for the primary owner for a case.

```
<field name="CASE.INVESTIGATOR_USER_ID" type="string" required="false"
       values="GetUserDisplayName(CASE.INVESTIGATOR_USER_ID)">
   <label>
      <message key="field.case.investigator_user_id.label.txt" />
   </label>
</field>
```

Users and groups can be referenced in configuration tables. You can specify the primary owner for newly created cases. You can also specify the initial set of groups that have access to newly created cases, incidents, and parties.

Defining Users in SAS Management Console

To define users, log on to the SAS Management Console as a user who has the capability to manage users, groups, and roles. For details, see the online Help for the New User page.

1. Select **Administration** ⇨ **Users and Group** ⇨ **Users** ⇨ **New**.

2. Click the **User Manager** plug-in.
3. Select **Actions** ⇨ **New** ⇨ **User**. The New User Properties window appears.
4. Enter valid data in the **Name** and **Display Name** text boxes.
5. Select the **Groups and Roles** tab.
6. Click the **Accounts** tab.
7. Click **New**. The New Login Properties dialog box appears.
8. Define logon information for the user.
9. Click **OK**. For more information, see the online Help in SAS Management Console or *SAS Management Console: Guide to Users and Permissions*.

Defining Groups in SAS Management Console

A SAS Enterprise Case Management group is defined by default during installation. If necessary, you can add users to this group to access SAS Enterprise Case Management. To create additional groups in SAS Management Console, complete these steps:

1. Log on to SAS Management Console as sasadm or as a user who has the capability to manage users, groups, and roles.
2. Click the **User Manager** plug-in.
3. Select **Actions** ⇨ **New** ⇨ **Group**. The New Group Properties window appears.
4. Enter valid data in the **Name** and **Display Name** text boxes.
5. Select the **Members** tab.
6. From the **Available Identities** list, select users that you want to be members of the group and move them to the **Current Members** list. Note that you can define users later if they are not yet defined.
7. Click **OK**.

   For more information, see the online Help in SAS Management Console or *SAS Management Console: Guide to Users and Permissions*.

Defining an Administrative User in the SAS Metadata Repository

There is no limit to the number of users that can administer SAS Enterprise Case Management. However, there must be least one. The following steps describe creating users in SAS metadata to act as SAS Enterprise Case Management administrators. In SAS Management Console:

1. Select the **User Manager** tab.
2. Select **New User** and enter the required information for the user, including adding a user account on the **Accounts** tab.
3. Make the user a member of the Enterprise Case Management Users group on the **Groups and Roles** tab.
4. Make the user a member of a role (or a group that is a member of a role) that has administrative capabilities for SAS Enterprise Case Management. Examples of this are the Case Management: Administrator and Case Management: Advanced roles.

Defining Roles for SAS Enterprise Case Management Access

Roles in SAS Enterprise Case Management are activity based. Roles are granted to users and are cumulative. For example, if a user is assigned to more than one role, then the capabilities will always honor the grant. If role 1 grants a user a specific capability but role 2 does not, the user will still have the capability. The **`Advanced`** role is provided with your SAS Enterprise Case Management installation by default.

1. Log on to SAS Management Console as sasadm or as a user who has the capability to manage users, groups, and roles.
2. Click the **User Manager** plug-in.
3. Select **Actions** ⇨ **New** ⇨ **Role**. The New Role Properties window appears.
4. Enter valid data in the **Name** and **Display Name** text boxes.
5. Select the **Members** tab.
6. From the **Available Identities** list, select users that you want to assign to the role and move them to the **Current Members** list. Note that you can define users later if they are not yet defined.
7. Select the **Capabilities** tab. All of the capabilities from all of the installed applications are displayed.
8. You can select a capability or capabilities.
9. Click **OK**. For more information, see the online Help in SAS Management Console or *SAS Management Console: Guide to Users and Permissions*.

Customizing Your SAS Enterprise Case Management Installation

Setting User Permission to the E-File Directory

The SAS Enterprise Case Management stored process generates e-files for regulatory reporting. The network account defined as the SAS Spawned Servers account should have Write permission to the following directories:

- *SAS-configuration-directory*/**`Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/efiles`**
- *SAS-configuration-directory*/**`Lev<num>/Applications/SASEnterpriseCaseManagementg/6.2/Source/efiles/fincen-sar-v1`**
- *SAS-configuration-directory*/**`Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/efiles/fincen-newctr-v1`**

Note: The subfolders **`fincen-sar-v1`** and **`fincen-newctr-v1`** are created when configuring e-filing, as discussed in "Configuring E-Filing " on page 173.

Loading the SAS Enterprise Case Management Configuration Tables

After you have completed your installation, you should customize your SAS Enterprise Case Management installation. To populate the SAS Enterprise Case Management configuration tables, you should provide your own data. The SAS file load_post_install_data.sas contains sample data.

To execute the sample file, follow these steps:

Note: If you selected **Bypass Database Initialization** during SAS Deployment Wizard configuration, refer to "Post-installation Database Steps Required after Unsuccessful SAS Deployment Wizard Database Installation" on page 237 to initialize the database tables before proceeding with the steps described below.

1. From the SAS Enterprise Case Management configuration directory (**`/Lev1/Applications/SASEnterpriseCaseManagement/6.2/Source/control`**), start a SAS session and open the ecm_autoexec.sas file in the session.
2. Select **Run** to execute this file in an interactive SAS session.
3. Enter **`%ecm_db_connect;`** in the program editor, and click **Run** to execute the macro. Verify that the SAS Enterprise Case Management tables were created in the ecm_db database. If you get errors when executing this macro, you are not a valid SAS Enterprise Case Management user. You should either log on with an ID that is a member of the SAS Enterprise Case Management Users group or add yourself as a SAS Enterprise Case Management user. The instructions for defining a SAS Enterprise Case Management user are discussed in "Defining Users in SAS Management Console" on page 37.
4. From the following directory, depending on your platform, open the installation configuration program, such as load_post_install_data.sas, in SAS:
   - Windows platforms: *!SASROOT***`casemgmtmva\sasmisc\sample\config`**
   - UNIX platforms: *!SASROOT*/**`misc/casemgmtmva/sample/config`**
5. Select **Run** to execute this file in SAS.

Uploading Definitions and Properties

Definitions and properties can be installed automatically by selecting the **Include optional sample programs** check box during the SAS Deployment Wizard process (see "Specifying DBMS Credentials" on page 17 for more information). However, if you prefer to install them manually, follow the steps in this section.

Clearing the Cache

SAS Enterprise Case Management caches various configuration data in memory for better performance. The following configurations are cached in memory:

- all user-defined field definitions
- all static and user-defined reference table values

- all user display names defined in the SAS Metadata Repository
- all search panel and global search configurations
- all custom resource bundle properties
- all properties defined in the SAS Metadata Repository for SAS Enterprise Case Management

If any of the above configurations are changed, then the administrator should go to the **Administration** tab in the SAS Enterprise Case Management web application and select the **Clear Cache** menu option to clear cached data.

Uploading User Interface Definitions

Upload the user interface definitions from one of the following paths, if they have not been uploaded during configuration:

- Windows platforms: *SAS-middle-tier-installation-directory*`\deploy\sample\UIDefinitions`
- UNIX platforms: *SAS-middle-tier-installation-directory*`/deploy/sample/UIDefinitions`

First, upload the user interface definitions for the standard object creation dialog boxes by performing the following steps:

1. Log on to SAS Enterprise Case Management as the admin user.
2. Select **Clear Cache** on the **Administration** tab to ensure that the application has the latest configuration data.
3. Select **UI Definitions** on the **Administration** tab.
4. Click the **Upload** icon to upload the needed user interface definitions. You must upload each file individually.

After uploading the object creation dialog boxes, upload user interface definitions for each entity type. Sample user interface definition files are provided from one of the following paths:

- Windows platforms: *SAS-middle-tier-installation-directory*`\deploy\sample\UIDefinitions`
- UNIX platforms: *SAS-middle-tier-installation-directory*`/deploy/sample/UIDefinitions`

To upload the samples, follow these steps:

1. Select **UI Definitions** on the **Administration** tab.
2. Upload the needed user interface definitions. You must upload each file individually.

Uploading Custom Properties

Upload your custom property definitions in SAS Enterprise Case Management from one of the following paths:

- Windows platforms: *SAS-middle-tier-installation-directory*`\deploy\sample\properties`
- UNIX platforms: *SAS-middle-tier-installation-directory*`/deploy/sample/properties`

For example, the following file that contains sample custom properties can be executed. To use this sample, follow these steps:

1. Log on to SAS Enterprise Case Management as the admin user.
2. Select **Custom Property Files** on the **Administration** tab.
3. Upload the needed custom property definitions. You must upload the files individually.
4. After all the custom properties files have been uploaded, click **Refresh Report Mart Labels** on the **Administration** tab.

   Note: Report mart labels should be refreshed any time changes are made to the table or column label translations in any of the custom properties files or any time a new language is added to the ECM_LOCALE table. Report mart labels are generated only for the locales in the ECM_LOCALE table.

Uploading Workflow Definitions

After you have set up and populated your database, you must upload your workflow definitions. You can provide your own workflow definitions or use the samples. Samples of the workflow definitions can be found from one of the following paths:

- Windows platforms: *SAS-middle-tier-installation-directory*`\deploy\sample\Workflows`
- UNIX platforms: *SAS-middle-tier-installation-directory*`/deploy/sample/Workflows`

To use these samples, follow these steps:

1. Start the Windows Workflow Studio Client by selecting **All Programs** ⇨ **SAS** ⇨ **SAS Workflow Studio 1.3**. For UNIX, you must copy the workflow to a directory that is accessible by SAS Workflow Studio.

   Note: SAS Workflow Studio will likely be located in *SAS-installation-directory* `\SASWorkflowStudio\1.3`.
2. Log on to the workflow engine by selecting **Server** ⇨ **Logon**. Connect to the web server running the BI web services using `http://<hostname>:<portnumber>`. This is an application that is part of the Web Infrastructure Platform (WIP), which by default is usually running on server 1 if there are multiple web servers running the SAS middle-tier web applications. Use the SAS Administrator (sasadm@saspw) to connect.
3. In the **File** menu, open all of the XML files from the sample directory.
4. Complete the following steps for the SERVER_URL data object when you create new workflows or when you upload the sample workflows.

   a. (Optional) In the process tree list of the workflow that you are working on, expand **Data Objects**.

   b. (Optional) Right-click **SERVER_URL** and select **Edit** to edit the data object. Specify the host name and port number of the SAS Enterprise Case Management web application in the **Properties Text** text field. There is more information about the expected value in the description of the Data Object.

   c. Upload the edited workflow process template by choosing **Server** ⇨ **Save to Repository**.

d. Check the **Activate** check box at the bottom of the dialog, if you would also like to activate the current version in workflow. Then click **OK**.

Note: If you do not click **Activate** for a workflow that is being uploaded for the first time, you will receive an error telling you that an invalid workflow has been configured when you open cases that are configured to use that workflow.

5. Repeat the **Save to Repository** process in SAS Workflow Studio for each workflow definition file.

Uploading Menu Definitions

SAS Enterprise Case Management installs a default navigation menu, which can be customized. To customize the navigation menu:

1. Log on to SAS Enterprise Case Management as the admin user.
2. Select **Menu Definitions** on the **Administration** tab.
3. Click the action menu next to **MainNavigationMenu.xml**, and select **Update Menu Definition**. Select a file to upload. This file will be uploaded with the name MainNavigationMenu.xml, regardless of the name of the local file that you select to upload to replace the existing uploaded menu definition.

   You can find the initial version of the file at *SAS-middle-tier-installation-directory* **`\deploy\sample\menus`**.

   Note: The sample menu definition file is on the middle tier, not with the other sample configuration files on the server tier.
4. Log out and then log back in for the navigation menu changes to take effect.

Capabilities in SAS Enterprise Case Management

Associating Capabilities

SAS Enterprise Case Management provides various functional capabilities that are applicable to cases, incidents, parties, reports, and e-files. The following tables show the different capabilities.

Note: All the capabilities listed apply to the **`Advanced`** role in SAS Enterprise Case Management.

Table 4.1 *SAS Enterprise Case Management – Case Capabilities*

Case Capability	Description
Search Cases	Enables users to search for cases. Users must have the Search Cases capability for any of the subsequent capabilities to take effect.
Create Case	Enables users to create a case.

Case Capability	Description
Edit Any Case Anytime	Enables users to edit any case anytime.
Edit My Case Anytime	Enables users to edit their own cases anytime.
Edit Any Closed Case	Enables users to edit any closed case.
Edit My Closed Case	Enables users to edit their own closed cases.
Add Comment To Any Case Anytime	Enables users to add a comment to any case anytime.
Delete Any Case Attachment	Enables users to delete any case attachment.
Delete Any Case Comment	Enables users to delete any case comment.
Delete My Case Comment	Enables users to delete any case comment that they created.
Reassign Any Case	Enables users to set the primary owner for any case and unlock any case.
Reassign My Case	Enables users to reassign any case that they own.
Subscribe to Any Case	Enables users to subscribe to any case for alerting when the case is modified.
Print Case	Enables users to preview or generate a printable case report.
Activate Case Workflow	Enables users to activate a particular workflow for the case.
Terminate Case Workflow	Enables users to terminate a workflow associated with the case.

Table 4.2 *SAS Enterprise Case Management – Incident Capabilities*

Incident Capability	Description
Search Incidents	Enables users to search for incidents. Users must have the Search Incidents capability for any of the subsequent capabilities to take effect.
Create Incident	Enables users to create an incident.
Edit Incident	Enables users to edit an incident.
Delete Any Incident Attachment	Enables users to delete any incident attachment.
Add Comment to Any Incident	Enables users to add a comment to any incident.

Incident Capability	Description
Delete Any Incident Comment	Enables users to delete any incident comment.
Delete My Incident Comment	Enables users to delete any incident comment that they created.
Subscribe to Any Incident	Enables users to subscribe to any incident for alerting when the incident is modified.

Table 4.3 *SAS Enterprise Case Management – Party Capabilities*

Party Capability	Description
Search Parties	Enables users to search for parties. Users must have the Search Parties capability for any of the subsequent capabilities to take effect.
Create Party	Enables users to create a party or subject.
Edit Party	Enables users to edit a party or subject.
Delete Any Party Attachment	Enables users to delete any party or subject attachment.
Add Comment To Any Party	Enables users to add a comment to any party or subject.
Delete Any Party Comment	Enables users to delete any party or subject comment.
Delete My Party Comment	Enables users to delete any party or subject comment that they created.
Subscribe to Any Subject	Enables users to subscribe to any party or subject for alerting when the party or subject is modified.

Table 4.4 *SAS Enterprise Case Management – E-Filing Capabilities*

E-Filing Capability	Description
Search E-Files	Enables users to search for e-files. Users must have the Search E-Files capability for any of the subsequent capabilities to take effect.
Create E-File	Enables users to create an e-file or subject.
Edit E-File	Enables users to edit an e-file or subject.
Delete Any E-File Attachment	Enables users to delete any e-file or subject attachment.

E-Filing Capability	Description
Add Attachment to E-File	Enables users to add an attachment to any e-file.
Add Comment To Any E-File	Enables users to add a comment to any e-file or subject.
Delete Any E-File Comment	Enables users to delete any e-file or subject comment.
Delete My E-File Comment	Enables users to delete any e-file or subject comment that they created.
Subscribe to Any E-File	Enables users to subscribe to any e-file or subject for alerting when the e-file or subject is modified.

Table 4.5 *SAS Enterprise Case Management – Report Capabilities*

Report Capability	Description
Search Reports	Enables users to search for reports. Users must have the Search Reports capability for any of the subsequent capabilities to take effect.
Create Report	Enables users to create a report.
Edit Report	Enables users to edit any report.
Add Comment To Any Report	Enables users to add a comment to a report.
Delete Any Report Attachment	Enables users to delete any report attachment.
Delete Any Report Comment	Enables users to delete any report comment.
Delete My Report Comment	Enables users to delete any report comment that they created.
Subscribe to Any Report	Enables users to subscribe to any report for alerting when the report is modified.
Add Attachment to Report	Enables users to add an attachment to a report.
Activate Report Workflow	Enables user to activate a particular workflow for the report.
Terminate Report Workflow	Enables the user to terminate a workflow associated with the report.

Table 4.6 *SAS Enterprise Case Management – Relational Capabilities*

Relational Capability	Description
Add Incident To Case	Enables users to add an incident to a case.

Table 4.7 *SAS Enterprise Case Management – Administration Capabilities*

Administration Capability	Description
Upload UI Definitions	Enables the user to upload UI definitions from the **Administration** menu.
Upload Custom Properties	Enables the user to upload Custom Properties files from the **Administration** menu.
Upload Search Configuration	Enables the user to upload Global Search configuration files from the **Administration** menu.
Clear Cache	Enables the user to clear the configuration data cache from the **Administration** menu.
Refresh Labels	Enables the user to recompute the labels associated with report mart tables from the **Administration** menu.
View Documentation	Enables the user to view the Custom Page Builder documentation from the **Administration** menu.

Table 4.8 *SAS Enterprise Case Management – Search Capability*

Search Capability	Description
Bulk Load Index	Enables the user to update the search index in SAS.

SAS Enterprise Case Management Capabilities – User Interface Impact

Table 4.9 *SAS Enterprise Case Management – Case Capability Impact*

Case Capability	User Interface Impact
Search Cases	The **Cases** application tab is visible.
Create Case	The **New Case** action is visible on the cases search panel toolbar.
Edit Any Case Anytime	The **Edit** menu action is always enabled.

Case Capability	User Interface Impact
Edit My Case Anytime	The **Edit** menu action is always enabled for any case that the user owns.
Edit Any Closed Case	The **Edit** menu action is always enabled for any closed case.
Edit My Closed Case	The **Edit** menu action is always enabled for any closed case that the user owns.
Delete Any Case Attachment	The Delete Attachment action icon is visible for all attachments on any case the user can edit.
Delete Any Case Comment	The Delete Comment action icon is visible for all comments on any case the user can edit or add a comment to.
Delete My Case Comment	The Delete Comment action icon is visible for all comments created by the user on any case the user can edit or add a comment to.
Add Comment To Any Case Anytime	The **Comment** input fields and button are always visible. Without this capability, the **Comment** input fields and button are visible only on cases the user can edit.
Reassign Any Case	The **Set Primary Owner** and **Unlock** menu actions for a case are always enabled.
Reassign My Case	The **Reassign Case** menu action is enabled for cases that the user owns.
Print Case	Users can preview or generate a printable case report.
Activate Case Workflow	Users can activate a particular workflow for the case.
Terminate Case Workflow	Users can terminate a workflow associated with the case.

Table 4.10 *SAS Enterprise Case Management – Incident Capability Impact*

Incident Capability	User Interface Impact
Search Incidents	The **Incidents** application tab is visible.
Create Incident	The **New Incident** action is visible on the incident search panel toolbar.
Edit Incident	The **Edit** menu action is always enabled.

Incident Capability	User Interface Impact
Delete Any Incident Attachment	The Delete Attachment action icon is visible for all attachments on any incident the user can edit.
Add Comment To Any Incident	The **Comment** input fields and button are always visible. Without this capability, the **Comment** input fields and button are visible only on incidents the user can edit.
Delete Any Incident Comment	The Delete Comment action icon is visible for all comments on any incident the user can edit or add a comment to.
Delete My Incident Comment	The Delete Comment action icon is visible for all comments created by the user on any incident the user can edit or add a comment to.

Table 4.11 *SAS Enterprise Case Management – Party Capability Impact*

Party Capability	User Interface Impact
Search Parties	The **Subjects** application tab is visible.
Create Party	The **New Subject** action is visible on the party search panel toolbar.
Edit Party	The **Edit** menu action is always enabled.
Delete Any Party Attachment	The Delete Attachment action icon is visible for all attachments on any party the user can edit.
Add Comment To Any Party	The **Comment** input fields and button are always visible. Without this capability, the **Comment** input fields and button are visible only on parties the user can edit.
Delete Any Party Comment	The Delete Comment action icon is visible for all comments on any party the user can edit or add a comment to.
Delete My Party Comment	The Delete Comment action icon is visible for all comments created by the user on any party the user can edit or add a comment to.

Table 4.12 *SAS Enterprise Case Management – Report Capability Impact*

Report Capability	User Interface Impact
Search Reports	The **Reports** application tab is visible.

Report Capability	User Interface Impact
Create Reports	The **Add Report** action is visible on the table of reports in a case or incident.
Edit Reports	The **Edit** menu action is always enabled.
Delete Any Report Attachment	The Delete Attachment action icon is visible for all attachments on any report the user can edit.
Delete Any Report Comment	The Delete Comment action icon is visible for all comments on any report the user can edit or add a comment to.
Delete My Report Attachment	The Delete Comment action icon is visible for all comments created by the user on any report the user can edit or add a comment to.
Add Comment To Any Report Anytime	The **Comment** input fields and button are always visible. Without this capability, the **Comment** input fields and button are visible only on reports the user can edit.
Activate Report Workflow	Users can activate a particular workflow for the report.
Terminate Report Workflow	Users can terminate a workflow associated with the report.

Table 4.13 *SAS Enterprise Case Management – Relational Capability Impact*

Relational Capability	User Interface Impact
Add Incident To Case	The **Related Cases** menu action (on incident search panel) and toolbar action (on incident detail panel) are always enabled for unassigned incidents.

Table 4.14 *SAS Enterprise Case Management – Administration Capability Impact*

Administration Capability	Description
Upload UI Definitions	**UI Definitions** is visible on the **Administration** menu.
Upload Custom Properties	**Custom Property Files** is visible on the **Administration** menu.
Upload Search Configuration	**Search Configuration** is visible on the **Administration** menu.
Clear Cache	**Clear Cache** is visible on the **Administration** menu.

Administration Capability	Description
Refresh Labels	**Refresh Report Mart Labels** is visible on the **Administration** menu.
View Documentation	**View Custom Page Builder Documentation** is visible on the **Administration** menu.

Table 4.15 *SAS Enterprise Case Management – Search Capability Impact*

Search Capability	Description
Bulk Load Index	No user interface impact.

You should consider the following specific details about capabilities and workflows and their impact on the SAS Enterprise Case Management user interface:

- If a user is able to work on a workflow activity as defined in the associated workflow for a case, then that case can be edited, regardless of the user's capabilities.
- Attachments can be added only to cases, incidents, or parties that can be edited. In addition, you can add only attachments that are 50 MB or less in size.
- Security access and restrictions within the case, incident, and party detail panels are controlled from within the user interface definitions.

Subscriptions and Notifications

Introduction to Subscriptions and Notifications

Several features in SAS Enterprise Case Management can be used to send notifications that are similar to Microsoft Outlook's task reminders.

- Users can subscribe to entities.
- Users can generate case reports offline and be notified whether these reports were generated successfully.
- Users can set reminders for tasks in the Task List component.

By default, SAS Enterprise Case Management is installed with templates. The templates reside on the content server under the path **`/sascontent/System/Applications/Templates/notification/en`**.

A utility included with the SAS installation can be found under ***`SAS-configuration-directory`***`/Lev<num>/Web/Utilities/DAVTree.bat`. You can browse the content server by using this utility and opening the following URL: **`http://localhost:8080/SASContentServer/repository/default`**. Use your SAS Administrator user name and password to log on.

You can modify the templates in place or add templates for additional languages using the DAVTree tool. To modify a template, right-click it and select **Edit**. After you have finished with your modifications, click **Save**. Since templates are cached in the content

server, restart SASServer1 and SASServer8 when you have finished modifying the template. Keep in mind that template names must remain the same.

The following sections outline the features that can send notifications and describe the templates.

Note: Special characters such as < > and & will be encoded in plain text and SMS notifications.

Event Alert Notifications

From the Search pop-up menu on the case, incident, party, report, or e-file and the menu bar on the case, incident, party, report, or e-file itself, a user has the option to subscribe to the entity to be alerted of changes. Any saved change to the entity triggers an alert notification using one of three templates. The success template names for case notifications are as follows:

- SAS_Solutions_ECM_Subscriber_Case.st
- SAS_Solutions_ECM_Subscriber_Case_Attachment.st
- SAS_Solutions_ECM_Subscriber_Case_Comment.st
- SAS_Solutions_ECM_Subscriber_Case_Link.st
- SAS_Solutions_ECM_Subscriber_Case_Reassign.st
- SAS_Solutions_ECM_Subscriber_Case_Save.st
- SAS_Solutions_ECM_Subscriber_Case_Unlock.st
- SAS_Solutions_ECM_Subscriber_Case_Workflow.st

The following template properties can be used:

Property	Value
$CASE_ID$	<CASE_ID>
$CASE_LINK$	A link to the case that generated the alert.
$CASE_DETAILS$	A short description of the event.

The success template names for incident notifications are as follows:

- SAS_Solutions_ECM_Subscriber_Incident.st
- SAS_Solutions_ECM_Subscriber_Incident_Attachment.st
- SAS_Solutions_ECM_Subscriber_Incident_Comment.st
- SAS_Solutions_ECM_Subscriber_Incident_Link.st
- SAS_Solutions_ECM_Subscriber_Incident_Save.st

The following template properties can be used:

Property	Value
$INCIDENT_ID$	<INCIDENT_ID>

Property	Value
$INCIDENT_LINK$	A link to the incident that generated the alert.
$INCIDENT_DETAILS$	A short description of the event.

The success template names for party or subject notifications are as follows:

- SAS_Solutions_ECM_Subscriber_Subject.st
- SAS_Solutions_ECM_Subscriber_Subject_Attachment.st
- SAS_Solutions_ECM_Subscriber_Subject_Comment.st
- SAS_Solutions_ECM_Subscriber_Subject_Link.st
- SAS_Solutions_ECM_Subscriber_Subject_Save.st
- SAS_Solutions_ECM_Subscriber_Subject_Unlock.st

The following template properties can be used:

Property	Value
$SUBJECT_ID$	<SUBJECT_ID>
$SUBJECT_LINK$	A link to the subject that generated the alert.
$SUBJECT_DETAILS$	A short description of the event.

The success template names for e-file notifications are as follows:

- SAS_Solutions_ECM_Subscriber_EFile_Attachment.st
- SAS_Solutions_ECM_Subscriber_EFile_Comment.st
- SAS_Solutions_ECM_Subscriber_EFile_Link.st
- SAS_Solutions_ECM_Subscriber_EFile_Save.st
- SAS_Solutions_ECM_Subscriber_EFile_Unlock.st

The following template properties can be used:

Property	Value
$EFILE_ID$	<EFILE_ID>
$EFILE_LINK$	A link to the e-file that generated the alert.
$EFILE_DETAILS$	A short description of the event.

The success template names for report notifications are as follows:

- SAS_Solutions_ECM_Subscriber_Report_Attachment.st
- SAS_Solutions_ECM_Subscriber_Report_Comment.st
- SAS_Solutions_ECM_Subscriber_Report_Link.st

- SAS_Solutions_ECM_Subscriber_Report_Save.st
- SAS_Solutions_ECM_Subscriber_Report_Unlock.st
- SAS_Solutions_ECM_Subscriber_Report_Workflow.st

The following template properties can be modified:

Property	Value
RR_ID	<RR_ID>
RR_LINK	A link to the report that generated the alert.
$RR_DETAILS$	A short description of the event.

Case Report Notifications

From a case's **Print** menu, a user has the option to generate a case report offline. This action triggers one of two notifications: either the case report was generated successfully, or it failed. The success template name is SAS_Solutions_ECM_Subscriber_CaseReport.st.

The failure template name is SAS_Solutions_ECM_Subscriber_CaseReport_Error.st

The following template properties can be modified:

Property	Value
$CASE_ID$	<CASE_ID>
$REPORT_LINK$	A link to the Generated Reports page for that user. Currently this property is supported only in the success template.

Task List Notifications

The Case and Subject UI definitions can be configured to use the Task List component, which allows users to set reminders for tasks. The template name for reminders is SAS_Solutions_ECM_ToDo_Reminder.st

The following template properties can be used within the e-mail to insert pertinent information:

Property	Value
OBJ	ENTITY-<*ENTITY_ID*>
$TASK	Task Description
DUE_DATE	Task Due Date

Property	Value
OBJ_LINK	A link back to the entity. Currently this property is supported only in the HTML template.

Adjusting the Reminder Interval for the Task List

SAS Enterprise Case Management is installed with a metadata property for the reminder timer interval that is set to 15 minutes by default. This means the reminder scheduler will wake up every 15 minutes to determine whether there are any alerts or reminders that need to be sent. This property is also configurable, but it is recommended to remain at 15-minute intervals. If you decide to modify this property, it is also recommended to set this property in multiples of 15, preferably 15, 30, or 60. Keep in mind that the Task List only allows for reminder times at a minimum of 15 minutes. Therefore, setting this interval at odd numbers will cause any alerts or reminders to be sent later than expected. You can find the **reminder.scheduler.interval.time** setting in SAS Management Console under the **Advanced** tab of the SAS Enterprise Case Management Application Management section.

Notifications and the SAS Information Delivery Portal

Configuring the SAS Information Delivery Portal for SAS Enterprise Case Management

To log on to the portal, enter `http://yourmachine/SASPortal` in the browser address bar. Use your configured user name and password to log on the same way you would for SAS Enterprise Case Management. To add the Alerts portlet to your portal, complete the following steps:

1. Click **Customize** on the top banner, and select **Add Page** if you do not already have a page specified.
2. Fill in the required name information and any other optional information. Click **Add**, and then click **Done**. You will now be in your new Tab area with the name that you selected.
3. Click **Customize** and select **Edit Page Content**.
4. Click **Add Portlets**.
5. Select the **Shared Alerts** portlet type if not already selected. Enter the required name and any other information. Click **Add**, and then click **Done**. Click **OK**. You will now see your Alerts portlet within your newly created Tab page.

What to Expect from the Portal

SAS Enterprise Case Management alerts are displayed in this portlet with the task name and the date that the alert was sent. You can clear any of the alert messages that you want to remove. Click a task name to open the case that the alert was created from.

Controlling Alert Notifications from the SAS Preferences Manager

The SAS Preferences Manager is a web application that provides a central facility for SAS application users to manage their preferences and settings. To open the SAS Preferences Manager, click **Options** ⇨ **Preferences** at the top right corner of the application window. Use the Portal section to configure options for notifications of both e-mail and alerts. By default, alert notifications will be set to Portlet only. To receive e-mail, you must select an option that includes e-mail alerts, and the user that you created in metadata must have a valid e-mail address.

For more information about using the SAS Preferences Manager, see the *SAS Intelligence Platform: Web Application Administration Guide*, available at **`http://support.sas.com`**.

Configuring the Web Service

By default, the SAS web service is configured to accept stored process results only when the SAS stored process is completed with no errors (for example, completion code=0). To allow warnings in SAS code, the BI web service for Java 9.4 properties should be changed as follows:

1. Use the **Plug-ins** tab of SAS Management Console to navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
2. Right-click **BI Web Service for Java 9.4** and select **Properties**. Click the **Settings** tab.
3. On the left pane, select **Application** ⇨ **General Configuration**.
4. On the right pane, enter *0-4* for **Acceptable SYSCC list**. Click **OK**.
5. You can now exit SAS Management Console. Restart the web application servers on the middle-tier machine.

For more information about this setting, see "Configuring SAS BI Web Services for Java" in the *SAS Intelligence Platform: Middle-Tier Administration Guide*, available at **`http://support.sas.com/documentation/onlinedoc/intellplatform`**.

Deploy SAS Spelling Correction

SAS Spelling Correction provides the spell-checking capability in SAS Enterprise Case Management. It includes basic spelling correction functionality in which misspelled words are identified and possible corrections are offered. No grammatical suggestions are made. To take advantage of this functionality, perform the following installation instructions:

Installation

1. Locate the spelling-server.zip file in the **`third_party`** folder of the SAS Software Depot, found specifically in the **`SAS_ Spelling_Correction \1_2\Portable_Entities`** directory.

   Note: For the best performance, the SAS Spelling Correction should be co-located on the mid-tier.

2. Extract the SAS Spelling Correction files to one of the following directories:

Windows (32– and 64–bit)	**`$:\Program Files\SASSpellingServer`**
Linux (32– and 64–bit)	**`/opt/SASSpellingServer`**
SAX	**`/opt/SASSpellingServer`**
S64	**`/opt/SASSpellingServer`**
HP-UX (ia64)	**`/opt/SASSpellingServer`**
AIX64/R64	**`/opt/SASSpellingServer`**

The structure of the extracted volume is as follows:

Windows (32–bit)	**`$:\Program Files\SASSpellingServer \TeragramSpellingServer\bin \win32_vc6\_spelling_server.exe`**
Windows (64–bit)	**`$:\Program Files\SASSpellingServer \TeragramSpellingServer\bin \win64_icl_mt \_spelling_server.exe`**
Linux (32–bit)	**`/opt/SASSpellingServer/ TeragramSpellingServer/bin/linux32/ _spelling_server`**
Linux (64–bit)	**`/opt/SASSpellingServer/ TeragramSpellingServer/bin/linux64/ _spelling_server`**
SAX	**`/opt/SASSpellingServer/ TeragramSpellingServer/bin/sunos_x86_64/ _spelling_server`**
S64	**`/opt/SASSpellingServer/ TeragramSpellingServer/bin/sunos64/ _spelling_server`**

HP-UX (ia64)	**/opt/SASSpellingServer/TeragramSpellingServer/bin/hpux_ia64/_spelling_server**
AIX64 / R64	**/opt/SASSpellingServer/TeragramSpellingServer/bin/aix64/_spelling_server**

3. Run the following:

Note: SAS Spelling Correction must be invoked from the **spelling-server** directory. The script is a simple command line program. It does not spawn a separate server process. The script continues to run until you manually stop it by pressing CTRL+C.

Windows (32–bit)	In the **$:\Program Files\SASSpellingServer\TeragramSpellingServer** directory, run the following command: **bin\win32_vc6_spelling_server.exe --port 9000 --spelling-config data\spelling.config --http-admin-port 8123**.
Windows (64–bit)	In the **$:\Program Files\SASSpellingServer\TeragramSpellingServer** directory, run the following command: **bin\win64_icl_mt_spelling_server.exe --port 9000 --spelling-config data\spelling.config --http-admin-port 8123**.
Linux (32–bit)	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/linux32/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.
Linux (64–bit)	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/linux64/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.
SAX	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/sunos_x86_64/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.
S64	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/sunos64/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.
HP-UX (ia64)	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/hpux_ia64/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.
AIX64 / R64	In the **/opt/SASSpellingServer/TeragramSpellingServer** directory, run the following command: **bin/aix64/_spelling_server --port 9000 --spelling-config data/spelling.config --http-admin-port 8123**.

Note: Ports 9000 and 8123 are used for illustration.

Configuration of SAS Spelling Correction

1. Open SAS Management Console.
2. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
3. Right-click **Ent Case Mgmt Mid-Tier 6.2** and select **Properties**.
4. On the **Advanced** tab, enter your environment's property values for ecm.spellChecker.host and ecm.spellChecker.port. Set the host and port based on where the script in step 3 on page 58 of the installation steps was run and what port number was chosen.

   Note: **Property Name** is case-sensitive.

Display 4.4 *Ent Case Mgmt Mid-Tier 6.2 Properties*

Ent Case Mgmt Mid-Tier 6.2 Properties

General | Internal Connection | External Connection | Settings | Advanced | Authorization

Property Name	Property Value
App.ClientSidePoolingAdminID	
DB.Schema	[illegible]_ECM
ECM.Indexing.HttpMethod	POST
ECM.Indexing.Password	
ECM.Indexing.ServerURL	
ECM.Indexing.Username	
ECM.Policy.AutoUnlockOnLogout	true
ECM.Policy.AutoUnlockStrategy	session
ECM.Search.Enabled	true
ECM.Search.URL	http://[illegible].sas.com:8983/solr/EntCaseMgmtCollection
ECM.ServerTier.App.Name	Ent Case Mgmt Server 6.2
Email.Host	mailhost.fyi.sas.com
Email.Port	25
Logon.Style	corporate
Logon.Target	ECMWelcome
Policy.DisplaySessionTimeoutWarning	true
Reassign.Case.User.Group.Or.Role	Ent Case Mgmt Users
Release.Dav.Subfolder	EntCaseManagement6.2
Root.Dav.Products.Folder	Products/SASEntCaseManagement
Table.Records.Per.Page	20
Version.Class	com.sas.solutions.casemgmt.i18n.AppProperties
WF.Sleep.Time.Millis	1000
WebApp.Actions	com.sas.solutions.casemgmt.i18n.AppResources
WebApp.Resources	com.sas.solutions.casemgmt.i18n.AppResources
ecm.administrator.role.name	
ecm.administrator.userid	
ecm.spellChecker.host	
ecm.spellChecker.port	
reminder.scheduler.interval.time	15

5. Restart SASServer8 or the server on which SAS Enterprise Case Management has been deployed.

Installing and Registering the SAS Spelling Server as a Windows Service

In your Install Depot, the spelling server software is delivered in a ZIP file named spelling-server.zip in **`third_party\SAS_Spelling_Correction\1_2\Portable_Entities`**.

1. Create a permanent directory to hold the ZIP file's contents on the machine where you want to run the SAS Spelling Server (for example **`C:\Program Files\SAS Spelling Server`**).
2. Unzip the complete contents of spelling-server.zip to that directory.
3. As a Windows Administrator, open a DOS prompt and go to the **`TeragramSpellingServer\bin\win32`** directory.
4. From the DOS prompt at that directory, run the following command:

   ```
   _tgwinsvc_wrapper.exe --install SASSpellingCorrection --child
   "C:\Program Files\SASSpellingServer\TeragramSpellingServer\bin\win64_icl_mt\_
   spelling_server .exe" --arg --port --arg 9000 --arg --spelling-config --arg
   data\spelling.config --arg --http-admin-port --arg 8123 --working-directory
   "C:\Program Files\SASSpellingServer\TeragramSpellingServer"
   ```

   Note: This does not print out any information to the DOS prompt, but does display a dialog box saying the service was installed.
5. You can start or stop the service from the Windows Services window, or by using the net start service-name and net stop service-name commands.
6. You can uninstall the service using the command **`_tgwinsvc_wrapper.exe --uninstall SASSpellingCorrection`** from the same directory where you ran the installation command. Replace **`SASSpellingCorrection`** with your name for the service if you specified a different name.

Chapter 5

Customizing SAS Enterprise Case Management

Introduction to Customizing **62**
User Interface Definitions 62
Workflows 64
Reference Tables 65
Customizable Search Panels 66

User-Defined Fields **66**
User-Defined Field Tables 66
Configuring User-Defined Fields in the Database 67
User-Defined Fields: Example 68

User-Defined Generic Data Tables **69**

User Interface Definitions **70**
User Interface Definition Files 70
Updating User Interface Definitions 70
Required User Interface Definition Files 70

Configurations **71**
Case Configurations 71
Incident Configurations 72
Party Configurations 73
Report Configurations 74
E-File Configurations 75

Data Security **76**
Data Security: Record Level 76
Data Security: Field Level 76

Resource Bundles **77**
Custom Resource Bundles 77

Workflows **78**
Defining Workflows 78
Data Objects 78
Data Object Types Supported in SAS Workflow Studio 79
Adding Data Objects 80
Defining Actors 80
Defining Static Actors 80
Dynamically Determined Actors 80
Statuses 80
Synchronizing Workflow Values 81
Versioning 82

Reference Tables **82**

Defining Reference Tables 82
Configuring Reference Tables in the Database 82
Defining Static Reference Tables 83
Adding User-Defined Reference Tables 85
Rules and Conventions for Database Fields and Reference Table Values 86

Search Panels **87**
Search Criteria 87
User-Specified Configurations 87
Searchable Fields 88
Field Labels 89
User Interface Controls 89
Search Filters 90
User-Specified Configurations 90
Filterable Fields 90
Field Labels 91
Search Results 91
User-Specified Configurations 91
Displayable Fields 91
Column Header Labels 92
SAS Metadata Repository Properties 92

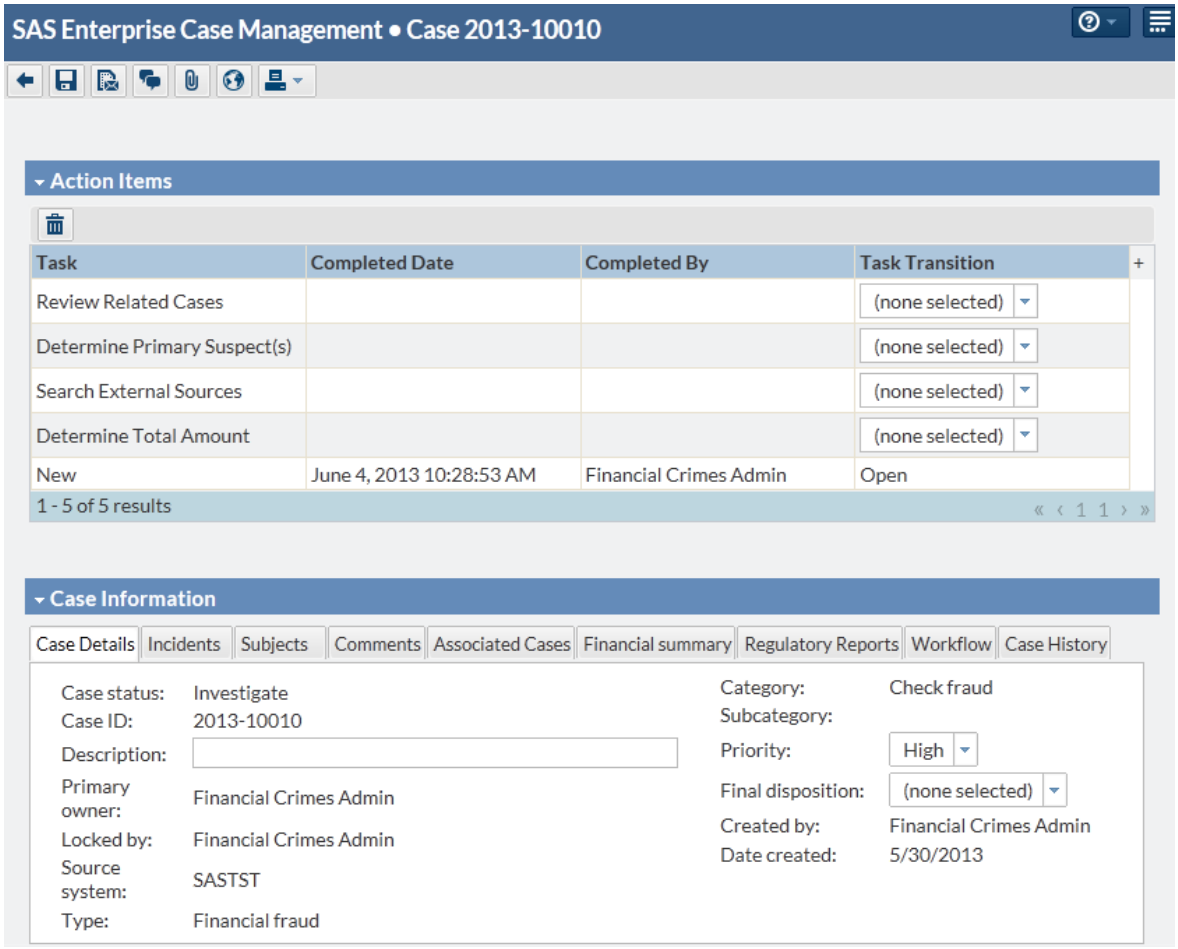
Adding Data to Relationships **93**
Customizing Case and Party Relationships 93
Adding Fields to Relationships 93
Displaying Relationship Data in a UI Definition 94
Designing a Screen to Edit Relationship Data 95

Introduction to Customizing

User Interface Definitions

User interface definitions are used to define customizable windows within the SAS Enterprise Case Management user interface. The following display shows a customizable window that is defined using a user interface definition.

Display 5.1 *Customized Window*



XML is used to describe user interface definitions. The following XML code is defined in the user interface definition used to render the previous example.

```
<section id="caseInfo">
   <label><message key="section.case.information.header.txt" /></label>
   <tab-section id="viewCaseTab">
      <tab id="caseTab">
         <label><message key="tab.case.details.header.txt" /></label>
         <field name="CASE.CASE_ID" type="string" readonly="true">
            <label>
               <message key="field.case.case_id.label.txt" />
            </label>
         </field>
         <field name="CASE.CASE_ID" type="hidden"/>
         <field name="CASE.CASE_DESC" type="textarea" length="40"
          required="false">
            <label>
               <message key="field.case.case_desc.label.txt" />
            </label>
         </field>
         <field name="CASE.INVESTIGATOR_USER_ID" type="string" required="false"
               values="GetUserDisplayName(CASE.INVESTIGATOR_USER_ID)">
            <label>
               <message key="field.case.investigator_user_id.label.txt" />
```

```
   </label>
</field>
<field name="CASE.SOURCE_SYSTEM_CD" type="dropdown" readonly="true"
       values="GetLabelValues('RT_SOURCE_SYSTEM')">
   <label>
      <message key="field.case.source_system_cd.label.txt" />
   </label>
</field>
<field name="CASE.CASE_TYPE_CD" type="dropdown" required="false"
 readonly="true"
       values="GetLabelValues('RT_CASE_TYPE')">
   <label>
      <message key="field.case.case_type_cd.label.txt" />
   </label>
</field>
```

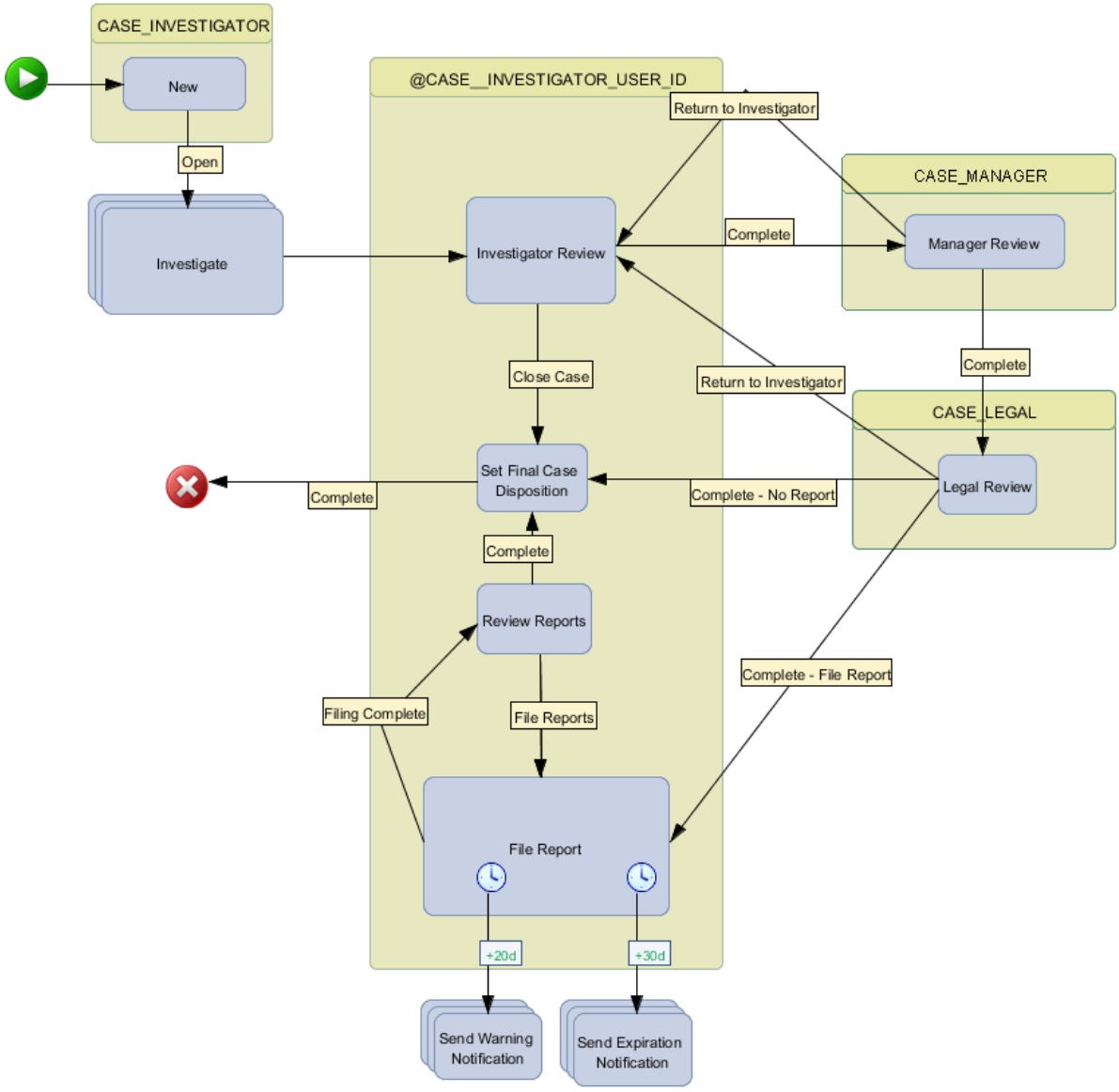
Workflows

Workflows are used to manage the investigative process. Workflow definitions define what activities are involved in the investigative process and which users can perform the activities. The workflow services used within SAS Enterprise Case Management support the following:

- automation of SAS processes
- routing of activities based on events, data, timers, groups, and/or roles
- e-mail notifications
- visual process designer
- concurrent activities
- decision gateways that allow conditional branching

The following display shows an example workflow template.

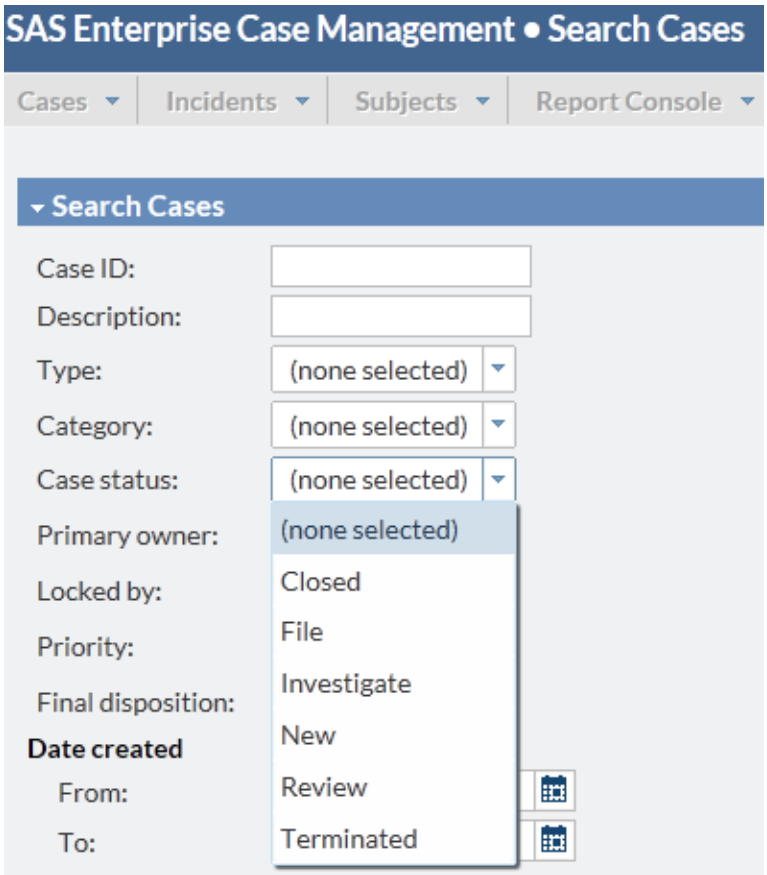
Display 5.2 *Workflow Template*



Reference Tables

Reference tables define the list of possible values for a particular field or selection list. In the following display, the drop-down list values for case status come from a configurable reference table named RT_CASE_STATUS. The coded values and displayable values for each selectable option are specified in the RT_CASE_STATUS reference table. User-defined reference tables can also be defined.

Display 5.3 *Reference Tables*



Customizable Search Panels

The case, incident, party, report, and e-file search panels are fully customizable. Any static or user-defined field can be used as a search criterion. Any static or user-defined field with possible values described using a reference table can be used as a search filter. The search results can display any static or user-defined field that holds a single value.

Note: User-defined fields that are part of a tables of values cannot be used in the search results because the UI table cannot show multiple values for a single field.

For more information on configuring search panels, refer to "Search Panels" on page 87.

User-Defined Fields

User-Defined Field Tables

There are seven types of data objects in SAS Enterprise Case Management. They are case, incident, party (also called subject), report, e-file, financial item, and generic data. The user-defined fields of each data object type are defined in the table <*data_object_type*>_UDF_DEF. Each row in the <*data_object_type*>_UDF_DEF table

represents a user-defined field definition for the data object. To define user-defined fields, add the appropriate data in these tables.

Configuring User-Defined Fields in the Database

The structure for all seven tables (CASE_UDF_DEF, INCIDENT_UDF_DEF, PARTY_UDF_DEF, RR_UDF_DEF, EFILE_UDF_DEF, FINANCIAL_ITEM_UDF_DEF, and GENERIC_DATA_UDF_DEF) is identical. Each contains the following columns:

- UDF_TABLE_NM
- UDF_NM
- UDF_TYPE_NM
- UDF_DESC
- MAX_CHAR_CNT

User-defined field names must have the following characteristics:

- The length must be 3 to 30 characters.
- The first two characters must be “X_”.
- The characters following “X_” can be any combination of uppercase letters, numbers, and underscores.

UDF_TABLE_NM
: contains the name of the user-defined field’s table. If a user-defined field contains one value, then this name will be the same as the data object name CASE, INCIDENT, PARTY, RR, or EFILE. UDF_TABLE_NM and UDF_NM together make up the unique key for a user-defined field. If a user-defined field can have more than one value, then this name must have the following characteristics:

  - The length must be 3 to 30 characters.
  - The first two characters must be “X_” .
  - The characters following “X_” can be any combination of uppercase letters, numbers, and underscores.
  - The name must be unique with respect to all other static and user-defined table names.

UDF_NM
: contains the name of the user-defined field. UDF_TABLE_NM and UDF_NM together make up the unique key for a user-defined field. User-defined field names must have the following characteristics:

  - The length must be 3 to 30 characters.
  - The first two characters must be “X_”.
  - The characters following “X_” can be any combination of uppercase letters, numbers, and underscores.

UDF_TYPE_NM
: contains the data type name for the user-defined field.

Table 5.1 *User-Defined Field Data Types*

Data Type	Description	Java Type
VARCHAR	Character string	String
BIGINT	Whole number	Long
DOUBLE	Double precision number	Double
BOOLEAN	Boolean (true/false)	Boolean
DATE	Date	Date
TIMESTAMP	Date and time	Timestamp
LNGVARCHAR	Character string more than 4000 characters	CLOB for Oracle; VARCHAR(MAX) for SQL Server; TEXT for PostgresSQL

Note: UDF fields with a type of BIGINT are stored in double precision floating point columns in the database. Therefore, they have only 53 bits of precision, not the full precision of a Long Java type.

UDF_DESC
contains a description of the user-defined field.

MAX_CHAR_CNT
contains the maximum number of characters for user-defined fields with a VARCHAR data type.

If an invalid value is loaded into the UDF_DEF tables, an error message is logged and that field is ignored by the application. Common errors include the following:

- including invalid characters in UDF_TABLE_NM or UDF_NM
- defining a UDF_TABLE_NM or UDF_NM longer than 30 characters
- not including a MAX_CHAR_CNT for a VARCHAR column
- defining a MAX_CHAR_CNT as < 0 or > 1000

User-Defined Fields: Example

In the following example configuration table, each case can have a loss amount specified. Because there is only one value for the loss amount field for each case, UDF_TABLE_NM is CASE. There can be zero or more accounts associated with a case. We also need to store whether each account is closed or not. Therefore, we create a user-defined table called X_ACCOUNT, which contains two user-defined fields: X_ACCOUNT_ID and X_CLOSED_FLG. We also need to track all suspicious activities related to the case; there could be more than one of these activities. We therefore create user-defined table X_SUSPCIOUS_ACTIVITY, which contains one user-defined field: X_SUSPCIOUS_ACTIVITY_CD.

UDF_TABLE_NM	UDF_NM	UDF_TYPE_NM	MAX_CHAR
CASE	X_LOSS_AMT	DOUBLE	
X_ACCOUNT	X_ACCOUNT_ID	VARCHAR	32
X_ACCOUNT	X_CLOSED_FLG	BOOLEAN	
X_SUSPICIOUS_ ACTIVITY	X_SUSPICIOUS_ ACTIVITY_CD	VARCHAR	3

User-Defined Generic Data Tables

Generic data look-up functions can be defined in a custom user interface (UI). Generic data tables are tables that are not directly linked to a particular case, incident, party, report, e-file, or financial item. All data fields in generic data tables are defined in the GENERIC_DATA_UDF_DEF table as user-defined fields, and they follow the same name-value pair structure as in other ECM tables.

One major difference between generic data tables and the other SAS Enterprise Case Management tables is that generic data does not have a live table. That means there is no data table for storing the most current records. To get a list of current data, records in GENERIC_DATA_UDF_<data type>_VALUE should be filtered by empty VALID_TO_DTTM. To verify that the data is loaded properly in generic data tables, see the step for placing a generic data table in a rectangular structure in "Adding Custom SAS Code" on page 217.

To facilitate FINCEN reporting, the sample code for defining financial institution and branch look-up tables is shipped with the solution. It can also be found in loadrr_config_fincen_base.sas in the following locations:

Platform	Path
Windows	*!SASROOT*`\casemgmtmva\sasmisc\sample\config`
UNIX	*!SASROOT*`/misc/casemgmtmva/sample/config`

The sample UI definitions rr-fincen-sardi-01.xml and rr-fincen-ctr-01.xml are examples of how a report form can be defined for SAR and CTR reporting. This file is located in the following locations:

Platform	Path
Windows	*SAS-middle-tier-installation-directory*`\deploy\sample\UIDefinitions`
UNIX	*SAS-middle-tier-installation-directory*`/deploy/sample/UIDefinitions`

Note: The UDF fields X_INSTITUTION_OPEN_DT and X_INSTITUTION_CLOSE_DT of X_INSTITUTION and X_BRANCH_OPEN_DT and X_BRANCH_CLOSE_DT of X_BRANCH are the open and close dates of the institution and branch. These fields are used in rr-fincen-sardi-01.xml and rr-fincen-ctr-01.xml to filter active institutions and branches based on the case creation date.

User Interface Definitions

User Interface Definition Files

User interface definition files specify the form and content of panels presented in SAS Enterprise Case Management, the data that is captured, and how that data is validated. The sample user interface definition files for all panels that make use of the Custom Page Builder are located in one of the following directories, depending on your platform:

- Windows platforms: ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions**
- UNIX platforms: ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions**

Updating User Interface Definitions

Over time, changes will need to be made to user interface definitions. Customers must decide whether they should update a user interface definition or create a new version of the user interface definition. For minor changes that don't cause existing cases, incidents, parties, reports, or e-files to become invalid, it is permissible to update an existing user interface definition. For major changes that might cause existing entities to become invalid, it is recommended that customers create a new version of the user interface definition (by giving it a new unique name). This version is used only for new records. Existing records continue to use the older version of the user interface definition. User interface definitions must be uploaded from the **Administration** tab.

Required User Interface Definition Files

When you create a new case, incident, subject, report, e-file, or financial item in SAS Enterprise Case Management, the system requires some basic information to determine which user interface definition to use to capture the new entity. To provide that information, the following user interface definitions can to be updated:

Windows platforms:

- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newCase-uidef.xml**
- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newIncident-uidef.xml**
- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newParty-uidef.xml**

- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newReport-uidef.xml**
- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newEFile-uidef.xml**
- ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions\newFinancialItem-ui def.xml**

UNIX platforms:

- ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newCase-uidef.xml**
- ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newIncident-uidef.xml**
- ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newParty-uidef.xml**
- ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newReport-uidef.xml**
- ***!SASROOT*/misc/casemgmtmva/install/*SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newEFile-uidef.xml**
- ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions/newFinancialItem-ui def.xml**

Users cannot create any of the entities in the system without these files. The files can be customized. However, it is important to note that none of the fields defined in these files should be removed. An example of a customization is to mark one of the existing fields as required. Another example would be to override the default generated ID for the entity.

These UI definitions should be downloadable from the **Administration** tab, under **UI Definition**. If any of the definition files are missing, they can be found in ***SAS-middle-tier-installation-directory*/deploy/required/UIDefinitions**.

Configurations

Case Configurations

The CASE_CONFIG and CASE_CONFIG_X_USER_GROUP tables are used to store information about how cases are associated with the following:

- user interface definition. This is used to render the case detail panel.
- workflow definition. This is used to create a workflow instance to manage the investigative process for the case.
- case owner. If specified, the case will be initially assigned to the specified user. Otherwise, the case will be initially unassigned.
- user groups. Each row in the CASE_CONFIG_X_USER_GROUP table specifies one user group that is associated with a case configuration. Any user in any of the associated user groups will have access to the case.

The CASE_CONFIG table must be configured to handle every possible type of case created in SAS Enterprise Case Management at the customer site. The CASE_CONFIG table contains the following columns:

CASE_CONFIG_SEQ_NO
: contains the sequence number of the case configuration. Case configurations are processed in order until a matching configuration is found.

CASE_TYPE_CD, CASE_CATEGORY_CD and CASE_SUBCATEGORY_CD
: are used to determine whether the newly created case matches this configuration. CASE_TYPE_CD is required. CASE_CATEGORY_CD and CASE_SUBCATEGORY_CD can be null. If null, these columns are not factored in when determining whether the newly created case matches this configuration.

UI_DEF_FILE_NM
: contains the filename of the user interface definition if the newly created case matches this configuration.

INVESTIGATE_WORKFLOW_DEF_NM
: contains the name of the investigation workflow definition if the newly created case matches this configuration.

REOPEN_WORKFLOW_DEF_NM
: is not used in this release of SAS Enterprise Case Management and should be left null.

INVESTIGATOR_USER_ID
: is the user ID of the user whom the case is initially assigned to. If null, the case will be initially unassigned.

The CASE_CONFIG_X_USER_GROUP table contains the following columns:

CASE_CONFIG_SEQ_NO
: contains the sequence number of the case configuration. This column maps to the corresponding configuration in the CASE_CONFIG table.

USER_GROUP_NM
: contains the name of a user group defined in the SAS Metadata Repository that has access to newly created cases that match this configuration. A user has to be in only one of the user groups from one of the CASE_CONFIG_X_USER_GROUP rows to access a case.

If a new case type, category, or subcategory is added to the system, then the reference tables must be updated so that items are available in the drop-down lists when creating a new case. If a new case type is added, then a corresponding reference table value should be added to RT_CASE_TYPE. If a new case category is added, then a corresponding reference table value should be added to RT_CASE_CATEGORY. If a new case subcategory is added, then a corresponding reference table value should be added to RT_CASE_SUBCATEGORY.

Incident Configurations

The INCIDENT_CONFIG and INCIDENT_CONFIG_X_USER_GROUP tables are used to store information about how incidents are associated with the following:

- user interface definition. This is used to render the incident detail panel.
- user groups. One or more user groups can be associated with an incident. Any user in the associated user groups will have access to the incident.

The INCIDENT_CONFIG table must be configured to handle every possible type of incident created in SAS Enterprise Case Management. The INCIDENT_CONFIG table contains the following columns:

INCIDENT_CONFIG_SEQ_NO
: contains the sequence number of the incident configuration. Incident configurations are processed in order until a matching configuration is found.

INCIDENT_TYPE_CD, INCIDENT_CATEGORY_CD and INCIDENT_SUBCATEGORY_CD
: are used to determine whether the newly created incident matches this configuration. INCIDENT_TYPE_CD is required. INCIDENT_CATEGORY_CD and INCIDENT_SUBCATEGORY_CD can be null. If null, these columns are not factored in when determining whether the newly created incident matches this configuration.

UI_DEF_FILE_NM
: contains the filename of the user interface definition, if the newly created incident matches this configuration.

The INCIDENT_CONFIG_X_USER_GROUP table contains the following columns:

INCIDENT_CONFIG_SEQ_NO
: contains the sequence number of the incident configuration. This column maps to the corresponding configuration in the INCIDENT_CONFIG table.

USER_GROUP_NM
: contains the name of a user group defined in the SAS Metadata Repository that has access to newly created incidents that match this configuration.

If a new incident type, category, or subcategory is added to the system, then the reference tables must be updated so that items are available in the drop-down lists when creating a new incident. If a new incident type is added, then a corresponding reference table value should be added to RT_INCIDENT_TYPE. If a new incident category is added, then a corresponding reference table value should be added to RT_INCIDENT_CATEGORY. If a new incident subcategory is added, then a corresponding reference table value should be added to RT_INCIDENT_SUBCATEGORY.

Party Configurations

The PARTY_CONFIG and PARTY_CONFIG_X_USER_GROUP tables are used to store information about how parties are associated with the following:

- user interface definition. This is used to render the party detail panel.
- user groups. One or more user groups can be associated with a party. Any user in the associated user groups will have access to the party.

The PARTY_CONFIG table must be configured to handle every possible type of party created in SAS Enterprise Case Management. The PARTY_CONFIG table contains the following columns:

PARTY_CONFIG_SEQ_NO
: contains the sequence number of the party configuration. Party configurations are processed in order until a matching configuration is found.

PARTY_TYPE_CD, PARTY_CATEGORY_CD and PARTY_SUBCATEGORY_CD
: are used to determine whether the newly created party matches this configuration. PARTY_TYPE_CD is required. PARTY_CATEGORY_CD and

PARTY_SUBCATEGORY_CD can be null. If null, these columns are not factored in when determining whether the newly created party matches this configuration.

UI_DEF_FILE_NM
contains the filename of the user interface definition if the newly created party matches this configuration.

The PARTY_CONFIG_X_USER_GROUP table contains the following columns:

PARTY_CONFIG_SEQ_NO
contains the sequence number of the party configuration. This column maps to the corresponding configuration in the PARTY_CONFIG table.

USER_GROUP_NM
contains the name of a user group defined in the SAS Metadata Repository that has access to newly created parties that match this configuration.

If a new party type, category, or subcategory is added to the system, then the reference tables must be updated so that items are available in the drop-down lists when creating a new party. If a new party type is added, then a corresponding reference table value should be added to RT_PARTY_TYPE. If a new party category is added, then a corresponding reference table value should be added to RT_PARTY_CATEGORY. If a new party subcategory is added, then a corresponding reference table value should be added to RT_PARTY_SUBCATEGORY.

Report Configurations

The RR_CONFIG and RR_CONFIG_X_USER_GROUP tables are used to store information about how reports are associated with the following:

- user interface definition. This is used to render the report detail panel.
- workflow definition. This is used to create a workflow instance to manage the investigative process for the report.
- user groups. One or more user groups can be associated with a report. Any user in the associated user groups has access to the report.

The RR_CONFIG table must be configured to handle every possible type of report created in SAS Enterprise Case Management. The RR_CONFIG table contains the following columns:

RR_CONFIG_SEQ_NO
contains the sequence number of the report configuration. Report configurations are processed in order until a matching configuration is found.

RR_TYPE_CD, RR_CATEGORY_CD and RR_SUBCATEGORY_CD
are used to determine whether the newly created report matches this configuration. RR_TYPE_CD is required. RR_CATEGORY_CD and RR_SUBCATEGORY_CD can be null. If null, these columns are not factored in when determining whether the newly created report matches this configuration.

UI_DEF_FILE_NM
contains the filename of the user interface definition if the newly created report matches this configuration.

FORM_AGENCY_CD and FORM_TYPE_CD
are the identifiers of a regulatory form defined in FORM_CONFIG. It is used to retrieve the most current government form.

EFILE_CONFIG_SEQ_NO
: is the identifier of an e-file configuration record in EFILE_CONFIG. It is used to define which e-file type, category, and subcategory combination should be used for filing the report.

WORKFLOW_DEF_NM
: contains the name of the workflow definition if the newly created report matches this configuration.

The RR_CONFIG_X_USER_GROUP table contains the following columns:

RR_CONFIG_SEQ_NO
: contains the sequence number of the report configuration. This column maps to the corresponding configuration in the RR_CONFIG table.

USER_GROUP_NM
: contains the name of a user group defined in the SAS Metadata Repository that has access to newly created reports that match this configuration.

If a new report type, category, or subcategory is added to the system, then the reference tables must be updated so that items are available in the drop-down lists when creating a new report. If a new report type is added, then a corresponding reference table value should be added to RT_RR_TYPE. If a new report category is added, then a corresponding reference table value should be added to RT_RR_CATEGORY. If a new report subcategory is added, then a corresponding reference table value should be added to RT_RR_SUBCATEGORY.

E-File Configurations

The EFILE_CONFIG and EFILE_CONFIG_X_USER_GROUP tables are used to store information about how e-files are associated with the following:

- user interface definition. This is used to render the e-file detail panel.
- user groups. One or more user groups can be associated with an e-file. Any user in the associated user groups have access to the e-file.

The EFILE_CONFIG table must be configured to handle every possible type of e-file created in SAS Enterprise Case Management. The EFILE_CONFIG table contains the following columns:

EFILE_CONFIG_SEQ_NO
: contains the sequence number of the e-file configuration. E-file configurations are processed in order until a matching configuration is found.

EFILE_TYPE_CD, EFILE_CATEGORY_CD and EFILE_SUBCATEGORY_CD
: are used to determine whether the newly created e-file matches this configuration. EFILE_TYPE_CD is required. EFILE_CATEGORY_CD and EFILE_SUBCATEGORY_CD can be null. If null, these columns are not factored in when determining whether the newly created e-file matches this configuration.

UI_DEF_FILE_NM
: contains the filename of the user interface definition if the newly created e-file matches this configuration.

FORM_AGENCY_CD and FORM_TYPE_CD
: are the identifiers of a regulatory form defined in FORM_CONFIG. It is used to retrieve the most current government form.

The EFILE_CONFIG_X_USER_GROUP table contains the following columns:

EFILE_CONFIG_SEQ_NO
: contains the sequence number of the e-file configuration. This column maps to the corresponding configuration in the EFILE_CONFIG table.

USER_GROUP_NM
: contains the name of a user group defined in the SAS Metadata Repository that has access to newly created e-file reports that match this configuration.

If a new e-file type, category, or subcategory is added to the system, then the reference tables must be updated so that items are available in the drop-down lists when creating a new report. If a new report type is added, then a corresponding reference table value should be added to RT_EFILE_TYPE. If a new report category is added, then a corresponding reference table value should be added to RT_EFILE_CATEGORY. If a new report subcategory is added, then a corresponding reference table value should be added to RT_EFILE_SUBCATEGORY.

Data Security

Data Security: Record Level

The initial user groups that have access to a case are determined and configured during case creation. The initial user groups are stored in the CASE_X_USER_GROUP table, which associates a case to one or more user groups defined in the SAS Metadata Repository. Any user in the associated groups, or the owner of the case, has access to the case. This means that editing the case is allowed unless the Edit Case capability was not granted to the user. Similarly INCIDENT_X_USER_GROUP, PARTY_X_USER_GROUP, RR_X_USER_GROUP, and EFILE_X_USER_GROUP store the incident, party, report, and e-file user group association.

If a user does not have access to an entity, the entity is not visible to the user in the system unless the user is looking at the entity within the context of its parents. For example, if a user is in group A, that user has access to only A-type cases and incidents. However, if there is an A-type case with a B-type incident, then the user is able to view, but not edit, the B-type incident if the user is assigned to that case.

If the type, category, or subcategory of an entity is changed, the permissions for the entity are redetermined and stored in <*ENTITY*>_X_USER_GROUP. Therefore, the permissions for an entity can change after creation.

Currently, there is not a way to modify user groups associated with an entity within SAS Enterprise Case Management without changing the type, category, and subcategory of the entity. This can be done only by an administrator who can modify the appropriate database tables directly.

Data Security: Field Level

Field-level security within the entity detail panels is controlled by the user interface definitions. Field-level security on the entity search panels is controlled by the search panel configuration, which is performed at the user level.

Resource Bundles

Custom Resource Bundles

Custom resource bundles are necessary for creating labels for user-defined fields in the customizable data model. When a user-defined field is created, the field can be referenced in the user interface definition file. A label can also be shown for that field. The label tag in the user interface definition has a message tag with a key attribute. The key is a reference to a property in the resource bundle.

For example, X_BRANCH_ID has been defined as a user-defined field for a case. To define a label for the field, the following entry is made in the custom resource bundle file:

```
field.case.x_branch_id.label.txt=Branch ID:
```

To reference this property, the following can be added to the user interface definition file:

```
<label><message key="field.case.x_branch_id.label.txt" /></label>
```

Customers can also override resource bundle properties defined in SAS Enterprise Case Management using the custom resource bundle. For example, you can change the label for a party full name as follows:

- `field.party.party_full_nm.label.txt=Person/Organization name:`
- `field.party.party_full_nm.header.txt=Person/Organization Name`

Note: The property ending in **`label.txt`** is used for text next to input fields. The property ending in **`header.txt`** is used as the heading of a column when fields are used in a table.

The following files contain all of the properties used in SAS Enterprise Case Management:

- com.sas.solutions.casemgmt.i18n.AppResources.properties
- com.sas.solutions.casemgmt.i18n.Actions.properties
- com.sas.solutions.casemgmt.i18n.WebServiceResources.properties
- com.sas.solutions.cpb.i18n.CPBResources.properties

The properties in all of these files can be overridden, except for the actions resource bundle (com.sas.solutions.casemgmt.i18n.Actions.properties.) If customers want to override a property in this file, they have to modify the actual file in the SAS Enterprise Case Management JAR file.

The naming convention for the custom resource bundle files is as follows:

- custom.properties
- custom _<locale>.properties

For example, a file for the Canada-French locale would be named custom_ca_FR.properties.

Note: When a table is empty, the following message is displayed: **No results found.** You can customize this message by changing the value of the table.default.no.rows.message.txt property as follows:

```
table.default.no.rows.message.txt = <Empty table message.>
```

Workflows

Defining Workflows

Each case within SAS Enterprise Case Management can be associated with a workflow instance (also known as a process instance) that is used to manage the investigative process. Workflow templates (also known as process templates) need to be defined using SAS Workflow Studio before workflow instances can be created for a case. See the *SAS Workflow Studio: User's Guide* for step-by-step instructions on defining workflow definitions.

Data Objects

The following root process-level data object is required in all workflow definitions for use in SAS Enterprise Case Management:

CASE__CASE_RK

- Set Type = Short Text
- Set Value = 0
- When the workflow instance is created, the case key of the associated case is set as the value for this data object.

The following optional root process-level data object can be used to automatically open the case when it is first edited in SAS Enterprise Case Management:

AUTO_OPEN_STATUS

- Set Type = Short Text
- Set Value = the name of the status to automatically apply when the case is first edited

If other static or user-defined case fields are needed within the workflow definition as input to decision gateways and policies, they can be added as root process-level data objects using the following naming convention:

```
<TableName>__<FieldName>
```

Here are some examples:

- CASE__REGULATORY_RPT_RQD_FLG (static case field)
- CASE__X_LOSS_AMT (single-valued user-defined case field)
- X_SUSPICIOUS_ACTIVITY__X_SUSPICIOUS_ACTIVITY_CD (multi-valued user-defined case field)

The following table describes how data object values are set for the different case field data types:

Table 5.2 Data Object Values

Data Type	Data Object Value Description
VARCHAR	Raw value
BIGINT	Numerical value
DOUBLE	Numerical value
BOOLEAN	String value – **`true`** or **`false`**
DATE	Formatted string value – **`yyyy.MM.dd`**
TIMESTAMP	Formatted string value – **`yyyy.MM.dd HH:mm:ss`**
Multi-valued field	Comma delimited string of the above formatted values

All root process-level data objects that map to case fields (static and user-defined) in the SAS Enterprise Case Management database are set to the current value in the database every time the case is saved. Incident fields and party fields cannot be used as data objects even though they might be associated with a case.

The following optional data object adds a delay between when an activity is started (such as saving a case or report) and when the UI refreshes.

SLEEP_TIME_MILLIS

- Set Type = Number
- Set Value = the length of the delay, in milliseconds

This data object can be added to both activities and to root-level processes. When defined, this value overrides the application-wide default workflow transition wait time. SAS Enterprise Case Management waits the given number of milliseconds after any case or report save. Any save to a case or report can trigger one or more policies to execute, and those policies might update the case or report. To guarantee that all policies are completed before the user edits the object again, this value might need to be tuned for your environment. If this data object is defined with a negative value, it is ignored.

When an activity executes, the workflow checks whether this data object is defined for the activity. If it is, its value is used as the wait time. If this data object is not defined for the activity, then the root process is checked for the data object. If it is defined there, that data object's value is used as the wait time for the activity. If the data object is defined at neither the activity nor root process-level, a wait time is retrieved by looking up the WF.Sleep.Time.Millis property in the metadata. (See "SAS Metadata Repository Properties" on page 92 for more information.)

Data Object Types Supported in SAS Workflow Studio

While designing custom workflow templates, keep in mind that the following list of data object types are no longer used in SAS 9.4:

- Check Box
- Database Object

- File
- Organizational Role
- Picklist
- User

The following data objects are supported in SAS Workflow Studio in SAS 9.4:

- Date
- E-mail
- Number
- Short Text
- Long Text
- URL
- XML Object
- ItemList

For more information about data object types, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Adding Data Objects

For information about adding data objects, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Defining Actors

For information about defining actors, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Defining Static Actors

For information about defining static actors, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Dynamically Determined Actors

For information about dynamically determined actors, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Statuses

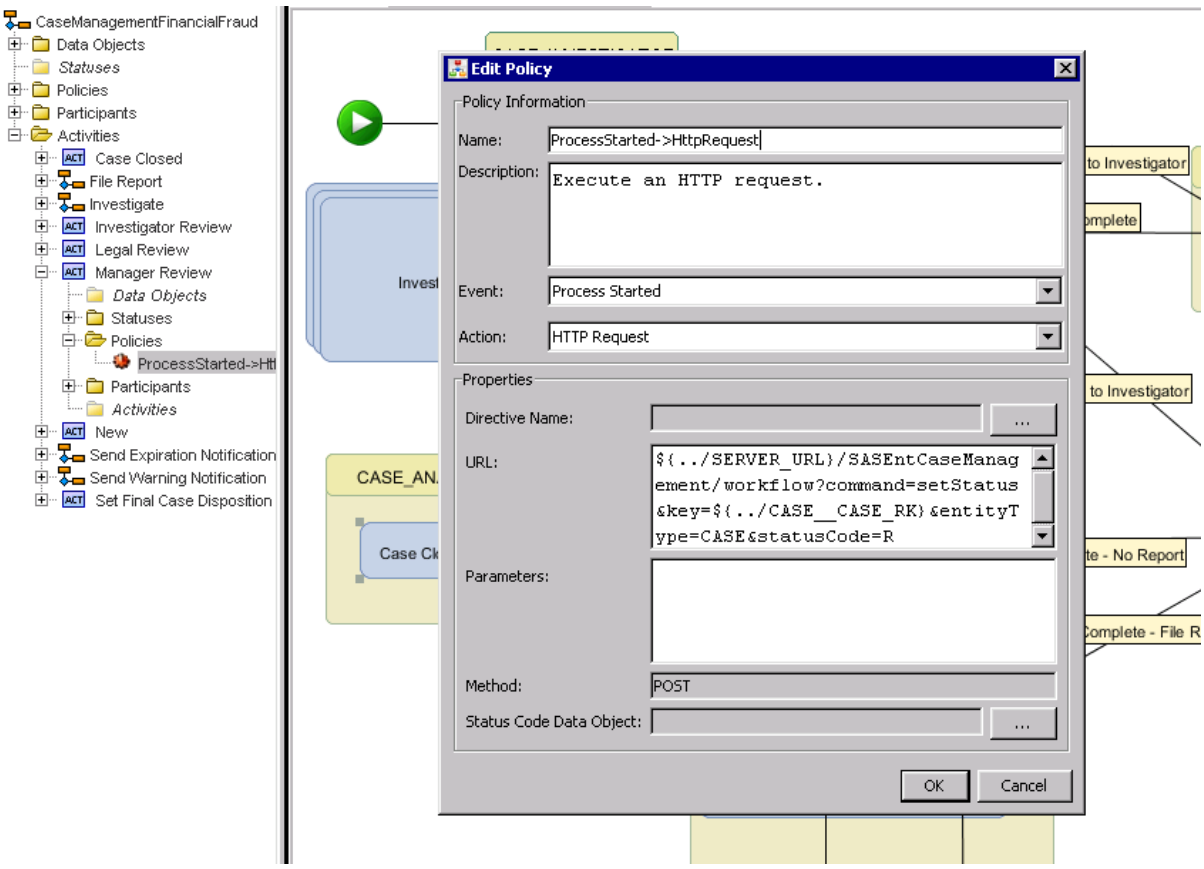
For information about statuses, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Synchronizing Workflow Values

The workflow definition can be configured to notify SAS Enterprise Case Management when certain events happen within the workflow. This ensures that the solution state is synchronized with the workflow. This is accomplished by using workflow policies, specifically the HttpRequest policy. This policy can be configured to invoke an HTTP URL in SAS Enterprise Case Management to send notifications of relevant data changes, such as status and operand values.

To set up a synchronization notification, go to the relevant triggering activity and right-click **Policies** to add a new policy. You can also edit existing policies from here, as shown in the following display.

Display 5.4 *Edit Policy*



In the **Event** field, specify the event that causes the policy to execute (for example, ProcessStarted, ProcessFinished). In the **Action** field, select **HttpRequest**. The following notifications are supported:

Set case status

HTTP URL:

```
/SASEntCaseManagement/workflow?command=setStatus&key=
${../CASE__CASE_RK}&statusCode=<caseStatusCode>
```

The CASE_STATUS_CD database field is set to the value of the statusCode request parameter shown as <caseStatusCode> in the preceding URL. The status code value should be defined in the RT_CASE_STATUS static reference table.

Set case closed

HTTP URL:

```
/SASEntCaseManagement/workflow?command=setStatus&key=
${../CASE__CASE_RK}&statusCode=<caseStatusCode>&caseClosed=true
```

The CASE_STATUS_CD database field is set to the value of the statusCode request parameter shown as <caseStatusCode> in the preceding URL. The status code value should be defined in the RT_CASE_STATUS static reference table.

The CLOSE_DTTM database field is set to the current date and time if the caseClosed request parameter is true. Otherwise, this field is set to null.

Set case opened

HTTP URL:

```
/SASEntCaseManagement/workflow?command=setOpened&key=
${../CASE__CASE_RK}
```

The OPEN_DTTM database field is set to the current date and time.

Set case reopened

HTTP URL:

```
/SASEntCaseManagement/workflow?command=setReopened&key=
${../CASE__CASE_RK}
```

The REOPEN_DTTM case database field is set to the current date and time.

HTTP Method should always equal **POST. HTTP User** and **HTTP Password** should always be left blank. If the host name and port are not included in the HTTP URL, the HTTP request is sent to the SAS Enterprise Case Management web application running on the same server as the workflow services on the SAS platform. For more information, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Versioning

For information about versioning, see the *SAS Workflow Studio: User's Guide* at **http://support.sas.com/documentation/onlinedoc/workflow/**.

Reference Tables

Defining Reference Tables

Reference tables define the list of possible values for a particular field or selection list. Each row in the REF_TABLE_VALUE table represents a possible value for a static or user-defined reference table. To define user-defined reference tables or add possible values for static reference tables, add the appropriate data in this table.

Configuring Reference Tables in the Database

The REF_TABLE_VALUE table contains the following columns:

REF_TABLE_NM
contains the name of the static or user-defined reference table. Reference table names must have the following characteristics:

- The length must be 3 to 30 characters.
- The first three characters must be “RT_” for static reference tables.
- The first two characters must be “X_” for user-defined reference tables.
- The characters following the above prefix can be any combination of upper case letters, numbers, and underscores.
- The name must be unique with respect to all other static and user-defined table names.

VALUE_CD
contains the coded value. REF_TABLE_NM and VALUE_CD together make up the unique key for a reference table possible value.

VALUE_DESC
contains the displayable value.

PARENT_REF_TABLE_NM
optionally contains the name of the parent reference table used for cascading prompts.

PARENT_VALUE_CD
optionally contains the name of the parent coded value used for cascading prompts.

DISPLAY_ORDER_NO
contains a number that determines the display order of the reference table value in the user interface. If two or more reference table values have the same display order, then they will be ordered alphabetically by the VALUE_DESC column.

Defining Static Reference Tables

The following static reference tables must be defined in the REF_TABLE_VALUE table. These reference tables are considered static because the SAS Enterprise Case Management application has hardcoded references to these reference tables. All other reference tables are considered user-defined.

RT_CASE_STATUS
contains case statuses. This reference table is not preloaded during the installation. Customers must add all possible case statuses.

RT_CASE_TYPE
contains case types for case classification. This reference table is not preloaded during the installation. Customers must add all possible case types.

RT_CASE_CATEGORY
contains case categories for case classification. This reference table is not preloaded during the installation. Customers must add all possible case categories if this reference table is needed.

RT_CASE_SUBCATEGORY
contains case subcategories for case classification. This reference table is not preloaded during the installation. Customers must add all possible case subcategories if this reference table is needed.

RT_INCIDENT_TYPE
contains incident types for incident classification. This reference table is not preloaded during the installation. Customers must add all possible incident types.

RT_INCIDENT_CATEGORY
contains incident categories for incident classification. This reference table is not preloaded during the installation. Customers must add all possible incident categories if this reference table is needed.

RT_INCIDENT_SUBCATEGORY
contains incident subcategories for incident classification. This reference table is not preloaded during the installation. Customers must add all possible incident subcategories if this reference table is needed.

RT_PARTY_TYPE
contains party types for party classification. This reference table is not preloaded during the installation. Customers must add all possible party types.

RT_PARTY_CATEGORY
contains party categories for party classification. This reference table is not preloaded during the installation. Customers must add all possible party categories if this reference table is needed.

RT_PARTY_SUBCATEGORY
contains party subcategories for party classification. This reference table is not preloaded during the installation. Customers must add all possible party subcategories if this reference table is needed.

RT_EFILE_TYPE
contains e-file types for e-file classification. This reference table is not preloaded during the installation. Customers must add all possible e-file types.

RT_EFILE_CATEGORY
contains e-file categories for e-file classification. This reference table is not preloaded during the installation. Customers must add all possible e-file categories if this reference table is needed.

RT_EFILE_SUBCATEGORY
contains e-file subcategories for e-file classification. This reference table is not preloaded during the installation. Customers must add all possible e-file subcategories if this reference table is needed.

RT_EFILE_STATUS
contains e-file statuses. This reference table is not preloaded during the installation. Customers must add all possible e-file statuses.

RT_RR_TYPE
contains e-file types for e-file classification. This reference table is not preloaded during the installation. Customers must add all possible e-file types.

RT_RR_CATEGORY
contains regulatory report categories for regulatory report classification. This reference table is not preloaded during the installation. Customers must add all possible regulatory report categories if this reference table is needed.

RT_RR_SUBCATEGORY
contains regulatory report subcategories for regulatory report classification. This reference table is not preloaded during the installation. Customers must add all possible regulatory report subcategories if this reference table is needed.

RT_RR_STATUS
: contains regulatory report statuses. This reference table is not preloaded during the installation. Customers must add all possible regulatory report statuses.

RT_RR_READY
: contains report-ready code that represents the life cycle of a report. This reference table is preloaded during the installation.

RT_SOURCE_SYSTEM
: contains source system code. It is preloaded with SASECM for SAS Enterprise Case Management.

SNA_CONFIG_MASTER
: contains labels for Case Network Analysis match criteria defined in the SNA_CONFIG_MASTER table. It is preloaded with NATIONAL_ID.

Table 5.3 *Example Static Reference Tables*

REF_TABLE_NM	VALUE_CD	VALUE_DESC	PARENT_TABLE	PARENT_VAL
RT_CASE_TYPE	ML	Money Laundering		
RT_CASE_TYPE	FF	Financial Fraud		
RT_CASE_CATEGORY	CF	Check Fraud	RT_CASE_TYPE	FF
RT_CASE_CATEGORY	CK	Check Kiting	RT_CASE_TYPE	FF
RT_CASE_CATEGORY	CCF	Credit Card Fraud	RT_CASE_TYPE	FF
RT_CASE_CATEGORY	DCF	Debit Card Fraud	RT_CASE_TYPE	FF
RT_CASE_STATUS	N	New		
RT_CASE_STATUS	I	Investigate		
RT_CASE_STATUS	R	Review		
RT_CASE_STATUS	F	File		

Adding User-Defined Reference Tables

User-defined reference tables are not preloaded during the installation. Customers must add all user-defined reference table values to REF_TABLE_VALUE.

Table 5.4 *Example User-Defined Reference Tables*

REF_TABLE_NM	VALUE_CD	VALUE_DESC	PARENT_TABLE	PARENT_VAL
X_COUNTRY	USA	United States		
X_COUNTRY	MEX	Mexico		
X_COUNTRY	CAN	Canada		
X_STATE_PROVINCE	AL	Alabama	X_COUNTRY	USA

REF_TABLE_NM	VALUE_CD	VALUE_DESC	PARENT_TABLE	PARENT_VAL
X_STATE_PROVINCE	AK	Alaska	X_COUNTRY	USA
X_STATE_PROVINCE	AB	Alberta	X_COUNTRY	CAN
X_STATE_PROVINCE	ON	Ontario	X_COUNTRY	CAN
X_NATIONAL_ID_TYPE	SSN	Social Security Number		
X_NATIONAL_ID_TYPE	EIN	Employer ID Number		

Rules and Conventions for Database Fields and Reference Table Values

When you are creating custom fields and associated reference table values, you must follow the rules and conventions described in this section. Otherwise, version history labels will not display properly.

Custom Fields

Labels in the custom.properties file must be prefixed with **`field`** and suffixed with **`.label.txt`**. The following is an example of how to create a property for a custom field CASE.X_FED_REGULATOR_CD where CASE is the custom table name and X_FED_REGULATOR_CD is the custom field name:

```
field.case.x_fed_regulator_cd.label.txt = Fed Regulator
```

Custom Table Headers

For custom table headers in the version history, the label must be prefixed with **`table`** and suffixed with **`.label.txt`**. The following example is for a custom table header “X_ACCOUNT”:

```
table.x_account.label.txt = Accounts
```

Reference Tables

Using the same example for determining a reference table value, any field name ending in a reserved suffix is looked up in the table REF_TABLE_VALUE in the database. Only suffix a field name if you intend it to be looked up as a reference to another value or formatted as a specific value. The following table shows the reserved suffixes and how they are formatted and looked up.

Reserved Field Name Suffix	Converted Format
_CD*	String reference
_FLG*	Boolean reference
_AMT	Currency format
_DT	Date format
_TM	Time format

Reserved Field Name Suffix	Converted Format
_DTTM	Date/time format

*These suffixes are referenced in the REF_TABLE_VALUE for a converted value to display. If none are found, the original value of the field is displayed.

When creating reference table values for a column, the naming convention for the reference table name is to prefix the base column name with “X_RT_” and to strip off the column suffix. For example, to create reference table values for a field named CASE.X_FED_REGULATOR_CD, create records in REF_TABLE_VALUE with REF_TABLE_NM = “X_RT_FED_REGULATOR”.

Search Panels

Search Criteria

Fields that appear as search criteria on the case search panel are configured in the CASE_SEARCH_CRITERIA_FIELD table. Fields that appear as search criteria on the incident search panel are configured in the INCIDENT_SEARCH_CRITERIA_FIELD table. Fields that appear as search criteria on the party search panel are configured in the PARTY_SEARCH_CRITERIA_FIELD table. Fields that appear as search criteria on the report search panel are configured in the RR_SEARCH_CRITERIA_FIELD table. Fields that appear as search criteria on the e-file search panel are configured in the EFILE_SEARCH_CRITERIA_FIELD table. The structure for all five tables is the same, and each table contains the following columns:

Note: Searches on LONGVARCHAR fields are not supported.

USER_ID
: contains the user ID of the user this configuration applies to. The value equals “*” for the default configuration for all users.

TABLE_NM
: contains the table name of the search criteria field.

FIELD_NM
: contains the name of the search criteria field.

DISPLAY_ORDER_NO
: contains the search criteria display order. If the value is greater than 100, then the search criteria appear in a second column within the search criteria section.

REF_TABLE_NM
: optionally contains the reference table name used to populate a drop-down list of possible values to search for.

User-Specified Configurations

Default configurations are installed with the product for case, incident, party, report, and e-file search criteria. The default configurations only reference static fields (no user-defined fields). Customers can change the default configurations by modifying the

previous database tables directly. Customers can also define user-specific search criteria configurations by setting the USER_ID column value appropriately. If there is no user-specific configuration for the user, the default configuration is used.

Searchable Fields

Any static or user-defined field on any business object (case, incident, party, report, or e-file) table can be used as search criteria. Customers can reference fields in associated business object tables as search criteria. For example, the customer can specify PARTY.PARTY_FULL_NM in the CASE_SEARCH_CRITERIA_FIELD table to allow users to search for cases by party full name. The search looks for all cases that have one or more parties associated with the case with full name equal to the specified party full name.

The following special field can be used as search criteria within the CASE_SEARCH_CRITERIA_FIELD table:

TEMP.WORK_LIST_CASE_FLG
: returns all cases that the currently logged-on user can work on as defined in the workflow instances associated with the cases.

The following special field can be used as search criteria within the INCIDENT_SEARCH_CRITERIA_FIELD table:

TEMP.UNASSIGNED_INCIDENT_FLG
: returns all unassigned incidents (incidents not associated with a case).

The following special field can be used as search criteria within the RR_SEARCH_CRITERIA_FIELD table:

TEMP.WORK_LIST_REPORT_FLG
: returns all reports that the currently logged-on user can work on as defined in the workflow instances associated with the reports.

The following special fields can be used as search criteria within the RR_SEARCH_CRITERIA_FIELD and EFILE_SEARCH_CRITERIA_FIELD tables:

FORM_CONFIG.FORM_AGENCY_CD
: returns all reports or e-files whose form configuration matches the value given for the form configuration agency.

FORM_CONFIG.FORM_TYPE_CD
: returns all reports or e-files whose form configuration matches the value given for the form configuration type.

FORM_CONFIG.FORM_COUNTRY_CD
: returns all reports or e-files whose form configuration matches the value given for the form configuration country code.

The following special fields can be used as search criteria within all of the search criteria field tables:

TEMP.ENTITY_SUBSCRIPTION_FLG
: returns all the objects of the type being searched for that the currently logged in user has a subscription to.

Note: It is not necessary to enter wildcards (*) when entering search criteria. The search returns results that include all instances of the search criteria. For example: Entering 2009 in the Case ID field returns all cases that contain 2009 in their Case ID. You do not have to enter a wildcard with 2009.

Field Labels

The labels for all static search criteria shipped in the default configuration are specified in the SAS Enterprise Case Management resource bundle file (com.sas.solutions.casemgmt.i18n.AppResources.properties). All other labels must be specified in the custom resource bundle file. The naming convention for search criteria field resource bundle keys is as follows:

- field.<lowerCaseTableName>.<lowerCaseFieldName>.label.txt
- Here is an example:
  - field.party.party_full_nm.label.txt=Subject name:

User Interface Controls

Field Data Type	UI Control	Description
Any	Drop-down list	If REF_TABLE_NM is specified, show a drop-down list of possible values from the reference table. The REF_TABLE_NM is also useful for Boolean fields when searching for checked or unchecked values. By creating entries in REF_TABLE_VALUE, you can create a drop-down list where REF_TABLE_NM is 'X_RT_SEARCH_FLG'. The VALUE_CD field will hold the Boolean values 0 and 1, and the VALUE_DESC field will contain whatever is needed as the label in the drop-down list, (for example, True/False or Yes/No). See "Configuring Reference Tables in the Database" on page 82 for more information.
VARCHAR	Text	Show text input field.
BIGINT / DOUBLE	Number range	Show number from or to input fields.
BOOLEAN	Check box	Show a check box. This is useful only for special case search fields as defined in "Searchable Fields" on page 88. An example is TEMP.WORK_LIST_CASE_FLG.

Field Data Type	UI Control	Description
DATE / TIMESTAMP	Date range	Show date from or to input fields. The date format is determined by the user locale.

Search Filters

Fields that appear as search filters on the case search panel are configured in the CASE_SEARCH_FILTER_FIELD table. Fields that appear as search filters on the incident search panel are configured in the INCIDENT_SEARCH_FILTER_FIELD table. Fields that appear as search filters on the party search panel are configured in the PARTY_SEARCH_FILTER_FIELD table. Fields that appear as search filters on the report search panel are configured in the RR_SEARCH_FILTER_FIELD table. Fields that appear as search filters on the e-file search panel are configured in the EFILE_SEARCH_FILTER_FIELD table. The structure for all of the tables is the same, and each contains the following columns:

USER_ID
: contains the user ID of the user this configuration applies to. The value equals "*" for the default configuration for all users.

TABLE_NM
: contains the table name of the search filter field.

FIELD_NM
: contains the name of the search filter field.

DISPLAY_ORDER_NO
: contains the search filter display order.

REF_TABLE_NM
: optionally contains the reference table name used to populate a drop-down list of possible values to filter by.

User-Specified Configurations

Default configurations are installed with the product for case, incident, party, report, and e-file search filters. The default configurations only reference static fields (no user-defined fields). Customers can change the default configurations by modifying the previous database tables directly. Customers can also define user-specific search filter configurations by setting the USER_ID column value appropriately. If there is no user-specific configuration for the user, the default configuration is used.

Filterable Fields

Any static or user-defined field on any business object (case, incident, party, report, and e-file) table that can be used in conjunction with a reference table can be used as a search filter. Customers can reference fields in associated business object tables as search filters. For example, the customer can specify PARTY.PARTY_TYPE in the CASE_SEARCH_FILTER_FIELD table to allow users to filter cases by party type. The search looks for all cases that have one or more parties associated with the case with the specified party type.

Field Labels

The labels for all static search filters shipped in the default configuration are specified in the SAS Enterprise Case Management resource bundle file (com.sas.solutions.casemgmt.i18n.AppResources.properties). All other labels must be specified in the custom resource bundle file. The naming convention for search filter field resource bundle keys is the same as specified for search criteria.

Search Results

Fields that appear as search results on the case search panel are configured in the CASE_SEARCH_RESULT_FIELD table. Fields that appear as search results on the incident search panel are configured in the INCIDENT_SEARCH_RESULT_FIELD table. Fields that appear as search results on the party search panel are configured in the PARTY_SEARCH_RESULT_FIELD table. Fields that appear as search results on the report search panel are configured in the RR_SEARCH_RESULT_FIELD table. Fields that appear as search results on the e-file search panel are configured in the EFILE_SEARCH_RESULT_FIELD table. The structure for all three tables is the same, and each contains the following columns:

USER_ID
: contains the user ID of the user this configuration applies to. The value equals "*" for the default configuration for all users.

TABLE_NM
: contains the table name of the search result field.

FIELD_NM
: contains the name of the search result field.

DISPLAY_ORDER_NO
: contains the search result column display order.

REF_TABLE_NM
: optionally contains the reference table name used to render coded values as displayable values.

User-Specified Configurations

Default configurations are installed with the product for case, incident, party, report, and e-file search results. The default configurations only reference static fields (no user-defined fields). Customers can change the default configurations by modifying the previous database tables directly. Customers can also define user-specific search result configurations by setting the USER_ID column value appropriately. If there is no user-specific configuration for the user, the default configuration is used.

Displayable Fields

Any static or user-defined field in the business object (case, incident, party, report, or e-file) that contains one value can be shown in the search result table. You cannot show fields in associated business object (case, incident, party, report, or e-file) tables as search results. The following derived field can be shown in the incident search result table:

INCIDENT.CASE_ID
shows the ID of the associated case or is blank if the incident is unassigned.

Column Header Labels

The labels for all static search result column headers shipped in the default configuration are specified in the SAS Enterprise Case Management resource bundle file (com.sas.solutions.casemgmt.i18n.AppResources.properties). All other labels must be specified in the custom resource bundle file. The naming convention for search result field resource bundle keys is as follows:

- field.<lowerCaseTableName>.<lowerCaseFieldName>.header.txt
- Here is an example:
  - field.party.party_full_nm.header.txt=Subject Name

SAS Metadata Repository Properties

The following SAS Enterprise Case Management properties in the SAS Metadata Repository can be manually set from the SAS Management Console.

Property Name	Description
DB.Schema	The name of the database schema that contains the SAS Enterprise Case Management tables. The initial value is prompted for during the installation.
ECM.Policy.FinancialItemMigration	When a user is linking incidents to a case, this option determines whether that user is given an option to also add financial items from those incidents to the case. The default value is true.
Reassign.Case.User.Group.Or.Role	The name of the group or role defined in the SAS Metadata Repository. This name is used to populate the drop-down list of users to set as primary owner. The initial value is “Ent Case Mgmt Users”, which contains all SAS Enterprise Case Management users.
Table.Records.Per.Page	The default number of records to show per page within tables and DataGrids that support pagination. The initial value is 20.

Property Name	Description
WF.Sleep.Time.Millis	The default workflow transition wait time. SAS Enterprise Case Management waits the given number of milliseconds after any case or report save when a workflow is updated. This helps ensure that any policies executed as a result of the workflow transition complete before the user edits the object again. The initial value is 1000. *Note:* It is not recommended to set the WF.Sleep.Time.Millis property to a value of less than 1000 milliseconds.

Adding Data to Relationships

Customizing Case and Party Relationships

Using user-defined fields attached to a case, you can add extra data to relationships between a case and a party. The user interface component used to display this relationship is EnhancedPartyRelationshipGrid.

The EnhancedPartyRelationshipGrid component is similar in appearance and function to the standard CasePartiesGrid. It presents relationship information between a set of parties and a case in a tabular view. However, it adds the following features not found in CasePartiesGrid:

- tabs, one for each relationship type, to easily show a filtered view of which parties are linked to the case by each relationship
- a **Relationship Details** panel, which is docked to the side of the component that displays the relationship data for the selected party
- the ability to easily modify and remove relationships between the case and a party

Adding Fields to Relationships

Associating data with case/party relationships is done by adding user-defined fields to cases. It is important that you set up your user-defined field data properly for this functionality to work.

First, you must create a reference table that defines the valid set of relationships from a case to a party (for example, X_RT_RELATION_TYPE). Then, create a field in CASE_UDF_DEF of type BIGINT that will be used to map cases to the parties linked to them (for example, X_REL.X_PARTY_RK). This is a many-to-one field, as a single case can be related to multiple parties.

Note: Although the name of this field can be set to any valid value, it is strongly recommended that you set the udf_nm value to X_PARTY_RK.

Next, for each relationship type defined in the reference table, you can create any number of fields in CASE_UDF_DEF that correspond to that relationship type. The

fields must follow a specific naming convention. Specifically, they must have the same udf_table_nm value as the party key field you created previously, and their names must be in this format: *tableName.X_relValueCd_fieldName*.

For example, suppose that you have a relationship type defined for suspects. This relationship type has a value_cd of S in REF_TABLE_VALUE. You want to create fields representing the start date and end date of each suspect relationship. Assuming you are using the standard UDF table name X_REL, you could do this by creating the following fields in CASE_UDF_DEF as type DATE:

- X_REL.X_S_START_DATE
- X_REL.X_S_END_DATE

These are many-to-one fields, as a single case can be related to multiple parties.

Displaying Relationship Data in a UI Definition

After relationship data is defined, you can display and edit it in UI definitions with the EnhancedPartyRelationshipGrid component. This component takes many of the same parameters that other DataGrids do. However, the following parameters are necessary to display relationship information:

Parameter	Description
relationshipTypeTable	The reference table containing the relationship types to display.
dialogScreenId	The ID of the window in the UI definition to use when editing relationships.
dialogWidth	The width of the dialog to use for the window.
dialogHeight	The height of the dialog to use for the window.
tableName	The name of the UDF table containing the relationship data fields for the relationship types (for example, X_REL).
partyKeyField	The name of the field in the UDF table that holds the keys of the related parties (for example, X_PARTY_RK).
singleGrid	If this parameter is set to **`true`**, the component will not have tabs providing filtering for each relationship type. Only a single grid will be displayed, displaying all relationships together.
multiSelect	By default, the user can only select a single row at a time for deletion. If this parameter is set to **`true`**, a check box column is displayed, enabling the user to select multiple rows to delete at the same time.

Other parameters exist to configure the appearance and behavior of the component. For a complete list of all options available in the EnhancedPartyRelationshipGrid, see the Custom Page Builder documentation under the **Administration** menu of SAS Enterprise Case Management.

Designing a Screen to Edit Relationship Data

Every EnhancedPartyRelationshipGrid references a window that is used to view and edit relationship details. This window is used to render both the preview display docked on the side of the grid itself, as well as the modal dialog box that allows the user to edit the relationship details.

In almost all instances, these windows will have the following requirements:

- When the window is editable, the user should be able to select the relationship types that link the party to the case.
- For any selected relationship types, the associated data fields should be displayed for edit or viewing.
- When the window is editable, when a user clicks **OK** to dismiss the dialog box, validation should occur that ensures at least one relationship type is selected.
- When the window is editable, when a user clicks **OK** to dismiss the dialog box, relationship data should be updated to reflect what was entered (assuming a relationship type was selected).

To meet these requirements, these windows will typically share a common design. This design can be seen in the sample UI definitions shipped with SAS Enterprise Case Management. A sample of this window design is also shown in the following code:

```
<screen id="relDetails">

      <initialize>
         <set name="TEMP.PARTY_NAME"
              value="GetPartyField(X_REL.X_PARTY_RK, 'PARTY.PARTY_FULL_NM')"/>
      </initialize>

      <field name='TEMP.PARTY_NAME_LABEL' type='component'
            component-name='LabelComponent'>
         <label>Party Name: <eval>TEMP.PARTY_NAME</eval></label>
         <param name='fontStyle' value="'bold'"/>
      </field>

      <!-- Relationship selector component creates and updates two temp fields:
           TEMP.REL_SELECTOR.RELATIONSHIPS and TEMP.REL_SELECTOR_RELATIONSHIP_DESCS.
           It should always be on screen (even if not visible) to make the right
           subset of fields below visible. -->
      <field type="component" component-name="PartyRelationshipSelector"
            name="TEMP.REL_SELECTOR" visible="!isScreenReadOnly()">
         <on-change>
            <set-visible name='TEMP.LABEL_SUSPECT'
               test='contains(TEMP.REL_SELECTOR.RELATIONSHIPS, "S")'/>
            <set-visible name='X_REL.X_S_START_DATE'
               test='contains(TEMP.REL_SELECTOR.RELATIONSHIPS, "S")'/>
            <set-visible name='X_REL.X_S_END_DATE'
               test='contains(TEMP.REL_SELECTOR.RELATIONSHIPS, "S")'/>
            <set-visible name='TEMP.LABEL_VICTIM'
               test='contains(TEMP.REL_SELECTOR.RELATIONSHIPS, "V")'/>
```

```
            <set-visible name='X_REL.X_V_STATEMENT'
               test='contains(TEMP.REL_SELECTOR.RELATIONSHIPS, "V")'/>
         </on-change>
      </field>

      <field name='TEMP.LABEL_SUSPECT' type='component'
            component-name='LabelComponent'>
         <label>Details for relationship "Suspect"</label>
         <param name='fontStyle' value="'bold'"/>
      </field>
      <field name='X_REL.X_S_START_DATE' type='date'>
         <label>Suspect Start Date</label>
      </field>
      <field name='X_REL.X_S_END_DATE' type='date'>
         <label>Suspect End Date</label>
      </field>

      <field name='TEMP.LABEL_VICTIM' type='component'
            component-name='LabelComponent'>
         <label>Details for relationship "Victim"</label>
         <param name='fontStyle' value="'bold'"/>
      </field>
      <field name='X_REL.X_V_STATEMENT' type='textarea' rows='3'>
         <label>Statement</label>
      </field>

      <finalize>
         <!-- Updates relationships to reflect what's in the party relationship
              selector, but only if 1 or more rows is selected. -->
         <validation test="SetRelationshipsToParty('X_REL.X_PARTY_RK',
               'TEMP.REL_SELECTOR')">
            <errmsg>At least one relationship type must be selected.</errmsg>
         </validation>
      </finalize>
   </screen>
```

These windows typically contain a PartyRelationshipSelector component to enable the user to select relationships with which to link a party to the case. This also controls what UDF fields are visible through on-change events. In the **`<finalize>`** block, the SetRelationshipsToParty() function is used to verify that at least one relationship is selected. Assuming that this criterion is met, it also updates the relationship information to reflect what was entered on the screen.

Note: By default, the PartyRelationshipSelector component displays a description field for each relationship type. In this field, the user can enter free-form information. This column of fields exists for backwards compatibility with earlier versions of SAS Enterprise Case Management, where this field was the only means users had to associate data with relationships. It is typically not needed now that you can use UDF fields to attach data to relationships. If you do not plan to use the description fields in the PartyRelationshipSelector, you can hide the column entirely by setting a parameter named “descriptions” to **`false`**. For more information about this parameter, or the PartyRelationshipSelector component in general, see the Custom Page Builder documentation under the **Administration** menu of SAS Enterprise Case Management.

Chapter 6

Using the Custom Page Builder

Overview of the Custom Page Builder **98**

Customizable User Interfaces **99**

Assigning the Custom Page Builder Permission **100**

Working with User Interface Definitions **100**
View User Interface Definitions 100
Edit the User Interface Definition 100
Upload the User Interface Definition 100
Delete the User Interface Definition 101

Valid XML Elements and Descriptions for User Interface Definitions **101**

Customizable Menus **116**
Menu Customization 116
Example Menu Definition 117
Menu Definition Structure 117
XML Elements 118
Examples 119

Working with Menu Definitions **120**
View Menu Definitions 120
Edit the Menu Definition 120
Upload the Menu Definition 121
Customize the Banner Title Content 121

Custom Page Builder: Creating Custom Help **122**

Expressions and Functions **123**
About Expressions and Functions 123

Lazy Initialization for Data Displayed on a Tab **123**
Overview 123
Best Practices 124
Example 124

Customization Examples **125**
How to Customize the User Interface Definition Files 125
About Required and Non-Required Fields 125
Customize Error Messages 126
Example: Hiding a Required Field 126
Example: Specifying a Read-Only Field 126
Example: Specifying the Number of Decimal Digits 127
Example: Validating Dates 127
Example: Specifying Drop-Down Lists and Radio Buttons 127

Example: Specifying a Text Area and a Text Field 127
Example: Creating Custom Filtered Subjects 128
Dynamic Conditional Logic in User Interface Definition Files 129
Example: Creating a Custom Function 131
Creating a Custom Component 132

DataStores and DataGrids 135
Overview 135
Example: A DataGrid with a Single DataStore 135
Example: Two Filtered DataGrids Sharing a Single DataStore 136
GridActions 138
Bulk Field Update Action 140
Incident Triage Action 141
Custom GridActions 143
DataGrid Column Renderers 144
DataGrid Column Sorters 147

Custom Page Builder Components 148
Overview 148
Static Component Field Formatters 149
Custom Action 149

Custom Page Builder Directives 152
Overview 152
Case 152
viewECMCaseReport 153
ViewECMObject 153
ViewGenericPage 154
PreviewECMReport 154
ECMWorkflowCallback 154
CorrectReport 155
ECMWelcome 155
NewECMObject 156

Overview of the Custom Page Builder

User interface (UI) definition files specify the form and content of pages presented in SAS Enterprise Case Management, the data that is captured, and how data is validated. UI definition files must be uploaded from the **Administration** tab within SAS Enterprise Case Management.

Using the Custom Page Builder, you can make fields either mandatory or optional. You can display or hide fields depending on the entries selected for other fields. In addition, default values and validations can be freely defined.

The following screens can be customized:

- case detail screen
- incident detail screen
- party (or subject) detail screen
- e-file detail screen
- report detail screen

In addition, the Custom Page Builder enables you to do the following:

- specify the order of fields on a screen
- group fields into sections, subsections, and tabs
- override the default labels for fields
- specify property keys to use strings from custom.properties (or other properties files)
- filter subjects to be displayed based on selected criteria
- hide or show fields, sections, subsections, and tab-sections
- specify whether a field is Read-Only or can be edited
- configure the number of decimal digits visible for numeric fields
- specify the default value for any field
- specify custom validation expressions that must be passed before the user is able to continue through a workflow
- specify the maximum and minimum values allowed in date and number fields
- control field rendering for certain fields:
  - For single-select list fields, you can choose between a drop-down list and radio buttons.
  - For string fields, you can choose between a text field and a text area, and also control the size of the text field or the text area.

Note: You can also apply the above list to auxiliary (aux) fields.

Certain fields are required by design, and you can specify optional fields as required fields. Although you cannot specify required fields as optional, you can hide these fields. For more information see "Example: Hiding a Required Field" on page 126.

Customizable User Interfaces

You can customize the following subsets of user interfaces with the Custom Page Builder for the following subject areas:

- Create/Edit Case
- Create/Edit Incident
- Create/Edit Subject
- Create/Edit Reports
- Create/Edit E-Files
- Add/Edit Row of User-Defined Table
- generic listing pages for showing lists of entities as well as data from external sources

Assigning the Custom Page Builder Permission

To load and update user interfaces and menus using the Custom Page Builder, you must assign a global capability to an existing or new role. Then you must assign the user to that role. Use SAS Management Console to specify the role and to assign the user to the role. See *SAS Management Console: Guide to Users and Permissions* for more information.

Working with User Interface Definitions

View User Interface Definitions

To view user interface definitions, select the **Administration** tab and select the user interface definition that you want to view.

Edit the User Interface Definition

To edit a user interface definition:

1. Select the **Administration** tab.
2. Click **Download User Interface Definition** from the pop-up menu.
3. Edit the file with a text editor or with your favorite XML editor.

   Note: You can validate the structure of the file against the uiDefinition.dtd file. The file can be found at *SAS-middle-tier-installation-directory*`/deploy/sample/uidef/uiDefinition.dtd` for UNIX platforms, or *SAS-middle-tier-installation-directory*`\deploy\sample\uidef\uiDefinition.dtd` for Windows platforms.
4. Upload the changes. For more information, see "Upload the User Interface Definition" on page 100.

Note: You do not need to restart the server after uploading the user interface definition.

Upload the User Interface Definition

To upload a user interface definition:

1. From the **Administration** menu, select **UI Definitions**.
2. Click **Upload UI definition**. The Upload User Interface Definition File window appears.
3. Enter the path to the file, or click **Browse** to navigate to it.
4. Enter a description. This step is optional.
5. Click **Validate XML**. Any warnings or errors found in the user interface definition file are displayed.
6. When you are satisfied with the results, click **Upload User Interface Definition**.

Note: You do not need to restart the server after uploading the user interface definition.

Delete the User Interface Definition

Deleting a user interface definition removes the file from the system. This is an unrecoverable operation. To delete a user interface definition:

1. Select the **Administration** tab.
2. For the corresponding user interface definition that you want to delete, click the action menu, and then click **Delete**.

Valid XML Elements and Descriptions for User Interface Definitions

A user interface definition file is an XML document consisting of a top-level **`<ui-definition>`** element with attributes and child elements that describe the form and content of the pages, their validations, derived fields, and conditional logic. The user interface definition files must be written in valid XML that conforms to the structure described in the document type definition (DTD) uiDefinition.dtd. For more information about where to locate a copy of this file, see "Edit the User Interface Definition" on page 100.

Note: For information about XML, including how to handle special characters, see **`http://www.w3.org/TR/REC-xml/`**.

The following table describes the XML format used in the user interface definition files:

Table 6.1 *XML Format*

Element	Description
`<ui-definition>`	The top-level element that describes the screens in the UI definition. **Attributes:** • **`id`** — A unique identifier for this user interface definition (user-defined). This must be a valid XML name. • **`type`** — Indicates the type of object that this UI definition is used for. Valid values include **`case`**, **`incident`**, **`party`**, **`report`**, and **`efile`**. **Child elements:** • An optional **`<title>`** element. The title appears on the administration page, but it is not visible to the end user (for example, the user who is editing an issue). • Zero or more **`<function>`**, **`<datagrid-renderer>`**, or **`<component>`** elements. • Zero or more **`<screen>`** elements.

Element	Description
`<function>`	Declares a custom function. **Attributes:** • **`name`** — The name that is used to reference the custom function. Custom function names must begin with “C_” (or “c_”). • **`qualified-class-name`** — The fully qualified class name. For more information about creating custom functions, see “Example: Creating a Custom Function” on page 131.
`<datagrid-renderer>`	Defines a custom renderer for the **`<datagrid-column>`** element. **Attributes:** • **`id`** (required) — A logical identifier that **`<datagrid-column>`** elements can reference this renderer with. This should begin with "C_" (or "c_"). **Child elements:** • A JavaScript function to use for rendering DataGrid cells. This should be wrapped in a CDATA block. The **`<message>`** and **`<eval>`** elements are permitted. See the “Custom Column Renderers” on page 146 for more information.
`<component>`	Declares a custom component. **Attributes:** • **`name`** — The name that is used to reference the custom function. Custom function names must begin with “C_” (or “c_”). • **`qualified-class-name`** — The fully qualified class name.

Element	Description

`<screen>`	Describes the appearance and behavior of a single screen.
	Attributes:
	• **`id`** — The screen ID. This must be unique for each screen. For each entity, the following specific IDs are required in the UI definition:
	  • **`<screen id="case">`**
	  • **`<screen id="incident">`**
	  • **`<screen id="party">`**
	  • **`<screen id="report">`**
	  • **`<screen id="efile">`**
	  • **`<screen id="financialItems">`**
	• **`help`** (optional) — A URL specifying the Help location for this screen.
	Child elements:
	• An optional **`<app-title>`** element providing the application title for a screen.
	• An optional **`<title>`** element providing the screen title.
	• An optional **`<menu>`** element.
	• An optional **`<help-text>`** element that provides an alternative means of specifying Help instead of using a URL in the **`help`** attribute.
	• An optional **`<initialize>`** section that describes any initialization that must be performed before the screen is rendered.
	• Any number of **`<action-group>`**, **`<field>`**, **`<datagrid>`**, **`<datastore>`**, **`<section>`**, **`<tab-section>`**, **`<column-layout>`**, or **`<if>`** elements.
	• An optional **`<finalize>`** section that describes any validations or computations that must be performed when the user clicks **Save**.

`<app-title>`	Used to override the main application name in the banner. **Child elements:** • Zero or more **`<eval>`** elements. • Zero or more **`<message>`** elements.
`<menu>`	Allows for the overriding of the main menu in the banner for generic screens. Only a menu with a single return button is supported. To provide this menu, use **`<menu>returnNavigationMenu</menu>`**.
`<help-text>`	An alternative means of specifying help instead of using a URL in the **`help`** attribute of a **`<screen>`**, **`<field>`**, or **`<datagrid>`**.

Element	Description
`<initialize>`	An optional section that contains code that is executed before the screen appears. **Child elements:** • Zero or more **`<set>`** elements that set variables to some computed value. • Zero or more **`<if>`** elements that contain conditional **`<set>`** or **`<validation>`** elements. • Zero or more screen-level **`<validation>`** elements that are performed before the screen is rendered.
`<column-layout>`	Allows the layout of fields in multiple columns. • **`id`** — The section ID. • **`visible`** (optional) — An expression that is evaluated by the expression handler to determine if the columns are visible. The default value is **`true`**. **Child elements:** • Any number of **`<column>`** or **`<if>`** elements.
`<column>`	Contains the contents of a single column in a **`<column-layout>`** element. **Attributes:** • **`required`** (optional) — An expression that is evaluated by the expression handler to determine if the tab must be filled in before saving. The default value is **`false`**. • **`visible`** (optional) — An expression that is evaluated by the expression handler to determine if the column is visible. The default value is **`true`**. **Child elements:** • Any number of **`<field>`**, **`<section>`**, **`<tab-section>`**, **`<column-layout>`**, or **`<if>`** elements.
`<finalize>`	An optional section that contains code that is executed when the screen is considered complete (usually when the user clicks **Save**). **Child elements:** • Zero or more **`<set>`** elements that compute derived fields. These fields are evaluated when the user clicks **Save**. • Zero or more **`<if>`** elements that contain conditional **`<set>`** or **`<validation>`** elements. • Zero or more screen-level **`<validation>`** elements that are checked when the user clicks **Save**.

Element	Description
`<set>`	Evaluates an expression and stores its value in the memory hash table. **Attributes:** • **`name`** — The variable name that corresponds to a field. If the name is not specified, this is treated like a function call having no return value. • **`value`** — An expression that is evaluated with the resulting value stored in the named variable in the memory hash table.

Element	Description

`<field>`	Describes a prompt for user input.
	Attributes:
	• **`name`** — The name of the field. This should be one of the names associated with the type of user interface definition that you are editing.
	• **`type`** — The GUI component that is used for the input control. The following are valid values:
	  • **`string`**
	  • **`number`**
	  • **`boolean`**
	  • **`dropdown`**
	  • **`checkbox`**
	  • **`radio`**
	  • **`date`**
	  • **`textarea`**
	  • **`readonly`**
	  • **`component`**
	• **`component-name`** (optional) — The name of a fixed screen component. Applicable only if **`type="component"`**. Ignored otherwise.
	• **`length`** (optional) — The width of the input control on the screen (not necessarily the field length in the database).
	• **`rows`** (optional) — Applies to text area type only. Indicates the number of rows in the text area.
	• **`max-length`** (optional) — The maximum length of the input content allowed in the text input.
	• **`decimal-digits`** (optional) — Limits the number of digits in number format.
	• **`min`** (optional) — Minimum value for dates and numbers.
	• **`max`** (optional) — Maximum value for dates and numbers.
	• **`minSelectableDate`** (optional) — Minimum selectable date. This attribute affects only the date chooser and not validation.
	• **`maxSelectableDate`** (optional) — Maximum selectable date. This attribute affects only the date chooser and not validation.
	• **`default`** (optional) — An expression whose value is used as the default value for the field. This value is used only when creating a new object.
	• **`values`** (optional) — An expression whose value is a list of items used to populate a drop-down list, check box, or radio button group. Each item in the list is a label and value pair, where label is the displayed value, and value is the internally used value. Using the values attribute with a check box displays a group of check boxes to be multi-selected.
	  Note: Although multiple values for a **`string`**, **`number`**, **`Boolean`**, or **`text area`** type field are permissible, a result will not be set if used, since these field types can handle only a single value.
	• **`align`** (optional) — The alignment of the input field's label. Valid values are **`top`**, **`left`**, and **`inline`**. The default is **`left`**.

Element	Description
`<field>`	Describes a prompt for user input. **Attributes:** • **`required`** (optional) — An expression that is evaluated by the expression handler to determine whether the user must complete the field before saving. Default is **`false`**. • **`visible`** (optional) — An expression that is evaluated by the expression handler to determine whether the field is visible. Default is **`true`**. • **`readonly`** (optional) — An expression that is evaluated by the expression handler to determine whether the field is Read-Only. Default is **`false`**. You can also specify **`neverBeenSaved`**. • **`escape-xml`** (optional) — By default, any XML character in a Read-Only field is escaped. Specifying **`false`** keeps the characters from being escaped. Default is **`true`**. Valid only for Read-Only fields. • **`label-width`** (optional) — Applies a width to the HTML element that holds a field's label. It is useful to line up fields and their labels when they are inside a column. An example is **`label-width="'25%'"`**. **Child elements:** • A **`<label>`** element that specifies the label and prompt for this input field. • Zero or more **`<validation>`** elements, which are evaluated when the user clicks **Save**. • Zero or more **`<param>`** elements, which are applicable only if **`type="component"`**. • An optional **`<on-change>`** element, which describes any dynamic actions to be executed when the field value changes. • An optional **`<true-label>`** element, which is applicable only if **`type="boolean"`**. • An optional **`<false-label>`** element, which is applicable only if **`type="boolean"`**.
`<datastore>`	Describes a list of data that can be shown in a DataGrid. **Attributes:** • **`id`** (required) — A unique name for the instance of **`<datastore>`**. This can be used by a **`<datastore-ref>`** element. • **`data`** (required) — A sorted map of fields representing the **`<datastore>`** data. This data will eventually map to cell data in a **`<datagrid>`**.
`<datastore-ref>`	A reference to a previously defined **`<datastore>`** element. **Attributes:** • **`id`** (required) — The ID of the previously defined **`<datastore>`** element.

Element	Description
`<datagrid>`	Describes a grid of tabular data. **Attributes:** • **`name`** (required) — A unique name for the instance of **`<datagrid>`**. • **`selectedKeyField`** (required for actions on row data) — Name of the column or field in the grid's data that represents the key field of the row. It is needed to pass a row's unique identifier for actions that act upon selected rows. • **`columns`** (optional) — Used to provide a function that returns a list of DataGrid column entries to be shown in the **`<datagrid>`** element instead of using **`<datagrid-column>`** elements. • **`visible`** (optional) — An expression that is evaluated by the expression handler to determine whether the field is visible. The default is **`true`**. • **`readonly`** (optional) — An expression that is evaluated by the expression handler to determine whether the field is read-only. The default is **`false`**. • **`component-name`** (optional) — The name of the component to use when adding a **`<datagrid>`** element-based component. **Child elements:** • An optional **`<label>`** element that specifies the label or description for this table. • An optional **`<help-text>`** section that is an alternative to specifying a Help URL attribute. • Zero or more **`<validation>`** elements, which are evaluated when the **Save** button is clicked. • Either one of a **`<datastore>`** or **`<datastore-ref>`** element. • An optional **`<layout-info>`** element that specifies layout information for the grid. • Zero or more **`<datagrid-column>`** elements that each define a single DataGrid column. • Zero or more **`<filter>`** elements that define what subset of data the DataGrid should display. • Zero or more **`<grid-action>`** elements that each define a single grid action. • Zero or more **`<param>`** elements, which are applicable only if **`type="component"`**. • An optional **`<on-change>`** element, which describes any dynamic actions to be executed when table data changes. • An optional **`<on-select>`** element, which describes any dynamic actions to be executed when a row is selected in the data grid.

Element	Description
`<datagrid-column>`	Provides a single column entry to a DataGrid. **Attributes:** • **`name`** (required) — The name of the field to be used for the column in each row. • **`id`** (optional) — If the ID is specified, then the ID is used for the **`<datagrid-column>`** element. Otherwise, the name attribute is used. If a DataGrid has two columns displaying the same field, but using different renderers, unique values for the **`id`** attribute should be given to each **`<datagrid-column>`** element. This allows the columns to be distinguishable from one another and rendered properly. **Child elements:** • A required **`<label>`** element that specifies the header for this column. • An optional **`<datagrid-renderer-ref>`** element that defines how to format the column data. • An optional **`<datagrid-column-sorter>`** element that defines how the column should be sorted.
`<layout-info>`	Specifies layout information for a DataGrid. **Attributes:** • **`page-size`** (optional) — Determines how many rows of data should be visible on one page. If omitted, the value of the metadata property Table.Records.Per.Page is used. • **`page-size-choices`** (optional) — A list of page sizes that are listed in the footer of the DataGrid. When a value in this list is clicked, the page size of the DataGrid is changed to that value. This should be a list of whole numbers separated by commas (for example, “10, 25, 50, 100”). A value of **`all`** can be used for the last value instead of a number. This adds an item that displays all available data on one page. If this attribute is omitted, no page size choices are displayed.

Element	Description
`<grid-action>`	Defines a URL or javascript that will be executed. **Attributes:** • **`id`** (required) — Specifies the ID field. Each ID must be unique for the screen. • **`url`** (optional) — If **`output-destination`** is not set to **`javascript`**, this specifies the URL that is called to execute the action. If **`output-destination`** is set to **`javascript`**, this is the JavaScript snippet to execute. • **`output-destination`** (optional) — Specifies the output format. The possible values are **`inline`**, **`window`**, **`new-window`**, or **`javascript`**. The default is **`window`**. A value of **`window`** specifies replacing the existing window's contents with the result page. A value of **`new-window`** specifies showing the result page in a pop-up window. A value of **`inline`** causes a refresh of the parent DataGrid when the action completes, but no other change in the structure of the current page. A value of **`javascript`** means that the **`url`** parameter should be treated as JavaScript to execute, instead of a URL to submit. • **`render-type`** (optional) — Specifies how the action is rendered. The possible values are **`always`**, **`select`**, and **`row`**. A value of **`always`** renders the action as a button at the top of the table that is always enabled. A value of **`select`** renders the action as a button at the top of that table that is enabled only when at least one row is selected. A value of **`row`** renders the action as an icon in each row and is an action that is performed just for that row. The default is **`always`**. • **`request-type`** (optional) — Specifies whether the URL should be made using **`http post`** or **`http get`**. The values are **`post`** or **`get`**. The default is **`post`**. • **`visible`** (optional) — Specifies whether the action is visible. The default is **`true`**. • **`enabled`** (optional) — Specifies whether the action is enabled. The default is **`true`**. **Child elements:** • A required **`<label>`** element that specifies the label on the button. • An optional **`<tool-tip>`** element that provides a tool tip for actions that are rendered as buttons. For **`<grid-action>`** elements with **`render-type`** set to **`row`**, this parameter is ignored. • A required **`<url>`** element that specifies the action being called. • **`<eval>`** — This element is a child of the **`<url>`** element. The text of this element is treated as an expression and evaluated. • An optional **`<image>`** element that defines the URL used to retrieve the image. This is needed when the value for render-type is row. • Zero or more **`<param>`** elements, which are used for **`<datagrid>`** components. • An optional **`<validations>`** element that contains any validations that must evaluate to true for the GridAction to execute. • An optional **`<callback>`** element that defines an action that will be executed on completion of the original action call.

Element	Description
`<validations>`	An optional section that contains code that is executed before a GridAction executes. **Child Elements:** • Zero or more **`<set>`** elements that set variables to some computed value. • Zero or more **`<validation>`** elements that are checked when the user attempts to execute the GridAction. If any of these validations do not evaluate to **`true`**, the appropriate error message is displayed and the GridAction is not executed.
`<callback>`	A JavaScript snippet that is executed when the GridAction is successful. Callbacks are only executed for GridActions with **`output-destination`** set to **`inline`**.
`<tool-tip>`	Specifies a tool tip for a component or action. **Attributes:** • None **Child Elements:** • The child text of this component makes up the tool tip text. It can contain any number of **`<message>`** or **`<eval>`** elements.
`<datagrid-renderer-ref>`	Specifies what renderer will be used to format cell contents in a DataGrid column. This is a reference to either a built-in renderer or a custom renderer. **Attributes:** • **`id`** (required) — The ID of a DataGrid renderer. • **`args`** (optional) — Optional arguments to the renderer. Multiple arguments are separated by colons (:). Not all renderers take arguments.
`<datagrid-column-sorter>`	Specifies sorting logic for a DataGrid column. **Attributes:** • name (required) — The name of the sorter to use (for example, **`userDisplayNameSorter`** or **`refTableSorter`**.) **Child elements:** • Zero or more **`<param>`** elements.
`<filter>`	Adds a filter expression that is run for each possible record in a DataGrid. Each **`<filter>`** expression must evaluate to **`true`** for that record to be displayed in the grid. This filter typically checks the value of one or more fields in the row of the grid against an expected value. **Attributes:** • **`name`** (optional) — The name of this filter. • **`value`** — The expression that must evaluate to **`true`** for a row to be displayed.

Element	Description
`<label>`	The label or prompt displayed for a field. If styled input is required (for example, multiple lines with bullets), you can specify this element as a CDATA section with embedded HTML. **Attributes:** • **`separator-visible`** — By default, a separator (colon) is added to the label. Specifying **`false`** suppresses the separator. **Child elements:** • The child text of this component makes up the label text. It can contain any number of **`<message>`** and **`<eval>`** elements.
`<param>`	A parameter passed to a fixed screen component. **Attributes:** • **`name`** — The parameter name. The name is required when the parameter is for a field, but should be empty when the parameter is for a message. • **`value`** — An expression that is evaluated, with the result used as the parameter value. If no value is specified, the content of the element is used. *Note:* The entire **`<param>`** statement, including the attributes and values, must be on one line.
`<validation>`	A test that is performed at the screen, section, or field level. The test is evaluated by the expression handler (usually, when the user clicks **Save**). If it is false, the error message is displayed and the user remains on the same screen. **Attributes:** • **`test`** — An expression that is evaluated by the expression handler using the current contents of the memory hash table (which will include the values from the database and everything the user has entered up to this point). For example, if the input field was a month number in a field named MONTH, the expression might be **`test="month ge 1and month le 12"`**. **Child elements:** • An **`<errmsg>`** element that describes the message to display if the test fails.
`<errmsg>`	An error message that is displayed if a validation fails. **Child elements:** • Zero or more **`<message>`** elements.

Element	Description
`<section>`	Describes the appearance and behavior of a section of other elements. **Attributes** • **`id`** — The section ID. This must be a valid XML name (also a valid SAS name). • **`required`** (optional) — An expression that is evaluated by the expression handler to determine whether the section should display the required indicator. Default is **`false`**. • **`visible`** (optional) — An expression that is evaluated by the expression handler to determine whether the section is visible. Default is **`true`**. • **`expanded`** (optional) — An expression that is evaluated by the expression handler to determine whether the section is expanded by default. Default is **`true`**. **Child elements:** • An optional **`<label>`** element providing the section label and title. • Zero or more **`<field>`** elements. • Zero or more **`<section>`** elements. • Zero or more **`<if>`** elements. • Zero or more **`<validation>`** elements, which are evaluated when the user clicks **Save**. • Zero or more **`<action-group>`** elements.
`<tab-section>`	A **`<tab-section>`** element can be nested under a **`<section>`**, **`<tab>`**, or **`<screen>`** element to define a set of tab pages. **Attributes:** • **`id`** — The section ID. This must be a valid XML name (also a valid SAS name). **Child elements:** • One or more **`<tab>`** elements.

Element	Description
`<tab>`	A tab is nested under a **`<tab-section>`** and defines a tab page. **Attributes:** • **`id`** — The section ID. This must be a valid XML name (also a valid SAS name). • **`required`** (optional) — An expression that is evaluated by the expression handler to determine whether the tab should display the required indicator. Default is **`false`**. **Child elements:** • An optional **`<label>`** element providing the tab label and title. • • An optional **`<initialize>`** element containing one or more child **`<set>`** elements. See "Lazy Initialization for Data Displayed on a Tab" on page 123 for more information. • Zero or more **`<field>`** elements. • Zero or more **`<section>`** elements. • Zero or more **`<if>`** elements. • Zero or more **`<validation>`** elements, which are evaluated when the user clicks **Save**. • Zero or more **`<action-group>`** elements.
`<if>`	A group of other elements that is conditionally included on the screen. **Attributes:** • **`test`** — An expression that is evaluated by the expression handler to determine whether the contents of the **`<if>`** element should be active or not. **Child elements:** • Any number of **`<field>`** elements. • Any number of **`<section>`** elements. • Any number of **`<if>`** elements. • Zero or more **`<action-group>`** elements.
`<message>`	A localized message. **Attributes**: • **`key`** — The key of the string in the resource bundle. **Child elements:** • Zero or more **`<param>`** elements.
`<true-label>`	True label for fields of type **`Boolean`**. **Child elements:** • Zero or one **`<message>`** elements.
`<false-label>`	False label for fields of type **`Boolean`**. **Child elements:** • Zero or one **`<message>`** elements.

Element	Description
`<on-change>`	A group of dynamic actions to be executed when the value of the field changes. **Child elements:** • Zero or more **`<set_visible>`** elements. • Zero or more **`<set_required>`** elements. • Zero or more **`<set_value>`** elements. • Zero or more **`<set-values>`** elements. • Zero or more **`<refresh>`** elements. • Zero or more **`<call>`** elements.
`<on-select>`	A group of dynamic actions to be executed when a row is selected in the DataGrid. **Child elements:** • Zero or more **`<set_visible>`** elements. • Zero or more **`<set_value>`** elements. • Zero or more **`<set_values>`** elements. • Zero or more **`<refresh>`** elements. • Zero or more **`<call>`** elements.
`<set-visible>`	A dynamic action that sets the visibility of a field. **Attributes:** • **`name`** — The name of a field on the current screen. • **`test`** — An expression that is evaluated by the expression handler to determine whether the field will be shown or hidden. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.
`<set-required>`	A dynamic action that sets the required state of a field. **Attributes**: • **`name`** — The name of a field on the current screen. • **`test`** — An expression that is evaluated by the expression handler to determine whether the field will be required or optional. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.
`<set-values>`	A dynamic action that sets the selectable values of a drop-down list. **Attributes:** • **`name`** — The name of a field on the current screen. • **`values`** — An expression that is evaluated by the expression handler to determine the selectable values that are allowed. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.

Element	Description
`<set-value>`	A dynamic action that sets the value of the field **Attributes:** • **`name`** — The name of a field on the current screen. • **`value`** — An expression that is evaluated by the expression handler to determine the value. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.
`<call>`	A dynamic action that executes a function. No screen components are updated. **Attributes:** • **`function`** — A function that is evaluated by the expression handler. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.
`<refresh>`	A dynamic action that refreshes a DataGrid or field by retrieving the relevant values from the server. **Attributes:** • **`name`** — The name of the data source or field to refresh. • **`if`** (optional) — An expression that, if specified, is evaluated by the expression handler to determine whether the action executes.

Customizable Menus

Menu Customization

Menus are configured using an XML file that is stored on a content server at **`/sasdav/Products/SASEntCaseManagement/SASEntCaseManagement6.2/Menus/MainNavigationMenu.xml`**

A default version of the menu is shipped with SAS Enterprise Case Management. However, you can customize the menu by following these steps:

1. From the **Administration** tab, download the existing MainNavigationMenu.xml file.
2. Make the desired changes.
3. Upload the new version of the file.
4. Log off of SAS Enterprise Case Management.
5. Log on to SAS Enterprise Case Management. The menu changes should appear.

Example Menu Definition

The following is an example of a menu definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<menu-definition>
<menu id="mainNavigationMenu">

<!-- Cases -->
 <item visible="IsCapable('Search Cases')" id="mainMenu_Cases">
  <label><message key="cases.txt" /></label>
   <menu>
     <item id="mainMenu_Cases_CaseSearch">
       <label><message key="search.label.txt" /></label>
         <url><eval>contextPath</eval>/CaseSearch.do</url>
     </item>
     <item visible="IsCapable('Create Case')" id="mainMenu_Cases_NewCase">
       <label><message key="new.case.txt" /></label>
         <url>javascript: AppMenu.newCase()</url>
     </item>
     <item visible="IsCapable('Print Case')" id="mainMenu_Cases_GeneratedReports">
      <label><message key="case.view.print.report.action.label" /> </label>
       <url><eval>contextPath</eval>/CaseReports.do?command=viewPrintReports</url>
     </item>
   </menu>
    </item>

 </menu>
</menu-definition>
```

Menu Definition Structure

The menu XML file has this general structure:

```
menu-definition
       menu (top-level)
            item (top-level)
                    menu (second-level)
                           item (second-level)
                           item (second-level)
            separator
            item (top-level)
                    menu (second-level)
                           item (second-level)
                           separator (second-level)
                           item (second-level)
            item (top-level)
                    menu (second-level)
                           item (second-level)
                           item (second-level)
```

XML Elements

Element	Description
`<menu-definition>`	The document root element. There can be only one **`<menu-definition>`** element per document. No attributes.
`<menu>`	A menu encapsulates zero or more menu items. **Attributes** • **`id`** — A unique identifier for the menu. This attribute is optional except for the menu that represents the Main Navigation Menu.
`<separator/>`	Separators are rendered for the menu bar and pop-up menus. The Main Navigation Menu and Main Help Menu render separators. No attributes.
`<item>`	Represents a clickable menu item. The label for the menu item is defined by a nested **`<label>`** element. The URL (if any) is defined by a nested **`<url>`** element. Parameters for the URL are defined by nested **`<param>`** elements. **Attributes** • **`id`** — (optional but recommended) A unique identifier for the menu item. • **`visible`** — An expression that is evaluated to determine whether a menu item is visible. • **`embed`** — If set to true, the target URL is shown in the content area of SAS Enterprise Case Management in an inline browser window (iframe).
`<if>`	A conditional expression that must evaluate to true for its child menu elements to be evaluated. **Attributes** • **`test`** — (required) This expression must evaluate to true for any child elements to be evaluated.
`<label>`	Defines the label for the parent menu item. No attributes.

`<url>`	Defines the base URL for the menu item. No attributes.
`<image>`	Defines an image to display for the menu item, through either a child **`<url>`** element or a **`src`** attribute. To specify an image for a menu item by CSS class, instead of URL, use the **`<imageClass>`** element. **Attributes** • **`src`** – (optional) The URL of the image file for this menu item. Ignored if a child **`<url>`** element is specified. • **`width`** – (optional) The width of the image. • **`height`** – (optional) The height of the image.
`<imageClass>`	Defines an image to display for a menu item by CSS class name. To specify an image for a menu item by URL instead of CSS, use the **`<image>`** element. **Attribute** • **`class`** — (required) The class specifying the image for the menu item.
`<param>`	Defines a parameter for the menu item's URL. **Atrributes** • **`name`** — The name of the parameter.

Examples

Embedding an External URL

```
<item id="flowers" embed="true">
      <label>Search for Flowers</label>
      <url>http://www.google.com/search</url>
      <param name="q">flowers</param>
</item>
```

Embedding a SAS Application

```
<item id="sampleReport" embed="true">
      <label>Sample Report</label>
      <url><eval>GetSASStoredProcessPath()</eval></url>
      <param name="_odsstyle">Plateau</param>
      <param name="_program">
             /Products/SAS Intelligence Platform/Samples/Sample: Year to Date Budget
      </param>
</item>
```

Launching a URL in a New Browser Window

```
<item id="swwNewWindow" target="_blank">
      <label>Open SWW in New Window</label>
      <url>http://sww.sas.com</url>
</item>
```

Internal SAS Enterprise Case Management Link

```
<item id="tasks">
      <label><message key="taskList.title" /></label>
      <url><eval>contextPath</eval>/Welcome.do</url>
</item>
```

Using Directives

```
<item id="preferencesDirective" embed="true">
      <label>Preferences (directive)</label>
      <url><eval>contextPath</eval>/Director</url>
      <param name="_directive">SASPreferencesLogon</param>
</item>
```

BI Dashboard

```
<item id="biDashboard" embed="true">
      <label>Sample Dashboard</label>
      <url><eval>contextPath</eval>/Director</url>
      <param name="_directive">PortletDisplayDashboard</param>
      <param name="dashboard">
           SBIP://METASERVER/Products/SAS Enterprise Case Management/
           Ent Case Mgmt Configuration 6.2/
           Dashboards/ECM Dashboard.dcx(Dashboard)
      </param>
      <param name="frameWidth">500</param>
      <param name="frameHeight">500</param>
      <param name="srcapp">saas</param>
</item>
```

Conditionally Hiding a Menu Item

```
<item id="users" visible="IsUserInRole('ROLE_NAME')">
      . . .
</item>
```

Working with Menu Definitions

View Menu Definitions

To view menu definitions, select **Menu Definitions** from the **Administration** menu and select the menu definition that you want to view.

Edit the Menu Definition

To edit a menu definition:

1. Click the **Administration** tab and select **Menu Definitions**.
2. Click **Download Menu Definition** from the pop-up menu for MainNavigationMenu.xml.
3. Edit the file with a text editor or with your favorite XML editor.
4. Upload the changes. For more information, see "Upload the Menu Definition" on page 121.

Upload the Menu Definition

To upload a menu definition:

1. From the **Administration** menu, select **Menu Definitions**.
2. From the action menu for the menu definition that you want to update, click **Upload Menu Definition**. The Upload Menu Definition File window appears.
3. Enter the path to the file, or click **Browse** to navigate to it.
4. Click **Validate XML**. Any warnings or errors found in the menu definition file are displayed.
5. When you are satisfied with the results, click **Upload Menu Definition**.

Note: You will need to log out and log back in to have the new menu definition take effect.

If problems are detected with the uploaded menu during validation, see "Main Navigation Menu Errors When Logging in" on page 247.

Customize the Banner Title Content

Banner titles consists of two parts: the application title and the page title. For pages that are rendering Custom Page Builder content, each of these parts can have their default values overridden by defining an **`<app-title>`** element and a **`<title>`** element inside the **`<screen>`** element. The following is an example of overriding an incident page:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ui-definition SYSTEM "uiDefinition.dtd">
<ui-definition id="incidentDetails" type="incident">
  <title>Incident</title>
  <screen id="incident">
    <app-title>SAS Financial Crimes Suite</app-title>
    <title>
      <eval>GetLabel(INCIDENT.INCIDENT_TYPE_CD, 'RT_INCIDENT_TYPE')</eval>:
      <eval>INCIDENT.INCIDENT_DESC</eval>
    </title>
```

In this example, the application title will change from "SAS Enterprise Case Management" to "SAS Financial Crimes Suite," and the page title will change from "Incident – Incident ID" to something like "ATM Fraud: My description text." This allows for configuring the page title to show any field or combination of fields from the incident.

Custom Page Builder: Creating Custom Help

You can customize the help tags for SAS Enterprise Case Management. by adding context-sensitive help tags on the **<screen>** and **<help-text>** elements. However, using both the help attribute for the **<screen>** element (external help) and the **<help-text>** element (inline help) on the same page is not supported. If you configure external help and inline help on the same screen, then a warning displays when you upload the screen definition file. In addition, only the external help is available for the screen.

The following example demonstrates how to link to a web page using the **help** attribute for the **<screen>** element:

Example Code 6.1 *Linking to an External Help Page*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ui-definition SYSTEM "uiDefinition.dtd">
<ui-definition id="actionplan-ui-def" type="ActionPlan">
  <title>ActionPlan UI Definition</title>
  <screen id="actionPlan" help="http://www.sas.com">
   ...
  </screen>
</ui-definition>
```

You can add context sensitive help tags on the **<screen>** and **<help-text>** elements. This example demonstrates how to add inline help text using the **<help-text>** element:

Example Code 6.2 *Inline Help Text*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ui-definition SYSTEM "uiDefinition.dtd">
<ui-definition id="issue-ui-def" type="Issue">
  <title>Issue UI Definition</title>
  <screen id="issue">
   <help-text>
    <![CDATA[
      <h1>Sample Custom Help</H1
        <ul>
         <li><b>Field #1:</b>Please help me with this field...</li>
         <li><b>Field #2:</b>Please help me with this field...</li>
         <li><b>Field #3:</b>Please help me with this field...</li>
       </ul>
       <p>This is example help text.</p>
    ]]>
  </help-text>
    ...
  </screen>
</ui-definition>
```

Expressions and Functions

About Expressions and Functions

An expression is any valid set of literals, variables, operators, and functions that evaluates to a single value. Quite a few element attributes support expressions, including the following examples:

- the **`test`** attribute of **`<validation>`** and **`<if>`** elements
- the **`value`** attribute of **`<set>`** elements and dynamic actions
- the **`visible`** and **`required`** attributes of **`<field>`** and **`<section>`** elements

Expressions can reference any valid field defined in the specific UI definition type. These fields include the fields of the primary object (for example, the Action Plan that is being edited) in addition to fields for any useful secondary objects (for example, the parent Issue of the Action Plan that is being edited). Temporary and derived fields are also supported. Expressions can use these fields in combination with the usual arithmetic operators (add, subtract, multiply, divide), relational operators (for example, a > b, c = 10, d <= 20), and logical operators (for example, a and b or c and d and not e). The use of expressions and operators enables you to express business rules directly in the user interface definition files.

In addition to arithmetic, relational, and logical operators, there is a set of functions that you can use in the expressions. Some of these functions could be used for field validation (for example, empty(), validNumber(), validDate()). Also, you might use functions to construct derived fields (for example, concat(), length(), if()). Finally, you might use functions to access data sources (for example, getCodeTableLabelValues(), getCodeEnumerationLabelValues(), or getAuxOptionLabelValues()).

For more information about expressions and functions, see the Custom Page Builder documentation found on the **Administration** menu of SAS Enterprise Case Management.

Lazy Initialization for Data Displayed on a Tab

Overview

When a Custom Page Builder window is first displayed, the data for the SAS Enterprise Case Management entity (if applicable), as well as any data loaded in the window's **`<initialize>`** block, is loaded before the page is rendered. This is usually sufficient. However, there might be times when you also want to load and display a large amount of data from an external data source but don't want to delay the rendering of the page. One solution for efficiently loading a large amount of external data is to use lazy initialization for **`<tab>`** sections.

The **`<tab>`** element allows an optional **`<initialize>`** child. If a child **`<initialize>`** block exists, it is not executed when the page first loads. Rather, it is executed when that particular tab is first selected. On selection, the tab's contents are masked and the **`<initialize>`** block is run asynchronously, allowing the user to navigate to other tabs and other sections of the page while the tab is loading. After the

<initialize> block completes, the tab's contents are unmasked and visible to the user. All fields on that tab are refreshed to reflect any data loaded through expressions in the **<initialize>** block.

Best Practices

In an **<initialize>** block, load only data that is displayed on the tab containing that **<initialize>** block. Only this tab's child fields are refreshed when the initialization completes.

Load only data that will be Read-Only in a tab's **<initialize>** block. The reason is that there is no guarantee that a user will activate any given tab. Therefore, a tab's **<initialize>** block might never run.

Example

The following is an example of a tab that loads data from an external source. The fields and DataGrid are displaying data that is fetched by the user-defined function "C_LoadPatientData()".

```
<tab id="lazy_initialized_tab">

   <label>Patient Data</label>

    <initialize>
      <set name="TEMP.LAZY_LOAD_SUCCESS"
         value="C_LoadPatientData('TEMP.PATIENT_DATA'))"/>
    </initialize>

    <field name="TEMP.PATIENT_DATA.PHYSICIAN" type="readonly">
      <label>Physician Name</label>
    </field>
    <field name="TEMP.PATIENT_DATA.HOSPITAL" type="readonly">
      <label>Hospital</label>
    </field>
    <field name="TEMP.PATIENT_DATA.DATE" type="readonly">
      <label>Record Date</label>
    </field>

    <datagrid name="GRID_LAZY_PATIENT_INFO" selectedKeyField="ID">
      <label>Patients</label>
      <datastore id="patient_info_store"
        data="C_GetValueFromContext('TEMP.PATIENT_DATA.PATIENTS')"/>
      <datagrid-column name="NAME">
        <label>Patient Name</label>
      </datagrid-column>
      <datagrid-column name="AGE">
        <label>Age</label>
      </datagrid-column>
      <datagrid-column name="WEIGHT">
        <label>Weight</label>
      </datagrid-column>
      <datagrid-column name="DIABETIC">
        <label>Diabetic</label>
      <datagrid-renderer-ref id="sas_booleanRenderer"/>
     </datagrid-column>
```

```
    </datagrid>

</tab>
```

Customization Examples

How to Customize the User Interface Definition Files

The following steps provide a high-level overview of the customization process for user interface definition files:

1. Log on to SAS Enterprise Case Management and download the user interface definition that you want to edit. For more information, see "Working with User Interface Definitions" on page 100.
2. Edit the user interface definition file by adding allowable XML elements to the file. For example, to add a new field to a screen, you add the **`<field>`** element. There are several customization examples provided in this chapter. For more information about supported XML elements, see "Valid XML Elements and Descriptions for User Interface Definitions" on page 101.

   Note: You do not need to change the application source code or the database. By default, almost every screen provides six numeric, alphanumeric, Boolean, currency, date, and drop-down fields. You can enable and label those fields as required. For more information about required fields, see "About Required and Non-Required Fields" on page 125.
3. Enter validation code to validate user entries in the user interface definition files.
4. Upload any values that you added to the user interface definition files (for example, an additional value in a drop-down list).
5. To use the new field or structure in the user interface definition files, upload the changed version of the user interface definition file. For more information, see "Upload the User Interface Definition" on page 100.
6. To make the new field available to users of the SAS reporting tools, you must give it a meaningful name that you can use within the target reports and the list of fields available to be included in reports. You can do this by using SAS Information Map Studio. In SAS Information Map Studio, you must select the referring field from the list of available fields in the database and give it a name. For more information, see the Help for SAS Information Map Studio, accessible within the product.

About Required and Non-Required Fields

Certain fields are required by design, and you can specify optional fields as required fields. Although you cannot specify required fields as optional, you can hide these fields. For more information, see "Example: Hiding a Required Field" on page 126. When you set a field as required, the system automatically checks whether the user has provided data. If a drop-down list is used, then the system checks whether the user made a selection. If the user does not provide the data, or if the user does not select a value from the drop-down list, then the system automatically displays an error message when the user tries to save the information. You can customize the text for error messages. For more information, see "Customize Error Messages" on page 126. If any fields within a section are required, then the section heading is marked as required. For example:

```
<section id="details">
     <label><message key="application.details.txt"/></label>
       <field name="issueShortDesc" type="string" required="true">
     <label><message key="issue.field.issueShortDesc
      .displayName.txt"/></label>
       </field>

     <field name="issueId" type="string" required="true">
       <label><message key="issue.field.issueId.displayName.txt"/>
       </label>
     </field>
</section>
```

Note: If you specify that a non-required field is required in a user interface definition file, the existing data loaders do not check the newly required field to ensure that there is information entered for it. For example, if you use the Issues data loader and you specify that a field on an issue is now required, then the data loader will load that field without any values because it does not know that you changed a non-required field to be required.

Customize Error Messages

SAS Enterprise Case Management provides you with customMessages.properties files for each supported language. You can customize error messages by adding the error messages from the server.properties file to the customMessages.properties file. For example:

```
errors.required.fmt.txt="{0}" is required.
```

Example: Hiding a Required Field

You can hide fields that are required by the database design by using the **`visible`** and **`default`** attributes on the **`<field>`** element. For example, to hide the Issue ID field:

```
<field name="issueId" type="string" required="true"
    default="toString(issueRk)" visible="false">
  <label><message key="issue.field.issueId.displayName.txt"/>
  </label>
</field>
```

In this example, the Issue ID field requires a unique value for each object. Note that you can also use functions on the default attribute. For example:

```
default="concat('ABC-', issueRk)"
```

Example: Specifying a Read-Only Field

Use the **`readonly="true"`** attribute to specify that a field is Read-Only. The default value of the **`readonly`** attribute is **`false`**. The following example demonstrates how to specify a Read-Only field:

```
<field name="sourceSystemCd" type="dropdown" required="true"
    default="'MON'" readonly="true">
  <label><message key="issue.field.sourceSystemCd
    .displayName.txt"/></label>
</field>
```

Example: Specifying the Number of Decimal Digits

To specify the number of decimal digits that should be used when formatting a number, use the **`decimal-digits`** attribute on the **`<field>`** element. The following example demonstrates how to specify 2 decimal digits on the Loss Amount field:

```
<field name="auxNum1" type="number" decimal-digits="2" required="true">
   <label>Loss Amount</label>
</field>
```

Example: Validating Dates

To validate date entries, use the **`type`** attribute to specify that the input control is a date. Then use the **`min`** or **`max`** attributes with a function on the **`<field>`** element for determining the date. The following example demonstrates how to specify that the minimum value entered for a date must be today's date:

```
<field name="targetDt" type="date" min="today()">
        <label><message key="issueEx.field.targetDt.displayName.txt"/>
        </label>
</field>
```

Example: Specifying Drop-Down Lists and Radio Buttons

To specify whether a drop-down list or radio buttons are used for single-select fields, use the **`type`** attribute on the **`<field>`** element. The following example demonstrates how to specify a drop-down list:

```
<field name="issuePriorityTypeCd" type="dropdown" required="true">
    <label><message key="issue.field.issuePriorityTypeCd.displayName.txt"/>
    </label>
</field>
```

The following example demonstrates how to specify radio buttons:

```
<field name="issuePriorityTypeCd" type="radio" required="true">
    <label><message key="issue.field.issuePriorityTypeCd.displayName.txt"/>
    </label>
</field>
```

Example: Specifying a Text Area and a Text Field

To specify whether a string field appears as a text area, use the **`type`** attribute on the **`<field>`** element. Using **`type="textarea"`** creates a text area on your form. In addition, you can specify the number of rows contained in a text area by using the **`rows`** attribute. The following example demonstrates how to specify a text area with 6 rows:

```
<field name="issueDesc" type="textarea" rows="6" required="true">
   <label><message key="issue.field.issueDesc.displayName.txt"/></label>
</field>
```

You can also use the **`type`** attribute to specify that a string field appears as a text field with a specified length for the field. Using the **`type="string"`** attribute creates a text field on your form. Using the **`length`** attribute specifies the length of the text field. The following example demonstrates how to specify a text field that is 32 characters long:

```
<field name="referenceNo" type="string" length="32" required="true">
```

```
    <label><message key="issue.field.referenceNo.displayName.txt"/></label>
</field>
```

Example: Creating Custom Filtered Subjects

Specify the criteria to use when filtering results by using the **<filter>** element. This enables you to specify how to filter the results of a search on subjects assigned to a case as well as how to filter the search results of cases assigned to a subject. You can create multiple filtered tables to display on the same page. Each filtered table must be separated by a **section** attribute.

In addition, you can specify the fields to display within the table and customize the output accordingly. The following example demonstrates how to specify multiple filtered tables in one UI definition:

Note: The **section** attribute is required and must contain a unique ID. The **<field>** tag is also required and must contain a unique name.

```
<section id="TypeCat" expanded="true">

       <label>Type and Category </label>



      <field type="component" required="false"
          component-name="FILTERCASEPARTIESTABLE"
          name="TypeCategoryTable">

             <param name="TEMP.DEFAULT_DISPLAY_SUBJECT_STATUS" value="'P'"/>

             <param name="TEMP.PARTY.RELATIONSHIP_TYPE_CD" value="'IND'" />
             <param name="TEMP.PARTY.RELATIONSHIP_CAT_CD" value="'REG'" />
             <param name="TEMP.PARTY.RELATIONSHIP_SUB_CD" value="'CTR'" />

             <param name="relationship_type_table" value="'X_RT_RELATION_TYPE'"/>
             <param name="party_type_table" value="'RT_PARTY_TYPE'"/>
             <param name="party_category_table" value="'RT_PARTY_CATEGORY'"/>
             <param name="field" value="'PARTY_ID'" />
             <param name="field" value="'PARTY_FULL_NM'" />
             <param name="field" value="'SOURCE_SYSTEM_CD:RT_SOURCE_SYSTEM'"/>
             <param name="field" value="'PARTY_TYPE_CD:RT_PARTY_TYPE'"/>
             <param name="field" value="'INDIVIDUAL_FLG:X_RT_INDIVIDUAL'"/>
             <param name="field" value="'TEMP.PARTY.RELATIONSHIP.DESCRIPTION'"/>
             <param name="field" value="'CREATE_DTTM:datetime'"/>

             <param name="filter" value="'RELATIONSHIP_TYPE=S'"/>
             <param name="filter" value="'TEMP.PARTY.RELATIONSHIP_TYPE_CD=IND'"/>
             <param name="filter" value="'TEMP.PARTY.RELATIONSHIP_CAT_CD=REG'"/>
      <param name="filter" value="'TEMP.PARTY.RELATIONSHIP_SUB_CD=CTR'"/>
       </field>
       </section>
```

In this example, a table will be displayed with a relationship code of S, a party type INV, a category of REG, and a subcategory of CTR.

The following is an example of the second table in the window:

```
<section id="TypeCat" expanded="true">
```

```
 <label>Type and Category </label>



<field type="component" required="false"
    component-name="FILTERCASEPARTIESTABLE"
    name="TypeCategoryTable">

       <param name="TEMP.DEFAULT_DISPLAY_SUBJECT_STATUS" value="'P'"/>

       <param name="TEMP.PARTY.RELATIONSHIP_TYPE_CD" value="'IND'" />
       <param name="TEMP.PARTY.RELATIONSHIP_CAT_CD" value="'REG'" />
       <param name="TEMP.PARTY.RELATIONSHIP_SUB_CD" value="'CTR'" />

       <param name="relationship_type_table" value="'X_RT_RELATION_TYPE'"/>
       <param name="party_type_table" value="'RT_PARTY_TYPE'"/>
       <param name="party_category_table" value="'RT_PARTY_CATEGORY'"/>
       <param name="field" value="'PARTY_ID'" />
       <param name="field" value="'PARTY_FULL_NM'" />
       <param name="field" value="'SOURCE_SYSTEM_CD:RT_SOURCE_SYSTEM'"/>
       <param name="field" value="'PARTY_TYPE_CD:RT_PARTY_TYPE'"/>
       <param name="field" value="'INDIVIDUAL_FLG:X_RT_INDIVIDUAL'"/>
       <param name="field" value="'TEMP.PARTY.RELATIONSHIP.DESCRIPTION'"/>
       <param name="field" value="'CREATE_DTTM:datetime'"/>

       <param name="filter" value="'RELATIONSHIP_TYPE=P'"/>
       <param name="filter" value="'TEMP.PARTY.RELATIONSHIP_TYPE_CD=IND'"/>
</field>
 </section>
```

In this example, a table will be displayed with a relationship code of P and a party type of INV. Because category and subcategory are optional, they are not present.

Dynamic Conditional Logic in User Interface Definition Files

You can perform the following types of dynamic actions when a field value changes:

- **`set-visible`** — shows or hides another field or section.

  This example demonstrates how to define conditional logic within a page. In this example, the Name of Spouse text field appears only if Y is selected in the Married drop-down list.

  Example: Using the **`set-visible`** action

```
<field name="auxOptionCd1" type="dropdown"
   values="getAuxOptionLabelValues('issue', 'auxOptionCd1')">
   <label>Married</label>
   <on-change>
      <set-visible name="auxStr1" test="auxOptionCd1 = 'Y'"/>
   </on-change>
</field>

<field name="auxStr1" type="string" visible="auxOptionCd1 = 'Y'">
   <label>Name of Spouse</label>
</field>
```

Note: The value that is specified for the **test** attribute of the **<set-visible>** element will most likely be the same as the value that is specified for the **visible** attribute of the target **<field>** element.

- **set-required** — makes another field required or optional.

  This example demonstrates how to define conditional logic within a page. It specifies that the Justification field is required if the value entered for the Loss Amount is greater than 1 million.

  Example: Using the **set-required** action

```
<field name="auxNum1" type="number" decimal-digits="2" required="true">
   <label>Loss Amount</label>
   <on-change>
      <set-required name="auxStr2" test="auxNum1 > 1000000"/>
   </on-change>
</field>

<field name="auxStr2" type="textarea" required="auxNum1 > 1000000">
    <label>Justification</label>
</field>
```

  Note: The value that is specified for the **test** attribute of the **<set-required>** element will most likely be the same as the value that is specified for the **required** attribute of the target **<field>** element.

- **set-values** — updates the selectable values of a drop-down list. Although radio buttons currently allow **<set-values>** in the DTD, this feature is not currently implemented. It is planned for a future release.

  Example: Using the **set-values** action

```
<initialize>
  <!-- create a filter expression to be used in the filterLabelValue
   ()function -->
  <set name="TEMP.myFilterExpr" value="'if(auxNum2 > 1000000,
    value = &quot;high&quot;, true)'"/>
</initialize>
:
<field name="auxNum2" type="number" decimal-digits="2" required="true">
   <label>Loss Amount</label>
   <on-change>
      <set-values name="auxOptionCd2"
        values="filterLabelValues(getAuxOptionLabelValues
          ('risk', 'auxOptionCd2'), TEMP.myFilterExpr)"/>
   </on-change>
</field>

<field name="auxOptionCd2" type="dropdown" required="true"
    values="filterLabelValues(getAuxOptionLabelValues
    ('risk', 'auxOptionCd2'),TEMP.myFilterExpr)">
  <label>Risk</label>
</field>
```

  Note: The value that is specified for the **values** attribute of the **<set-values>** element will most likely be the same as the value that is specified for the **values** attribute of the target **<field>** element.

Example: Creating a Custom Function

You can write your own custom functions and reference them in the user interface definition expressions. To create a custom function for use in the user interface definition files:

1. Write the Java code that represents the custom function (and compile it into a class). For example:

```
package com.sas.cpb.customFunctions;

import com.sas.solutions.cpb.expr.function.Function;
import com.sas.solutions.cpb.runtime.EvaluationException;

/**
 * A custom function to uppercase a String.
 */
public class UpperFunction extends Function {

    /**
     * Returns the number of arguments required by the function.
     * This function expects one argument.
     */
    @Override
    public int getArgumentCount() {
      return 1;
    }

    /**
     * Evaluates the function using arguments specified
     * in the XML file.
     *
     * @param args the arguments passed to the
     * function (specified in the XML)
     * @throws EvaluationException
     */
    @Override
    public Object evaluate(Object[] args) throws
   EvaluationException {
        if (args[0] != null) {
            return args[0].toString().toUpperCase();
        }
        return null;
    }
}
```

2. Register the custom function in the user interface definition file. Before the **`<screen>`** element, insert the **`<function>`** tag in the user interface definition file. For example:

```
<function name="C_upper" qualified-class-name="com.sas.cpb.customFunctions
  .UpperFunction"/>
```

Note: You must prefix the custom function names with “C_” to prevent naming conflicts with the Custom Page Builder standard components.

Creating a Custom Component

You can create your own custom components for use in the screen definition. Components are graphical widgets that appear on the screen (for example, the color chooser). A custom component is one that you can write to supplement the components that are provided in SAS Enterprise Case Management.

Follow these steps to create a custom component:

1. Write the Java code that represents the custom component. For example:

```
package com.sas.cpb.customComponents;

import java.io.IOException;
import java.util.Locale;
import java.util.Map;
import java.util.ResourceBundle;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.PageContext;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import com.sas.solutions.cpb.runtime.EvaluationException;
import com.sas.solutions.cpb.runtime.UIContext;
import com.sas.solutions.cpb.runtime.component.CustomComponent;
import com.sas.solutions.cpb.screendefs.Field;
import com.sas.solutions.cpb.server.ApplicationProperties;
import com.sas.solutions.cpb.util.FieldUtil;
import com.sas.solutions.cpb.web.runtime.FieldRenderContext;
import com.sas.solutions.cpb.web.util.HTMLUtil;
import com.sas.solutions.cpb.web.util.RequestUtil;

/**
* An example custom component to allow the user
* to choose a color (and store it
* in the UI context as a String in the form "#RRGGBB").
*/
public class ColorChooserComponent extends com.sas.solutions.cpb.runtime.component.
   CustomComponent {

private static final Log log = LogFactory.getFactory().getInstance
   (ColorChooserComponent.class);

   /**
   * Renders the HTML for this component.
   *
   * @param value the current value of the field being rendered by the component
   * @param field the field being rendered by the component
   * @param readOnly if the field should be rendered as read-only
   * @param formName the name of the HTML form that will contain the field
   * @param uiContext the current UIContext
```

```
* @param pageContext the current PageContext
* @param out the output stream to use for writing HTML
* @param locale the clients locale
*/

/**
* Update the UI context with the field value.
*
* @param field the field being rendered by the component
* @param uiContext the current UIContext
* @param parameters a map of parameters specified for the field (if any)
* @param request the HttpServletRequest
*/
@Override
public boolean updateContext(Field field, UIContext uiContext, Map<String, Object>
      parameters, HttpServletRequest request) {
   // this should match the name of the HTML input we rendered above
   String fieldName = FieldUtil.formatFieldName(field.getName());
   // retrieve the new value from the request
   String value = request.getParameter(fieldName);
   // If the value is null, then it wasnt submitted, so dont clear the
   // value in the UI context.
   if (value != null) {
      uiContext.setValue(fieldName, value);
      return true;
   }
   return false;
}

@Override
public void render(FieldRenderContext renderContext) throws IOException,
      ServletException, EvaluationException {
   try {
      Field field = renderContext.getField();
      UIContext uiContext = renderContext.getUIContext();
      Object value = renderContext.getFieldValue();
      JspWriter out = renderContext.getOut();
      String fieldName = field.getUnqualifiedFieldName();
      PageContext pageContext = renderContext.getPageContext();
      Locale locale = renderContext.getRequest().getLocale();

      // retrieve the application resource bundle
      final ResourceBundle bundle = ApplicationProperties.getBundle();
      // the name of the field, properly formatted for use as the name of an
         HTML String fieldName = FieldUtil.formatFieldName(field.getName());
      // get the localized label for the field (from one of the properties files)
      String label = FieldUtil.getLabel(field, uiContext, true);
      // if the current value isnt a String, dont use it
      String color = value instanceof String ? (String) value : "";
      // output the HTML for this component
      out.println("<div class=\"section\" style=\"padding:5px\">");
      out.println("<div id=\"colorPicker" + fieldName + "\" ></div>");
      out.println("<script type=\"text/javascript\">");
      out.println("function colorClicked(td) {");
      out.println(" var color = td.getAttribute(clr);");
      out.println(" elementById(\"" + fieldName + "\").value = color;");
```

```
            out.println(" elementById(\"colorBox" + fieldName + "\").style.background =
                color;");
            out.println("}");
            out.println("var colorCodes = [FF, CC, 99, 66, 33, 00]; ");
            out.println("var codeCount = colorCodes.length;");
            out.println("var html = \"<table>\";");
            out.println("for (var i = 0; i < codeCount; i++) {");
            out.println(" html += \"<tr>\";");
            out.println(" for (var j = 0; j < codeCount; j++) {");
            out.println(" for (var k = 0; k < codeCount; k++) {");
            out.println(" var color = \"#\" + colorCodes[i] + colorCodes[j] +
                colorCodes[k];");
            out.println(" html += \"<td style=cursor:default; background: "
                  + "\" + color + \" onclick=colorClicked(this)"
                  + " clr=\" + color + \">   </td>\";");
            out.println(" }");
            out.println(" } ");
            out.println(" html += \"</tr>\";");
            out.println("}");
            out.println("html += \"</table>\";");
            out.println("elementById(\"colorPicker"
              + fieldName + "\").innerHTML = html;");
            out.println("</script>");
            out.println("<br/>");
            // output the required and error indicators
            HTMLUtil.writeFieldIndicators(fieldName,
           FieldUtil.isRequired(field, uiContext),
            false, // this field will never be confidential
                  RequestUtil.isFieldValueInvalid(fieldName, pageContext), locale, out);
            // output the label
            out.println("<span class=\"fieldlabel\">" + label + "</span>");
            // display the previously selected color
            String boxImgUrl = bundle.getString("application.colorbox.image");
            out.print("<img align=\"absmiddle\" src=\""
                 + boxImgUrl + "\" id=\"colorBox" +
                 fieldName + "\" style=\"background:" + color
                  + "; vertical-align: top;border: 0px solid black\" />" + " ");
            // output an HTML input so that it will be submitted with the form
            out.println("<input id=\"" + fieldName
                  + "\" type=\"text\" name=\"" + fieldName
                  + "\" value=\"" + color + "\" />");
            out.println("</div>");
         } catch (EvaluationException e) {
            log.error("Error rendering component.", e);
         } catch (JspException e) {
            // TODO Add your JSP catch errors here
            e.printStackTrace();
         }

      }
   }
```

2. Compile the Java file and place the resulting class somewhere in the application server's classpath.

3. Register the custom component in the screen definition file. Before the **`<screen>`** element, insert the **`<component>`** tag in the screen definition file. For example:

```
<component name="c_colorPicker"
qualified-class-name="com.sas.cpb.customComponents.ColorChooserComponent" />
```

Note: You must prefix the custom component names with "c_" to prevent naming conflicts with the Custom Page Builder standard components.

4. Reference the custom component in a **`<field>`** element. For example:

```
<field name="auxStr3" type="component" component-name="c_colorPicker">
<label>Choose a color</label>
</field>
```

DataStores and DataGrids

Overview

DataStores are objects that represent tabular data. They can be defined as part of a DataGrid or defined separately and used by multiple DataGrids.

DataGrids allow for a generic way of defining and displaying tabular data on a Custom Page Builder window. DataGrids use DataStores to define their data. However, DataGrids further define which actions can be performed on the data and which data columns are actually displayed. They also define a filter that can be used to show a subset of the data.

Example: A DataGrid with a Single DataStore

This example shows a DataGrid with its own DataStore. The DataStore contains all incidents that have not been assigned to a case.

```
<datagrid name="ALERTS.GRID" selectedKeyField="INCIDENT.INCIDENT_RK">
<label><message key="aml.available.alerts.txt" /></label>

<datastore id="ALERTS.STORE"
   data="GetEntityListAsMapList(GetUnassignedIncidents())" />

<datagrid-column name="INCIDENT.INCIDENT_ID" >
<label><message key="field.incident.incident_id.header.txt"/> </label>
<datagrid-renderer-ref id="sas_entityLinkRenderer"
   args="Incident:INCIDENT.INCIDENT_RK"/>
</datagrid-column>
<datagrid-column name="INCIDENT.SOURCE_SYSTEM_CD">
<label><message key="field.incident.source_system_cd.header.txt"/> </label>
<datagrid-renderer-ref id="sas_refTableRenderer" args="RT_SOURCE_SYSTEM"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'RT_SOURCE_SYSTEM'"/>
</datagrid-column-sorter>
</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_TYPE_CD">
<label><message key="field.incident.incident_type_cd.header.txt"/> </label>
<datagrid-renderer-ref id="sas_refTableRenderer" args="RT_INCIDENT_TYPE"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'RT_INCIDENT_TYPE'"/>
</datagrid-column-sorter>
```

```
</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_DESC">
<label><message key="field.incident.incident_desc.header.txt"/> </label>
</datagrid-column>
<datagrid-column name="INCIDENT.UPDATE_USER_ID">
<label>Updated by</label>
<datagrid-renderer-ref id="sas_userNameRenderer"/>
<datagrid-column-sorter name="userDisplayNameSorter"/>
<label>Updated by</label>
</datagrid-column>

<!--  This action will create a case with type equal to 'FIN'
and auto link any selected incidents in the grid. -->
<grid-action id="ALERTS.GRIDACTION"
output-destination="inline" render-type="select" >
<label>Create a case</label>
<url><eval>contextPath</eval>/rest/caseIncidents.json</url>
<image>
<url><eval>GetThemedImageURL("newform")</eval></url>
</image>
<param name="case_type_cd" value="'FIN'"></param>
</grid-action>

</datagrid>
```

Example: Two Filtered DataGrids Sharing a Single DataStore

This example shows two DataGrids sharing a single DataStore. The DataStore contains all incidents that have not been assigned to a case. Each DataGrid is using a filter to show a subset of the data in the DataStore. The first DataGrid shows only unassigned incidents that have an incident type of SIR. The second DataGrid shows only unassigned incidents that have an incident type of GEN.

```
<datastore id="ALERTS.STORE.STATIC"
data="GetEntityListAsMapList(GetUnassignedIncidents())" />

<datagrid name="ALERTS.GRID.SIR"
selectedKeyField="INCIDENT.INCIDENT_RK">
<label><message key="aml.available.alerts.txt" /> - Incident Report</label>

<datastore-ref id="ALERTS.STORE.STATIC" />

<datagrid-column name="INCIDENT.INCIDENT_ID" >
<label><message key="field.incident.incident_id.header.txt"/></label>
<datagrid-renderer-ref id="sas_entityLinkRenderer"
args="Incident:INCIDENT.INCIDENT_RK"/>
</datagrid-column>
<datagrid-column name="INCIDENT.SOURCE_SYSTEM_CD">
<label><message
key="field.incident.source_system_cd.header.txt"/>
</label>
<datagrid-renderer-ref id="sas_refTableRenderer"
args="RT_SOURCE_SYSTEM"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'RT_SOURCE_SYSTEM'"/>
</datagrid-column-sorter>
```

```
</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_TYPE_CD">
<label>
<message key="field.incident.incident_type_cd.header.txt"/>
</label>
<datagrid-renderer-ref id="sas_refTableRenderer"
args="RT_INCIDENT_TYPE"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="’RT_INCIDENT_TYPE’"/>
</datagrid-column-sorter>

</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_DESC">
<label>
<message key="field.incident.incident_desc.header.txt"/>
</label>
</datagrid-column>
<datagrid-column name="INCIDENT.UPDATE_USER_ID">
<label>Updated by</label>
<datagrid-renderer-ref id="sas_userNameRenderer"/>
<datagrid-column-sorter name="userDisplayNameSorter"/>
</datagrid-column>

<filter value="INCIDENT.INCIDENT_TYPE_CD = 'SIR'" />

<grid-action id="ALERTS.GRIDACTION.SIR" output-destination="inline"
render-type="select" >
<label>Create a case</label>
<url><eval>contextPath</eval>/rest/caseIncidents.json</url>
<image>
<url><eval>GetThemedImageURL("newform")</eval></url>
</image>
<param name="case_type_cd" value="'FIN'"></param>
</grid-action>

</datagrid>

<datagrid name="ALERTS.GRID.GEN" selectedKeyField="INCIDENT.INCIDENT_RK">
<label><message key="aml.available.alerts.txt" /> - Generic</label>

<datastore-ref id="ALERTS.STORE.STATIC" />

<datagrid-column name="INCIDENT.INCIDENT_ID" >
<label>
<message key="field.incident.incident_id.header.txt"/>
</label>
<datagrid-renderer-ref id="sas_entityLinkRenderer"
args="Incident:INCIDENT.INCIDENT_RK"/>
</datagrid-column>
<datagrid-column name="INCIDENT.SOURCE_SYSTEM_CD">
<label>
<message key="field.incident.source_system_cd.header.txt"/>
</label>
<datagrid-renderer-ref id="sas_refTableRenderer"
args="RT_SOURCE_SYSTEM"/>
<datagrid-column-sorter name="refTableSorter">
```

```
   <param name="referenceTable" value="'RT_SOURCE_SYSTEM'"/>
</datagrid-column-sorter>
</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_TYPE_CD">
<label>
<message key="field.incident.incident_type_cd.header.txt"/>
</label>
<datagrid-renderer-ref id="sas_refTableRenderer"
args="RT_INCIDENT_TYPE"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'RT_INCIDENT_TYPE'"/>
</datagrid-column-sorter>
</datagrid-column>
<datagrid-column name="INCIDENT.INCIDENT_DESC">
<label>
<message key="field.incident.incident_desc.header.txt"/>
</label>
</datagrid-column>
<datagrid-column name="INCIDENT.UPDATE_USER_ID">
<label>Updated by</label>
<datagrid-renderer-ref id="sas_userNameRenderer"/>
<datagrid-column-sorter name="userDisplayNameSorter"/>
</datagrid-column>

<filter value="INCIDENT.INCIDENT_TYPE_CD = 'GEN'" />

<grid-action id="ALERTS.GRIDACTION.GEN.2"
output-destination="inline" render-type="select" >
<label>Create a case</label>
<url><eval>contextPath</eval>/rest/caseIncidents.json</url>
<image>
<url><eval>GetThemedImageURL("newform")</eval></url>
</image>
<param name="case_type_cd" value="'FIN'"></param>
</grid-action>

</datagrid>
```

GridActions

A GridAction is an action that can be performed from a DataGrid, often manipulating the data displayed in it. A DataGrid can have any number of GridActions associated with it. SAS Enterprise Case Management includes useful actions that can be used as GridActions, or you can create your own.

GridActions can operate on the data from either a single row of a DataGrid or on multiple rows simultaneously (selected by the check box column). This is configured with the **`render-type`** attribute of the **`<grid-action>`** element.

`render-type` Value	**Description**
row	The action only operates on a single row. The action is rendered as a menu item in the context menu of each row.

`render-type` Value	**Description**
always	The action is always available and is displayed as a button in the DataGrid's tool bar.
select	The action operates on the rows that are selected through the check box column. This action is displayed as a button in the DataGrid's tool bar. However, it is enabled only when one or more rows are selected.

When a GridAction completes, it can update the UI in many ways. This is controlled by the **`output-destination`** attribute.

`output-destination` Value	**Description**
inline	The DataGrid containing the GridAction is refreshed when the GridAction completes. This value should be used when the URL being called manipulates the data that the DataGrid is displaying.
window	The browser window is redirected to the URL of the GridAction. If you are in a case, incident, party, report, or e-file editor, you are prompted to save any unsaved changes, if necessary, before being redirected.
new-window	A pop-up browser window displays the contents of the URL of the GridAction. *Note:* SAS Enterprise Case Management only supports running in a single browser window, so this output destination should be used only to display content outside of the application.
javascript	The value of the URL child element of the GridAction is interpreted as JavaScript instead of a URL, and is executed locally. Any parameters specified for the GridAction are ignored.

When the **`output-destination`** parameter is set to **`inline`**, **`window`**, or **`new-window`**, the values from the selected rows' **`key field`** column are appended to the URL as a parameter named **`keys`**. This allows the URL to know what entities or data the user has selected in the DataGrid.

When the **`output-destination`** parameter is set to **`javascript`**, useful data is made available to your JavaScript handler in the form of a variable named **`actionData`**. **`actionData`** is a map that contains the following fields:

actionData.rowIndex
: specifies the index of the row whose action menu you activated in the DataGrid.

actionData.gridName
: specifies the name of the DataGrid containing the GridAction that is clicked.

actionData.checkedRowKeys
: specifies the value of the **key field** cells for all checked rows, if any, as an array. This is the same set of values that is passed as a **keys** parameter to URL-based output destinations. If no rows are checked, this is an empty array.

actionData.selectedRowKey
: specifies the value of the **key field** cell in the row that is clicked.

actionData.selectedRowData
: is a map containing the value of all cells, both hidden and visible, in the row that is clicked. This field maps **datagrid-column** names to the cell values in that row.

Bulk Field Update Action

Overview

An action provided by SAS Enterprise Case Management that is useful as a GridAction is the Bulk Field Update action. If you are displaying a list of SAS Enterprise Case Management entities in a DataGrid, you can use this action to update one or more of their fields in bulk. Only the selected entities have their fields updated, and the DataGrid is refreshed when the transaction completes to reflect the new field values.

Note: Changes made by the Bulk Field Update GridAction take effect immediately.

URL

The URL for the Bulk Field Update action is ***<contextPath>*/controller/updateFields.json**.

Parameters

The Bulk Field Update action has the following parameters:

entityType (required)
: specifies the type of entities displayed in the DataGrid. The value should be either **'case'**, **'incident'**, **'party'**, **'rr'**, or **'efile'**.

Others (optional)
: Any other parameters are assumed to be fields to update and their new values.

Example

The following example shows a DataGrid displaying a list of all unassigned incidents. It has a GridAction that updates the INVESTIGATOR_USER_ID field of all selected incidents to a new value.

```
<datagrid name="INCIDENTS.GRID" selectedKeyField="INCIDENT.INCIDENT_RK">

   <label>Unassigned Incidents</label>
   <datastore id="INCIDENTS.GRID.STORE"
     data="GetEntityListAsMapList(GetUnassignedIncidents())"/>

   <datagrid-column name="INCIDENT.INCIDENT_ID">
      <label>Incident</label>
       <datagrid-renderer-ref id="sas_entityLinkRenderer"
         args="Incident:INCIDENT.INCIDENT_RK"/>
   </datagrid-column>
   <datagrid-column name="INCIDENT.INCIDENT_DESC">
      <label>Description</label>
```

```
      </datagrid-column>
      <datagrid-column name="INCIDENT.INCIDENT_DISPOSITION_CD">
         <label>Disposition</label>
         <datagrid-renderer-ref id="sas_refTableRenderer"
           args="X_RT_INCIDENT_DISPOSITION"/>
   <datagrid-column-sorter name="refTableSorter">
      <param name="referenceTable" value="'X_RT_INCIDENT_DISPOSITION'"/>
   </datagrid-column-sorter>
      </datagrid-column>
      <datagrid-column name="INCIDENT.INVESTIGATOR_USER_ID">
         <label>Investigator</label>
         <datagrid-renderer-ref id="sas_userNameRenderer"/>
         <datagrid-column-sorter name="userDisplayNameSorter"/>
      </datagrid-column>

      <grid-action id="INCIDENTS.GRID_BULK_UPDATE_ACTION" output-destination="inline"
            render-type="select">
         <label>Reassign to New Investigator</label>
         <url><eval>contextPath</eval>/controller/updateFields.json</url>
         <param name="entityType"                    value="'incident'"/>
         <param name="INCIDENT.INVESTIGATOR_USER_ID" value="TEMP.NEW_INVESTIGATOR"/>
      </grid-action>

   </datagrid>
```

Incident Triage Action

Overview

If you are displaying a list of SAS Enterprise Case Management incidents in a DataGrid, you can use this action to create a new case with any selected incidents pre-linked to it.

Note: When this action executes, the case will already be saved when you are redirected to the case editor. The case creation cannot be canceled.

URL

The URL for the Incident Triage action is ***<contextPath>*/controller/createCaseFromIncidents.json**.

Parameters

The Incident Triage action has the following parameters:

case_type_cd (required)
: specifies the type code of the case to create.

case_category_cd (optional)
: specifies the category code of the case to create.

case_subcategory_cd (optional)
: specifies the subcategory code of the case to create.

link_subjects (optional)
: specifies whether to automatically link all subjects that are linked to any selected incidents directly to the new case. The default value is false.

link_financial_items (optional)
: specifies whether to automatically copy any financial items associated with the selected incidents into the new case. The default value is false.

redirect (optional)
: specifies whether the user should be redirected into an editor for the new case when the action completes. The default value is true.

Example

The following example shows a DataGrid displaying a list of all unassigned incidents. It has a GridAction that creates a new case, with all selected incidents linked to that case. Any subjects and financial items associated with those incidents are copied into the new case, and the user is redirected to the case editor.

```
<datagrid name="ALERTS.GRID" selectedKeyField="INCIDENT.INCIDENT_RK">
 <label>Available Alerts</label>

 <datastore-ref id="ALERTS.STORE" />

 <datagrid-column name="INCIDENT.INCIDENT_ID" >
   <label>Alert ID</label>
   <datagrid-renderer-ref id="sas_entityLinkRenderer"
     args="Incident:INCIDENT.INCIDENT_RK"/>
 </datagrid-column>
 <datagrid-column name="INCIDENT.SOURCE_SYSTEM_CD">
   <label>Source System</label>
   <datagrid-renderer-ref id="sas_refTableRenderer" args="RT_SOURCE_SYSTEM"/>
<datagrid-column-sorter name="refTableSorter">
      <param name="referenceTable" value="'RT_SOURCE_SYSTEM'"/>
   </datagrid-column-sorter>
 </datagrid-column>
 <datagrid-column name="INCIDENT.INCIDENT_TYPE_CD">
   <label>Alert Type</label>
   <datagrid-renderer-ref id="sas_refTableRenderer" args="RT_INCIDENT_TYPE"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'RT_INCIDENT_TYPE'"/>
</datagrid-column-sorter>
 </datagrid-column>
 <datagrid-column name="INCIDENT.INCIDENT_DESC">
   <label>Description</label>
 </datagrid-column>
 <datagrid-column name="INCIDENT.UPDATE_USER_ID">
   <label>Updated by</label>
   <datagrid-renderer-ref id="sas_userNameRenderer"/>
   <datagrid-column-sorter name="userDisplayNameSorter"/>
 </datagrid-column>

 <grid-action id="ALERTS.GRIDACTION.CREATECASE" output-destination="window"
    render-type="select">
   <label>Create Case</label>
   <url><eval>contextPath</eval>/controller/createCaseFromIncidents.json</url>
   <image><url><eval>GetThemedImageURL("newform")</eval></url></image>
   <param name="case_type_cd"         value="'FIN'"/>
   <param name="link_subjects"        value="true"/>
   <param name="link_financial_items" value="true"/>
   <param name="redirect"             value="true"/>
```

```
 </grid-action>

</datagrid>
```

If you want the case to be created, but you do not want to be redirected to it in the UI, the **`output-destination`** attribute should be set to **`inline`**, and the **`redirect`** parameter set to false. This allows the case to be created without redirecting the browser to the case editor window. The following is how the GridAction from the preceding example would look with these changes:

```
<grid-action id="ALERTS.GRIDACTION.CREATECASE" output-destination="inline"
    render-type="select">
  <label>Create Case</label>
  <url><eval>contextPath</eval>/controller/createCaseFromIncidents.json</url>
  <image><url><eval>GetThemedImageURL("newform")</eval></url></image>
  <param name="case_type_cd"         value="'FIN'"/>
  <param name="link_subjects"        value="true"/>
  <param name="link_financial_items" value="true"/>
  <param name="redirect"             value="false"/>
</grid-action>
```

Custom GridActions

Overview

You can create custom GridActions and add them to a DataGrid. Following are examples of custom GridActions.

Example 1: View and Edit Row Menu Items for Entities

Suppose you have a DataGrid that is displaying incidents. You can create row-level actions allowing you to view or edit each incident as follows:

```
<datagrid name="INCIDENTS.GRID" selectedKeyField="INCIDENT.INCIDENT_RK">

 <label>Unassigned Incidents</label>
 <datastore id="INCIDENTS.GRID.STORE"
   data="GetEntityListAsMapList(GetUnassignedIncidents())"/>

 <datagrid-column name="INCIDENT.INCIDENT_ID">
  <label>Incident</label>
  <datagrid-renderer-ref id="sas_entityLinkRenderer"
    args="Incident:INCIDENT.INCIDENT_RK"/>
 </datagrid-column>
 <datagrid-column name="INCIDENT.INCIDENT_DESC">
    <label>Description</label>
 </datagrid-column>
 <datagrid-column name="INCIDENT.INVESTIGATOR_USER_ID">
    <label>Investigator</label>
    <datagrid-renderer-ref id="sas_userNameRenderer"/>
    <datagrid-column-sorter name="userDisplayNameSorter"/>
 </datagrid-column>

 <grid-action id="INCIDENTS.ROW_ACTION_EDIT" output-destination="javascript"
   render-type="row">
    <label>Edit Incident</label>
    <url>
```

```
    var incidentKey = actionData.selectedRowKey;
    var url = "<eval>contextPath</eval>/ViewObject.do?objectType=incident&amp;key="
        + incidentKey + "&amp;popup=false&amp;readOnly=false";
    ecmApi.redirect(url);
    </url>
  </grid-action>
  <grid-action id="INCIDENTS.ROW_ACTION_VIEW" output-destination="javascript"
    render-type="row">
    <label>View Incident</label>
    <url>
    var incidentKey = actionData.selectedRowKey;
    var url = "<eval>contextPath</eval>/ViewObject.do?objectType=incident&amp;key="
        + incidentKey + "&amp;popup=false&amp;readOnly=true";
    ecmApi.redirect(url);
    </url>
  </grid-action>

</datagrid>
```

Example 2: Case Network Analysis Menu Item for Parties

If you have a DataGrid displaying a list of parties, you can create a row-level menu item that displays the Case Network Analysis graph for a party, as follows:

```
<datagrid name="PARTIES.GRID" selectedKeyField="PARTY.PARTY_RK">

  <label>Grid of Parties</label>

  <datastore id="PARTIES.STORE"
    data="GetEntityListAsMapList(C_GetPartiesToDisplay()))"/>

  <datagrid-column name="PARTY.PARTY_FULL_NM">
    <label>Party</label>
    <datagrid-renderer-ref id="sas_entityLinkRenderer" args="Party:PARTY.PARTY_RK"/>
  </datagrid-column>

  <grid-action id="PARTIES.SHOW_CNA_GRAPH" output-destination="javascript"
    render-type="row">
    <label>Show CNA Graph</label>
    <url>
    var partyKey = actionData.selectedRowKey;
    var url = "<eval>contextPath</eval>/Director?_directive=viewSNANetwork&amp;
        analyticContext=ECM&amp;actionableEntityID=" + partyKey
        + "&amp;actionableEntityType=P";
    window.open(url);
    </url>
  </grid-action>

</datagrid>
```

DataGrid Column Renderers

Overview

The way DataGrids render their contents can be customized on a per-column basis. You might want to render Boolean values as Yes and No, or render a column of numbers

such that positive values are drawn in green and negative values are drawn in red. This type of customization is handled by DataGrid renderers.

The following is an example of how you can specify a renderer to use for a DataGrid column:

```
<datagrid-column name="STATECOL">
<label>Ref Table Lookups</label>
<datagrid-renderer-ref id="sas_refTableRenderer" args="X_RT_STATE_PROVINCE"/>
<datagrid-column-sorter name="refTableSorter">
   <param name="referenceTable" value="'X_RT_STATE_PROVINCE'"/>
</datagrid-column-sorter>
</datagrid-column>
```

The **`<datagrid-renderer-ref>`** element takes the ID of the DataGrid renderer to use for that column. If the renderer takes arguments, they can also be specified by the **`args`** attribute. Multiple arguments are separated by colons (:).

Note: When using sas_userNameRenderer or sas_refTableRenderer, it is important to use the corresponding DataGrid column sorter to ensure that the data is sorted by display value and not its coded value. See "DataGrid Column Sorters" on page 147 for more information.

Built-in Column Renderers

There are a number of built-in DataGrid column renderers for common use cases, including the following:

Renderer ID	Arguments	Item Rendered
sas_entityLinkRenderer	• Entity type to link to (either Case, Incident, Party, Report, or EFile) • The name of the **`<datagrid-column>`** element containing the key for the entity to link to	A link to another entity. The text value of the link is the contents of the cell. The entity linked to is specified by the arguments.
sas_dateRenderer	None	A date.
sas_dateTimeRenderer	None	A date or timestamp.
sas_currencyRenderer	None	A currency value.
sas_userNameRenderer	An optional argument with value **`"showUserId"`**. If **`"showUserId"`** is specified, the user ID of each user is displayed beside the display name.	The user ID of a member of the SAS Enterprise Case Management Users group that is using the user's display name.
sas_refTableRenderer	The name of the reference table.	Localized values from a reference table.

Renderer ID	Arguments	Item Rendered
sas_booleanRenderer	An optional argument whose value is either **"checkboxes"** or **"text"**.	By default, renders Boolean values as Yes and No (using localized text). If the optional argument is specified as **"checkboxes"**, disabled check boxes are rendered instead of text.
sas_financialItemLinkRenderer	None	This renderer should only be used in FinancialItemsGrids. It renders a link that displays the financial item in a modal window for editing or viewing. The text value of the link is the contents of the cell.
sas_financialItemOriginRenderer	None	This renderer should be used only in FinancialItemGrids. It renders the type of entity that was the source of this Financial Item (for example, "CASE" or "INCIDENT").

Custom Column Renderers

You can also create DataGrid column renderers specific to your application. To do this, you embed a JavaScript snippet that handles the rendering of cells inside a UI definition, using the **<datagrid-renderer>** element.

The ID of the renderer is specified by the **id** attribute of the **<datagrid-renderer>** element. The child text of the element should be JavaScript code. This code can be thought of as the contents of a JavaScript function. The code has a single predefined variable (data), which is the contents of the cell to render. The JavaScript should return a valid HTML snippet, which will be embedded in the DataGrid cells for the relevant column.

SAS Enterprise Case Management does not validate that custom column renderers generate valid JavaScript. Therefore, the following precautions should be taken when writing them:

- The JavaScript code should be wrapped in a CDATA block to prevent malformed markup.
- If you use nested **<message>** tags to add localized text into a custom renderer, make sure that all localized property values have any necessary escape characters so that they do not cause malformed JavaScript when inserted into your renderer. For example, you need to add an extra backslash (\) character before a single quotation mark or double quotation mark in a property value to escape it if that property is going to be embedded in a JavaScript string literal in your renderer.
- If you use nested **<eval>** tags to add the results of expressions into your custom renderer, make sure that the result does not contain character sequences that result in invalid JavaScript in your custom renderer.

The following is an example custom column renderer that highlights negative numbers with a red background and positive numbers with a green background:

```
<datagrid-renderer id="C_customRenderer">
```

```
<![CDATA[
   var result = "<div style='background-color: ";
   if (data < 0) {
      result += "#ffd0d0";
   }
   else {
      result += "#d0ffd0";
   }
   result += "'>";
   if (data < 0) {
      result += "<i>" + data + "</i>";
   }
   else {
      result += "<b>" + data + "</b>";
   }
   result += "</div>";
   return result;
]]>
</datagrid-renderer>
```

Note: You must prefix custom cell renderer IDs with “C_” to prevent naming conflicts with SAS standard renderers.

DataGrid Column Sorters

Overview

The data for grids is stored in memory on the server. Therefore, sorting needs to be done in memory as well. The Custom Page Builder supports sorting of base-type objects without having to specify a sorter on a column. Because SAS Enterprise Case Management supports reference table and user name renderers, sorters are provided for these column types. Custom sorters are not supported at this time.

Built-in Column Sorters

SAS Enterprise Case Management supports the following sorters:

Sorter	Arguments	Sorting Details
refTableSorter	referentTable: the name of the reference table	Sorts by either the DISPLAY_ORDER_NO column or the VALUE_DESC column defined in the REF_TABLE_VALUE database table.
userDisplayNameSorter	None	Sorts by the display name defined for the user in the metadata or by the value of the field if no display name is defined.

Examples

The following is an example of refTableSorter:

```
<datagrid-column name="INCIDENT.INCIDENT_TYPE_CD">
   <label><message key="field.incident.incident_type_cd.header.txt"/></label>
   <datagrid-renderer-ref id="sas_refTableRenderer" args="RT_INCIDENT_TYPE"/>
   <datagrid-column-sorter name="refTableSorter">
       <param name="referenceTable" value="'RT_INCIDENT_TYPE'" />
   </datagrid-column-sorter>
</datagrid-column>
```

The following is an example of userDisplayNameSorter

```
<datagrid-column name="INCIDENT.UPDATE_USER_ID">
   <label>Updated by</label>
   <datagrid-renderer-ref id="sas_userNameRenderer"/>
    <datagrid-column-sorter name="userDisplayNameSorter"/>
   </datagrid-column>
```

Custom Page Builder Components

Overview

A collection of built-in components is provided for SAS Enterprise Case Management. These components can be added to user interface (UI) definition XML files. UI definition XML files are used to define the layout of fields for viewing or editing cases, incidents, and parties.

The Custom Page Builder supports components that can extend the **`<datagrid>`** element functionality defined earlier in this chapter. Components that you want to use the **`<datagrid>`** element with should extend the class **`com.sas.solutions.cpb.runtime.component.CustomGridComponent`**.

Components have the ability to listen to the following events, so they can provide their own additional functionality after the save of a screen has occurred.

PRE_SAVE
: Called before a save or save draft.

SAVE
: Called after the save of an entity.

SAVE_DRAFT
: Called after the save draft of an entity. Currently only supported by report entities.

PRE_INITIALIZE
: Called just before the initialize block in a screen definition is evaluated.

POST_INITIALIZE
: Called just after the initialize block in a screen definition is evaluated.

Here is an example of how a component can listen to the events:

```
@Override
public void handleEvent(
        Field field, UIContext uiContext,
        HttpServletRequest request, RuntimeEvent event)
throws EvaluationException
{
  switch (event)
```

```
    {
    case SAVE:
    case SAVE_DRAFT:
      // Perform post save work here.
      postSave(field, uiContext, request);
      break;
    case PRE_SAVE:
      preSave(field, uiContext, request);
      break;
    case PRE_INITIALIZE:
      preInit(field, uiContext, request);
      break;
    case POST_INITIALIZE:
      postInit(field, uiContext, request);
      break;
    }
  }
```

For more information about components, see the Custom Page Builder documentation found on the **Administration** menu of SAS Enterprise Case Management.

Note: The **`name=`** field should be unique for each component when the same component is used multiple times in the UI definition.

Static Component Field Formatters

The term “field” within the context of a Custom Page Builder component is different from a stand-alone field. Only a field within a static component can currently use formatters. Stand-alone “field” types cannot use formatters as defined in this section.

In cases where you can specify a formatter, the following names can be used:

- Boolean
- check box (Read-Only view)
- currency
- date
- datetime
- decimal
- generic
- hyperlink
- integer
- party
- user_name
- any valid reference table name (for example: X_RT_PRIORITY)
- strip (removes the number formatting, for example 1.0 becomes 1)

Custom Action

The **`<action-group>`** element is rendered as a tool bar. It contains **`<action>`** elements. The **`<action>`** element enables you to provide links and buttons on a screen.

For a list of the attributes that you can use with the **<action>** element, see "Valid XML Elements and Descriptions for User Interface Definitions" on page 101 . The **<action-group>** element does not support attributes. You can nest **<action>** elements and conditional logic within **<action-group>** elements. You can nest the **<action-group>** element under **<screen>**, **<section>**, and **<tab>** elements.

Note: Custom action functionality is available on the main-level Custom Page Builder window. However, it is not available on windows spawned from the GenericEntityTable component.

A custom action should not be used to open a SAS Enterprise Case Management entity in a popup window (specifying "window" for the output-destination parameter). This includes all means of opening an entity, for example from the ViewECMObject and NewECMObject directives, or direct linking through the Case.do action. SAS Enterprise Case Management only supports running in a single browser window. Opening multiple browser windows for a single SAS Enterprise Case Management session may result in unexpected errors.

Element	Description
<action-group>	A group of actions. **Attributes:** None **Child elements:** Zero or more **<action>** elements

Element	Description
`<action>`	**Attributes:** **`URL`** specifies the landing URL after a submit. • **`<eval>`** — This element is a child of the **`URL`** attribute. The text of this element is treated as an expression and evaluated. **`id`** (optional) specifies the ID field. **`output-destination`** (optional) specifies the output format, either **`window`** or **`inline`**. The default is **`window`**. A value of **`window`** specifies showing the result page in a pop-up window. A value of **`inline`** specifies no change in the structure of the current page, and no pop-up window. The output-destination could be a field replacement javascript call. The value **`ignore`** specifies a server side trigger, with no UI change. **`trigger`** (optional) specifies the trigger action, The trigger action is currently labeled as “support save”. **`fail-on-error failed`** (optional) specifies the Boolean field that enables you to decide whether to continue running other trigger actions if there is an error in the action execution. **`visible`** (optional) specifies whether the action is visible. **`enabled`** (optional) specifies whether the action is enabled. **`content-type`** (optional) specifies the response content type for a backend action. This attribute currently supports text and HTML. --- **Child elements:** • An optional **`<label>`** element. This element provides the window label or title. • An optional **`<param>`** element. This element provides parameters for the URL. • An optional **`<url>`** element. This element is used if an attribute URL is not specified. It provides a more convenient way to specify a URL.

The following example demonstrates how to trigger the Save action on a screen.

Example Code 6.3 *Save Trigger*

```
<action-group>
  <action url="'http://yourserver.com:8080/SASStoredProcess/do'">

    <!-- Configure the behavior of the execution -->
    <param name="_action" value="'form,properties,execute,nobanner,newwindow'"/>

    <!-- Path to the stored process to execute -->
    <param name="_program" value="'/Samples/SAS Enterprise Case Management/
Stored Process/stored_process_name"/>

    <!-- All other parameters are used to initialize the prompts -->
    <param name="favoriteColor" value=""/>
  </action>

<!-- The Save action -->
  <action url="'http://www.google.com/search'" output-destination="inline"
trigger="save">
  <param name="q" value="'flower'"/>
  </action>
</action-group>
```

The following example demonstrates how to use a value from a drop-down list with a custom action.

Example Code 6.4 *Using a Value from a Drop-Down List with a Custom Action*

```
<field name="sourceSystemCd" type="dropdown" default="ECM"
visible="false"/>
  <action-group>
   <action url="'http://www.google.com/search'"enabled="true">
     <label>google search source system code</label>
     <param name="q"><eval>sourceSystemCd</eval></param>
  </action>
</action-group>
```

Custom Page Builder Directives

Overview

Directives provide a means for external applications to call into SAS Enterprise Case Management.

Case

Description

Deprecated. Use the ViewECMObject directive instead.

Parameter

key (required) specifies the case to display.

Example

/Director?_directive=Case&key=10008

viewECMCaseReport

Description

Displays generated case reports.

Parameter

None.

Example

/Director?_directive=viewECMCaseReport

ViewECMObject

Description

Generic object display directive. Opens a SAS Enterprise Case Management entity for viewing or editing.

Parameter

objectType (required)
: specifies the type of object to view. This should be either **`case`**, **`incident`**, **`party`**, **`report`**, or **`efile`**.

key (required)
: specifies the key of the SAS Enterprise Case Management entity to view. You should specify either this parameter, or both id and source_system_cd, but not all three.

id (optional)
: if key is not specified, used together with source_system_cd to specify the SAS Enterprise Case Management entity to view.

source_system_cd (optional)
: if key is not specified, used together with id to specify the SAS Enterprise Case Management entity to view.

readOnly (optional)
: determines whether to open the entity in Edit Mode or Read-Only mode. The default is **`true`**.

popup (optional)
: determines whether to open the entity in a pop-up window. The default value is **`true`**.

Example

/Director?_directive=ViewECMObject&objectType=CASE&key=10008&popup=false

ViewGenericPage

Description

Displays a generic page in SAS Enterprise Case Management (that is, a page that is not tied to a specific SAS Enterprise Case Management entity).

Parameter

uiDefinitionName (required)
: specifies the name of the UI definition.

screenID (optional)
: specifies the window in the UI definition to navigate to. The default value is **`main`**, if this parameter is omitted.

Others (optional)
: Any other parameters starting with "TEMP." will be available in the UI definition as variables.

Example

/Director?
_directive=ViewGenericPage&uiDefinitionName=myCustomUIDef&TEMP.PARAM=
value

PreviewECMReport

Description

Displays a PDF preview of a report.

Parameter

key (required)
: specifies the key of the report.

objectKey (optional)
: is a synonym for the key parameter.

rr_rk (optional)
: is a synonym for the key parameter.

Example

/Director?_directive=PreviewECMReport&key=10025

ECMWorkflowCallback

Description

The workflow callback directive for SAS Enterprise Case Management.

Parameter

command (required)
: specifies the workflow action to execute. This should be one of **`setStatus`**, **`setOpened`**, **`setReopened`**, or **`setFields`**.

entityType (required)
: specifies the type of entity whose workflow should be updated. This should be either **`case`** or **`rr`**.

key (required)
: specifies the case or regulatory report key. This is usually stored in the data object CASE__CASE_RK or RR__RR_RK.

Note: Other parameters are available by command (see the following examples).

Example

- /Director?_directive=ECMWorkflowCallback& command=setStatus&key=${../CASE__CASE_RK}&statusCode=**`<caseStatusCode>`**
- /Director?_directive=ECMWorkflowCallback& command=setStatus&key=${../CASE__CASE_RK}&statusCode=**`<caseStatusCode>`**&caseClosed=true
- /Director?_directive=ECMWorkflowCallback& command=setOpened&key=${../CASE__CASE_RK}
- /Director?_directive=ECMWorkflowCallback& command=setReopened&key=${../RR__RR_RK}
- /Director?_directive=ECMWorkflowCallback& command=setFields&entityType=CASE&FIELD1=value1&FIELD2=value2&…
- /Director?_directive=ECMWorkflowCallback& command=setFields&entityType=RR&FIELD1=value1&FIELD2=value2&…

CorrectReport

Description

Files a corrected report.

Parameter

key (required) specifies the key of the report to correct.

Example

/Director?_directive=CorrectECMReport&key=10001

ECMWelcome

Description

The default target for SAS Enterprise Case Management. This opens the user's typical logon page.

Parameter

None.

Example

/Director?_directive=ECMWelcome

NewECMObject

Description

Creates a new SAS Enterprise Case Management entity.

Parameter

objectType (required)
: specifies the type of object to view. This should be either **`case`**, **`incident`**, **`subject`**, **`report`**, or **`efile`**.

type (required)
: specifies the type code of the object to create.

category (optional)
: specifies the category code of the object to create.

subcategory (optional)
: specifies the subcategory code of the object to create.

Others (optional)
: Any other parameters that match the name of fields of the object are used as initial values for those fields. Only standard fields and single-value UDF fields are currently supported. Some fields that are important to the SAS Enterprise Case Management data model, such as key fields like CASE.CASE_RK and INCIDENT.INCIDENT_RK, cannot be set with this directive and are ignored.

Example

/Director?
_directive=NewECMObject&objectType=CASE&type=FIN&CASE.CASE_DESC=My
%20New%20Case

Chapter 7

Using and Configuring Global Search

Introduction 157
Indexing Data 157
Global Search Configuration XML Format 158
Valid XML Elements and Descriptions for Global Search Configuration 160
Global Search Configuration Administration 165
Managing Multiple Global Search Configurations 165
Global Search Directive 166
Bulk Load Capability 166
Global Search Re-Indexing 166
ECM.Search.URL Parameter 167
Enabling or Disabling Global Search 167
Manually Installing Global Search 167

Introduction

The global search feature in SAS Enterprise Case Management enables you to search across all content fields from a single search box displayed in the application header. The results are returned as a list of customizable tiles. The title, icon, and details displayed in the results are customizable by object type. Those details are controlled by entries in the global search configuration files.

To enable the global search feature, you must install the search engine. For more information on the configuration process, see "Pre-Installation: Global Search" on page 13.

Indexing Data

When global search is enabled, SAS Enterprise Case Management automatically indexes case management data and all comments. Indexes are updated as data changes. The data can be re-indexed, as necessary, when your indexing configuration changes. For more

information about refreshing the search index, see "Global Search Re-Indexing" on page 166.

The global search index can also be configured to index data from external data sources. Because SAS Enterprise Case Management seamlessly displays external data side-by-side with data stored by SAS Enterprise Case Management, it is important for global search to also be able to find and index that external data. There are two approaches for indexing data.

- providing a REST API for the indexing service to call
- executing a batch job that loads data into the index

More information about configuring a REST service can be found in "Global Search Configuration XML Format" on page 158. Information about batch loading data into the search index can be found by reviewing the macro ecm_srch_index_driver in *SAS-configuration-directory*/**Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros**.

Global Search Configuration XML Format

This section contains all of the elements available in the global search configuration XML. The following is an example XML file.

```
<?xml version="1.0"?>
<search_config>
  <document_types>

    <document_type>
        <document_type_name>ECM-Cases</document_type_name>

        <data_sources>
          <data_source>
            <pull_url>http://<hostname>/SASEntCaseManagement/
          api/cases.json?start=0&amp;limit=100</pull_url>
            <append_cas_ticket>true</append_cas_ticket>
            <debug_disable_paging>false</debug_disable_paging>
            <items_property>items</items_property>
          </data_source>
        </data_sources>

        <type_field_value>Case</type_field_value>
        <security_field>groups</security_field>

        <unique_id_field>CASE.CASE_RK</unique_id_field>
        <unique_id_prefix>C_</unique_id_prefix>

        <type_discriminator_field_name>entityType</type_discriminator_field_name>
        <type_discriminator_field_value>CASE</type_discriminator_field_value>

        <fields>
            <field>
                <name_in>CASE.CASE_RK</name_in>
                <datatype>string</datatype>
            </field>
            <field>
```

```
                <name_in>CASE.CASE_DESC</name_in>
                <name_out>DESCRIPTION</name_out>
                <datatype>string</datatype>
            </field>
            <field>
                <name_in>CASE.CREATE_DTTM</name_in>
                <datatype>date</datatype>
            </field>
            <field>
                <name_in>TRANSACTION_AMOUNT</name_in>
                <datatype>double</datatype>
            </field>
            <field>
                <name_in>EMPLOYEENUM</name_in>
                <datatype>long</datatype>
            </field>
            <field>
                <name_in>INTERNAL_FLG</name_in>
                <datatype>boolean</datatype>
            </field>
        </fields>

        <display_layout>
            <icon_class_name>case_48</icon_class_name>
            <title_fields>
                <field_name>CASE.CASE_RK</field_name>
            </title_fields>
            <meta_fields>
                <field_name>CASE.CREATE_DTTM</field_name>
                <field_name>TRANSACTION_AMOUNT</field_name>
            </meta_fields>
            <link>
                <url>http://<hostname>:8080/someapp?id={CASE.CASE_RK}</url>
                <open_new_window>true</open_new_window>
            </link>
        </display_layout>
    </document_type>

  </document_types>
</search_config>
```

Each search configuration XML file consists of one or more document types. A document type represents a type of object or entity to be indexed (and later, displayed in the search results). In SAS Enterprise Case Management, each document type might correspond to an entity (case, incident, party, report, or e-file). You should only configure document types for the entities that you want to query. For example, if you track incidents but are not interested in querying for them, then no document type should be configured for incidents in the search configuration XML.

The example search configuration XML above contains only one **`<document_type>`** for brevity. However, multiple **`<document_type>`** elements are allowed within the **`<document_types>`** element.

Note: Number formatting is not supported.

Valid XML Elements and Descriptions for Global Search Configuration

The following table describes the XML format used in the global search configuration files:

Table 7.1 *Global Search Configuration XML Format*

Element	Required	Description
`<document_type_name>`	Yes	A short descriptive name for this type. It must be unique across all types. This name is used as a parameter to specify what source to load when performing a bulk load operation.

Element	Required	Description

`<data_sources>`	No	A list of **`<data_source>`** declarations. The data sources are used only for bulk loading data into the search index. They are not relevant to external callers pushing data to the search index. A data source contains settings that are relevant to the actual pulling or fetching of the raw data.
		`<data_sources>`
		` <data_source>`
		`  <items_property>items</items_property>`
		`<pull_url>http://somehost.com/`
		`  SASEntCaseManagement/`
		`  api/cases.json?start=0&`
		`  limit=100</pull_url>`
		`<append_cas_ticket>true</append_cas_ticket>`
		`<debug_disable_paging>false</debug_`
		` disable_paging>`
		`  </data_source>`
		`  </data_sources>`
		Child elements:
		`<items_property>` (Required: No)
		The default, if not provided, is **`items`**. When fetching or pulling data from the **`<pull_url>`** element, this string specifies the JSON element that contains the JSON array of items to be indexed. The default name **`items`** is compatible with the JSON currently returned by the SAS Enterprise Case Management REST service for pulling entities.
		`<pull_url>` (Required: Yes)
		The URL that the indexer will call during a bulk load operation. The response must be in JSON format. Other formats might be supported in a later release. Note that because this search config file format is XML, certain characters such as ampersand (&) must be properly escaped, as shown in the example.
		`<append_cas_ticket>` (Required: No)
		The default, if not provided, is **`true`**. When fetching or pulling data from the **`<pull_url>`** element, if this option is **`true`**, then the indexer code automatically appends a CAS security ticket to the pull URL. This is necessary when calling REST services hosted by SAS Enterprise Case Management to pass single sign-on credentials. If the URL does not point to a SAS application, such as a REST service on the Internet, then this option should be set to **`false`**.
		`<debug_disable_paging>` (Required: No)
		The default if not provided, is **`false`**. When fetching or pulling data from the **`<pull_url>`** element, if this option is **`true`**, the fetcher code does not attempt to do an HTTP GET of the URL more than once, even if the response contained a “next” link indicating that additional pages are available. This is useful only for testing or debugging.

Element	Required	Description
<type_field_value>	Yes	When the indexer creates each search document, it populates a field named **__type** with the literal value specified here. This value is used by the UI when rendering search results. However, its value does not necessarily appear in the search results. This element must be unique across all document types.
<security_field>	No	The default, if not provided, is **groups**. When the indexer creates each search document, it populates a special security field. This field is multivalued. This field is populated with the value(s) of the incoming field whose name is given in the **<security_field>** element. This is the list of OMR groups whose members are allowed to view the item in the search results. At query time, the results are filtered so that the current user can see only items that he or she has access to see. The default name **groups** was chosen to be compatible with the JSON currently returned by the SAS Enterprise Case Management REST service for pulling entities.
<unique_id_field> and **<unique_id_prefix>**	Yes	When the indexer creates each search document, it generates a unique ID value (that is, unique within the search index). This generated unique ID is placed in the search document in a field named **__id**. The value of this generated ID is the literal value of **<unique_id_prefix>** concatenated with the value of the incoming field whose name is given in **<unique_id_field>**. For example, suppose that cases are being bulk loaded (pulled), and the first case item has the field CASE.CASE_RK with value 123. The generated unique ID would be C_ concatenated with 123, resulting in C_123. The field named in **<unique_id_field>** must also appear in the **<fields>** element.
<type_discriminator_field_name> and **<type_discriminator_field_value>**	Yes	When items are pushed to the search indexer (as in the search API's addUpdate operation), the indexer must determine which **<document_type>** declaration to associate with the item, so that the item can be properly parsed and indexed. *Note:* The preceding example search configuration XML contains only one **<document_type>** element. However, typically there are multiple document types. The indexer inspects each incoming item for a field whose name is given in **<type_discriminator_field_name>**. If it exists, and its value matches the value given in **<type_discriminator_field_value>**, then the indexer will know to associate the item with this document type. Incoming items that cannot be associated to any document type is skipped.

Element	Required	Description

`<fields>`	Yes	A list of **`<field>`** declarations. Every field to be indexed or returned in search results must be declared here. If the search indexer receives a field that is not declared here, it is silently ignored. Each **`<field>`** element has the format as shown in the following example.
		`<field>`
		`  <name_in>CASE.CASE_DESC</name_in>`
		`  <name_out>DESCRIPTION</name_out>`
		`  <datatype>string</datatype>`
		`</field>`
		Child elements:
		`<name_in>` (Required: Yes)
		The name of the incoming field, as it appears when items are either pulled or pushed to the search indexer.
		`<name_out>` (Required: No)
		The default, if not provided, is the same as declared in **`<name_in>`**. The name that the field is given when it is added to the search document. It only needs to be specified if you want the field name to be different in the search index (and therefore in the search results) than it is in the source system.
		`<datatype>` (Required: No)
		The default, if not provided, is **`string`**. It specifies the data type used in the search engine to store this field's value. The supported data types are: string, date, double, long, and Boolean. The search indexer can perform some data type conversions, as shown in the Data Type Conversions table on page 165.

Element	Required	Description

`<display_layout>`	Yes	This section is not involved in indexing, but is important when search results are rendered in the UI. The format is shown in the following example:
		
		`<display_layout>`
		` <icon_class_name>case_48</icon_class_name>`
		` <title_fields>`
		` <field_name>CASE.CASE_RK</field_name>`
		` </title_fields>`
		` <meta_fields>`
		` <field_name>CASE.CREATE_DTTM`
		`  </field_name>`
		` <field_name>TRANSACTION_AMOUNT`
		`  </field_name>`
		` </meta_fields>`
		` <link>`
		` <url>http://<hostname>:8080/`
		`  someapp?id={CASE.CASE_RK}</url>`
		` <open_new_window>true</open_new_window>`
		` </link>`
		`</display_layout>`
		
		Child elements:
		
		`<icon_class_name>` (Required: Yes)
		This is passed to the search results UI for rendering purposes. It should be the name of a CSS class. This determines which image is rendered in the search result tile for all search results of this type.
		
		`<title_fields>` (Required: Yes)
		Contains one or more **`<field_name>`** elements. Causes the value of the named field or fields to be rendered on the first line in the search result tile.
		
		`<meta_fields>` (Required: Yes)
		Contains one or more **`<field_name>`** elements. Causes the value of the named field or fields to be rendered on the second line in the search result tile.
		
		`<link>` (Required: Yes)
		This element directs you to the appropriate location when a search result of this type is clicked. The **`<url>`** element is required, and specifies the location.
		
		Note that the URL string can contain one or more field names, each enclosed by curly braces, such as {CASE.CASE_RK}, as shown in the preceding **`<display_layout>`** example.
		
		When the link is followed (the search result clicked), the braces and field name are replaced by that field's value. The **`<open_new_window>`** element is optional, and defaults to **`false`**. The search results UI uses this Boolean value to determine whether a new browser window or tab should be opened when a user clicks the search result.

The following table provides conversion information for data types in the `<datatype>` element.

Table 7.2 *Data Type Conversions*

	In (from source)				
	String	**Boolean**	**Long**	**Double**	**Date**
Out (to index)					
String		toString()	toString()	toString()	toString()
Boolean	Boolean. parseBoolean()		Log a warning	Log a warning	Log a warning
Long	Long. parseLong()	Log a warning		longValue()	Log a warning
Double	Double. parseDouble()	Log a warning	doubleValue()		Log a warning
Date	ISO8601 string is converted to a Date	Log a warning	Log a warning	Log a warning	

Global Search Configuration Administration

The search_config.xml file can be managed through the Search Configuration window found under the **Administration** tab.

Note: This action requires the Upload Search Configuration capability.

The upload utility places the search configuration file on the content server and then reloads the search configuration in SAS Enterprise Case Management. This does not clear the search index or re-index the sources defined in the search_config.xml file. It simply resets the configuration for the next time search is re-indexed.

Managing Multiple Global Search Configurations

It is possible to upload multiple global search configuration files. Each search configuration file can be used to index data from different sources. There is no restriction on the name of the file being uploaded. It only needs to be a valid search configuration XML document. In addition, you can delete search configuration files by selecting **Delete Search Configuration** from the drop-down menu next to the configuration. A pop-up window appears prompting you to either download or delete the configuration row.

Global Search Directive

A directive named ECMSearch was created to access search directly in SAS Enterprise Case Management. The only required parameter is q, which specifies the search term. The following example URL directs you to the search results that all contain “john.”

http://ecm5.grc.sas.com/SASEntCaseManagement/Director?_directive=ECMSearch&q=john

Note: Special characters need to be URL encoded if they are passed in the q parameter of the ECMSearch directive.

Bulk Load Capability

The Bulk Load Index capability is part of the Administration and Advanced Administration groups. This capability allows you to call the clear, bulkload, and bulkloadall HTTP endpoints for bulk modification of the search index. The following are the URLs used to perform these calls:

/SASEntCaseManagement/api/search/clearindex.json
clears the search index. Consider doing this prior to a full re-index through bulkloadall.

/SASEntCaseManagement/api/search/bulkloadall.json
loads data from all data sources. This does not clear the index.

/SASEntCaseManagement/api/search/ECM-Cases/bulkload.json
Initiates the indexing of a particular record type (for example, "ECM-Cases") as defined in the search_config.xml file

/SASEntCaseManagement/api/search/showindexinfo.json
shows how many items exist of each type in the index and tells you whether the background indexer is running.

Global Search Re-Indexing

SAS Enterprise Case Management can be re-indexed manually. This is useful for a number of reasons. For example, you might index manually if you changed your global search configuration and want to pick up the new fields. The same could be true for removing fields. As another example, if you were migrating from an older version of SAS Enterprise Case Management, you might re-index in order to index old values.

Note: SAS Enterprise Case Management updates each document individually when a save occurs, to keep the search index fresh. For example, if you index the CASE.CASE_DESC and then update that field on the case screen, when you save, CASE.CASE_DESC will be indexed for search.

The following are the three options for search re-indexing, as well as the URLs to use:

`http://server:port/SASEntCaseManagement/api/search/clearindex.json`
clears the search index.

`http://server:port/SASEntCaseManagement/api/search/bulkloadall.json`
loads the entire index.

`http://server:port/SASEntCaseManagement/api/search/<name>/bulkload.json`
loads a portion of the index. The **`<name>`** is the name of the document you are trying to index. It is the **`<document_type_name>`** element in the search configuration XML.

ECM.Search.URL Parameter

The metadata property ECM.Search.URL is used to point to the search index. The URL should be in the form of **`http://<server>:<port>/solr/<collection_name>`** when search is used. This value is typically set during installation. You can edit the property in SAS Management Console by navigating to **Plug-ins** ⇨ **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Ent Case Mgmt Mid-Tier 6.2** . Right-click **Ent Case Mgmt Mid-Tier 6.2** and select **Properties**. On the **Advanced** tab, modify the value. After the value has been modified, you must clear the cache in SAS Enterprise Case Management in order to pick up the new value in the web application. Your search queries should now be pulling from the new search index.

Enabling or Disabling Global Search

The metadata property ECM.Search.Enabled can enable or disable search. It requires a value of **`true`** to enable searching or **`false`** to disable it. Search is initially enabled or disabled based on what was selected during the SAS Deployment Wizard installation. You can edit the property in SAS Management Console by navigating to **Plug-ins** ⇨ **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Ent Case Mgmt Mid-Tier 6.2**. Right click **Ent Case Mgmt Mid-Tier 6.2** and select **Properties**. Go to the **Advanced** tab. Set the value to **`true`** or **`false`**. If you are enabling global search, you also need to provide the URL of the Solr server in the ECM.Search.URL property. After those properties have been modified, you must clear the cache in SAS Enterprise Case Management in order to pick up the new values in the web application.

After modifying the ECM.Search.Enabled flag, you must log off and log back on to ensure that your menus get updated correctly.

Manually Installing Global Search

If you initially decided not to install global search and want to install it after the fact, follow these steps:

1. In SAS Management Console, set ECM.Search.Enabled to **`true`**. See "Enabling or Disabling Global Search" on page 167 for details on performing this step.
2. In SAS Management Console, set ECM.Search.URL to the URL of your search index. See "ECM.Search.URL Parameter" on page 167 for details on performing this step.
3. In SAS Enterprise Case Management, add the Upload Search Configuration capability to your user. See "Global Search Configuration Administration" on page 165 for additional information on this step.
4. Clear the cache, and then log off of SAS Enterprise Case Management and log back on. **Search Configuration** should be visible under the **Administration** tab.
5. You can either upload your own search configuration or use the samples that are shipped with SAS Enterprise Case Management. You can find the samples at *SAS-middle-tier-installation-directory***`/deploy/sample/SearchConfig`**. You will need to rename the file to *.xml. You will also need to edit those files to replace the **`<pull_url>`** element with the correct URL for pulling in data.
6. You should now see the global search box at the top of the page. If not, see "Using the Search Functionality in SAS Enterprise Case Management " on page 242.

Chapter 8

Regulatory Reports and E-Filing

Introduction **170**

FINCEN SAR Implementation **170**

FINCEN CTR Implementation **171**

E-Filing Process **172**

Configuring E-Filing **173**
Steps 173
Configuring Report User-Defined Fields 174
Configuring the UI Definition of the Report Container 174
Configure the Report UI Definition 175
Configure the Report Workflow Definition 176
Configure the E-file User-Defined Field 176
Configure the E-File UI Definition 176
Enable Report Update in the E-File Stored Process 176

Regulatory Report Form Configuration **177**
Steps 177
Define the Regulatory Report Form in FORM_CONFIG 178
Define the SAS Macro to Generate the Regulatory Report Form for Regulatory Report Preview 179
Define the SAS Macro to Transform Regulatory Report Data into Regulatory Report Form Data 179
Define the Regulatory Report Form Layout 179
Define the Regulatory Report Form Section Configuration Data in FORM_PREVIEW_SECTION_CONFIG 180
Define the Regulatory Report Form Field Configuration Data in FORM_PREVIEW_FIELD_CONFIG 180
Define the SAS Macro to Format Source Data for Form Preview 181
Define the SAS Macro to Generate the E-File 181
Define the SAS Macro to Transform Regulatory Report Data into E-File-Ready Data 181
Define the E-File Record Configuration Data in FORM_EFILE_RECORD_CONFIG 181
Define the E-File Field Configuration Data in FORM_EFILE_FIELD_CONFIG 182

Support of Multiple Versions of Report Forms **183**

FINCEN **183**

SAR **186**
SAR Files 186
SAR Report UI 186

CTR **187**
CTR Files 187
CTR Report UI 188

Introduction

The U.S. Department of the Treasury, and specifically the Financial Crimes Enforcement Network, require financial institutions to report suspicious activity and large cash transactions with a regulatory report. To create a regulatory report for e-filing, the user interface definition of the case or incident must have the ReportsTable component defined. This component shows a table of reports that have been created for the case or incident. SAS Enterprise Case Management is delivered with the capability to produce the Suspicious Activity Report (SAR) and Currency Transaction Report (CTR) published by FINCEN in March 2012. This chapter describes the steps that a user can follow in order to produce a complete regulatory report and e-file that report in a batch of other similar reports.

FINCEN SAR Implementation

The following steps summarize the post-installation procedure to implement FINCEN SAR reporting. This procedure assumes that the post-installation process described in Chapter 4, "Post-installation Requirements and Tasks," on page 31 is complete.

1. Load SAR configuration data. Follow the instructions in "Loading the SAS Enterprise Case Management Configuration Tables" on page 40 to load loadrr_fincen_new.sas, loadrr_fincen_new_base.sas, and loadrr_config_fincen_sar.sas, if they have not been loaded. If you have upgraded from SAS Enterprise Case Management 2.3 or later, loadrr_fincen_new_base.sas can be skipped.
2. Clear the cache. Follow the instructions in "Clearing the Cache" on page 40 to clear the old configuration data in memory.
3. Upload the user interface definitions. Follow the instructions in "Uploading User Interface Definitions" on page 41 to upload rr-fincen-sar-01.xml and efile-fincen-new-01.xml, if they have not already been uploaded.
4. Upload the workflow definitions. Follow the instructions in "Uploading Workflow Definitions" on page 42 to upload FINCENReport.xml, if the workflow has not been uploaded.
5. Set the user permission to the e-file directory. Follow the instructions in "Setting User Permission to the E-File Directory" on page 39 to enable the SAS stored process to create e-files in ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/efiles**.
6. Install Winzip software. Winzip software is needed for e-filing FINCEN SAR attachments. For most UNIX environments, Winzip should be installed in **/usr/bin**. For Windows, make sure that wzzip.exe exists.
7. Edit ecm_global_mvar.sas. The following parameters defined in ecm_global_mvar.sas under ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros** should be specified for SAR e-filing:

- ecm_win_zip_cmd — The Windows command for Winzip.
- ecm_unix_zip_cmd — The UNIX command for Winzip.
- ecm_webusername — The user name for connecting to the middle tier. This user is required to have Edit Report capability.
- ecm_webpassword — The password of the user account for ecm_webusername. This password can be encoded with SAS PROC PWENCODE.

If you prefer not to use a specific user for e-filing, you can leave these parameters empty. The credential of the user who is generating the e-file will be used. That user is required to have the Edit Report capability. If the user does not have that capability, the transaction sequence numbers generated during the e-file generation process, cannot be added to the reports.

8. Configure the SAS stored process server to run system commands. The new FINCEN e-file process requires reports to be zipped together. In order to execute the zip command, the XCMD option must be enabled for the SAS Stored Process Server so that the SYSTEM function can be used. To enable the XCMD option:
   a. Using SAS Management Console, log on as an adminstrative user.
   b. From the **Plug-ins** tab, select **Environment Management** ⇨ **Server Manager** ⇨ **SASApp** ⇨ **SASApp - Logical Stored Process Server** ⇨ **SASApp - Stored Process Server**.
   c. Right-click and select **Properties**.
   d. On the SASApp - Stored Process Server Properties window, select the **Options** tab, and click **Advanced Options**.
   e. On the Advanced Options window, click the **Launch Properties** tab.
   f. Check the **Allow XCMD** check box.
   g. Click **OK** on the Advanced Options window and again on the SASApp - Stored Process Server Properties window.
   h. Stop and then restart your SAS 9.4 object spawner.

FINCEN CTR Implementation

The following steps summarize the post-installation procedure to implement FINCEN CTR reporting. This procedure assumes that the post-installation process described in Chapter 4, “Post-installation Requirements and Tasks,” on page 31 is complete.

1. Load CTR configuration data. Follow the instructions in “Loading the SAS Enterprise Case Management Configuration Tables” on page 40 to load loadrr_fincen_new.sas, loadrr_fincen_new_base.sas, and loadrr_config_fincen_newctr.sas, if they have not been loaded. If you have upgraded from SAS Enterprise Case Management 3.1 or 2.3, loadrr_fincen_new_base.sas can be skipped.
2. Clear the cache. Follow the instructions in “Clearing the Cache” on page 40 to clear the old configuration data in memory.
3. Upload the user interface definitions. Follow the instructions in “Uploading User Interface Definitions” on page 41 to upload rr-fincen-ctr-02.xml and efile-fincen-new-01.xml, if they have not already been uploaded.

4. Upload the workflow definitions. Follow the instructions in "Uploading Workflow Definitions" on page 42 to upload FINCENReport.xml, if the workflow has not been uploaded.
5. Edit ecm_global_mvar.sas. The following parameters defined in ecm_global_mvar.sas under ***SAS-configuration-directory*/Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros** should be specified for CTR e-filing:
   - ecm_webusername — The user name for connecting to the middle tier. This user is required to have Edit Report capability.
   - ecm_webpassword — The password of the user account for ecm_webusername. This password can be encoded with SAS PROC PWENCODE.

   If you prefer not to use a specific user for e-filing, you can leave these parameters empty. The credential of the user who is generating the e-file will be used. That user is required to have the Edit Report capability. If the user does not have that capability, the transaction sequence numbers generated during the e-file generation process cannot be added to the reports.
6. Set the user permission to the e-file directory. Follow the instructions in "Setting User Permission to the E-File Directory" on page 39 to enable the SAS stored process to create e-files in ***SAS-configuration-directory*/Lev<num>/Applications/SASEnterpriseCaseManagement/6.2/Source/efiles**.

E-Filing Process

The following sequence describes the typical e-filing process.

1. Create a report container. A report container is a case or incident that is configured to contain one or more reports.
2. Collect data in the reporting container. If there is any subject involved in the case or incident, add the subjects and save.
3. Create a report under a report container. To create a report, click **Add Report** and select the report type. Depending on the workflow of the case, **Add Report** may be hidden until the case is ready for filing the report.
4. Collect data in the report. Depending on the report UI definition, the information in the report container can be automatically transferred to the report. Enter all missing data into the report and save it. **Save Draft** enables you to save the report without conforming to the validation rules. If you click **Save**, warnings are displayed when validation rules are violated. Edit the report until all the warnings are resolved.
5. Preview the report. At any point during the collection of data for a report, you can preview the e-file report in the government form by clicking **Preview Report**.
6. Review and mark the report as ready for submission. Click **Save** to validate the report. After it is validated, you can submit it for managerial review from the **Action Items** panel by selecting **Submit for Review** from the drop-down menu under **Activity Status**. Your manager can then set the activity status to either **Return to Edit** or **Ready to Submit**. The action items might vary depending on the workflow configuration. In the sample SAR UI and sample workflow, the report prepared date (X_PREPARED_DT) is set to the current date when the report is marked as **Ready to Submit**.

7. Create an ad hoc e-file. Ad hoc e-files can be created through the e-file UI, and reports marked **Ready to Submit** can be selectively added to the e-file. For the steps to create an e-file, see the *SAS Enterprise Case Management: User's Guide*. Click **Generate E-File** to generate a submission-ready e-file. If it is successful, the name and other information about the submission-ready e-file is displayed.
8. Send the e-file to the government. Go to the e-file repository, which is typically configured under ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\efiles** to look for the submission-ready e-file, and then submit it to the government. In additional, for FINCEN e-files, the e-file report sequence numbers should be populated in the E-File Report table.
9. Update the e-file and report submission status and information. After you have submitted the e-file to the government, update the **Post-filing Details** panel of the e-file with information, such as the confirmation ID that you received from the government website. Also, update the status of the e-file from the drop-down list on the **EFile Details** panel. The reports in the e-file must also have the **Activity Status** of the **Action Items** panel updated to indicate that they have been sent to the government.
10. Edit rejected reports. If the e-file is rejected because of errors, edit the reports by changing or removing bad reports. After you have done that, override the entire e-file and resubmit it.
11. After a report has been accepted by the regulatory agency, new information or an error might be discovered that changes the report. This change needs to be reported by filing a correction report. The original report is used as the basis for this correction report, which follows the process as if it were a new report. A report reference number, which is either generated by SAS Enterprise Case Management or provided by the regulatory agency, is required to identify which report in the regulatory agency computer system to amend. Then, create a new e-file with the new correction reports.

Configuring E-Filing

Steps

Configuring e-filing involves the following steps:

1. Configure user-defined fields. Define any custom fields for a report that will be necessary to collect data for e-filing.
2. Configure the UI definition of the report container.
3. Configure the report UI definition.
4. Configure the report workflow definition.
5. Configure the e-file user-defined field.
6. Configure the e-file UI definition.
7. Configure the form. For more information, see "Regulatory Report Form Configuration" on page 177.

Configuring Report User-Defined Fields

To collect data for the eventual e-file, you must define fields that will capture this data. For example, if "SAR involved amount" is a field that will be needed for the e-file, then it might be necessary to define a custom field named X_SAR_INVOLVED_AMT that is applicable to reports. This custom field should then be added to the UI definition. See "Configuring the UI Definition of the Report Container " on page 174.

After the custom field is defined and added to the UI definition, values collected for this field are stored with the report. Refer to the following for sample SAR custom fields and the lookup tables that are useful for the SAR report UI definition:

- Windows platforms: ***SAS-configuration-directory*\Lev*<num>*\Applications\SASEnterpriseCaseManagement\6.2\Source\form_templates\rr_udf_def_sar.csv**
- UNIX platforms: ***SAS-configuration-directory*/Lev*<num>*/Applications/SASEnterpriseCaseManagement/6.2/Source/form_templates/sar-efile_field_config_ref.csv**

For similar CTR samples, refer to the following:

- Windows platforms: ***SAS-configuration-directory*\Lev*<num>*\Applications\SASEnterpriseCaseManagement\6.2\Source\form_templates\rr_udf_def_newctr.csv**
- UNIX platforms: ***SAS-configuration-directory*/Lev*<num>*/Applications/SASEnterpriseCaseManagement/6.2/Source/form_templates/newctr-efile_field_config_ref.csv**

Configuring the UI Definition of the Report Container

Cases or incidents that can contain reports must be configured to use a UI definition that contains a ReportsTable component. The button for adding a report can be conditionally enabled or disabled based on the evaluation of an expression. This is accomplished by adding a **readonly** parameter to the ReportsTable component. The following are examples of how to set the **readonly** parameter based on workflow activity.

```
<tab id="regulatoryReportTab">
    <label><message key="reporting.header.txt" /></label>
     <field type="component" required="false" component-name="ReportsTable"
       readonly="!isWorkableActivity('File Report')">
       <param name="objectName" value="'RR'" />
      <param name="field" value="'RR_ID'" />
      <param name="field" value="'RR_DESC'" />
      <param name="field" value="'RR_STATUS_CD:RT_RR_STATUS'" />
      <param name="field" value="'CREATE_DTTM'" />
  </field>
  </tab>
```

For more information about ReportsTable definitions, refer to the Custom Page Builder documentation found on the **Administration** menu of SAS Enterprise Case Management.

Configure the Report UI Definition

Report UI definitions can gather any user-defined values that are needed, but should have a section that displays or gathers any information needed for the RR_LIVE table entry. They should also contain a History tab section.

The following is a sample report data section:

```
<tab id="reportTab">
 <label><message key="tab.report.details.header.txt" /></label>
 <column-layout>
                <column width="50%">
             <field name="RR.RR_ID" type="hidden"/>
        <field name="RR.PARENT_OBJECT_NM" type="hidden"/>
             <field name="RR.RR_DESC" type="string">
                <label><message key="field.rr.rr_desc.label.txt" /></label>
         </field>
         <field name="RR.RR_STATUS_CD" type="dropdown" readonly="true"
           values="GetLabelValues('RT_RR_STATUS')">
                 <label><message key="field.rr.rr_status_cd.label.txt" /></label>
             </field>
         <field name="RR.RR_TYPE_CD" type="dropdown" readonly="true"
           values="GetLabelValues('RT_RR_TYPE')">
                <label><message key="field.rr.rr_type_cd.label.txt" /></label>
         </field>
         <field name="RR.RR_CATEGORY_CD" type="dropdown" readonly="true"
           values="GetLabelValues('RT_RR_CATEGORY')">
               <label><message key="field.rr.rr_category_cd.label.txt" /></label>
             </field>
         <field name="RR.RR_SUBCATEGORY_CD" type="dropdown" readonly="true"
           values="GetLabelValues('RT_RR_SUBCATEGORY')">
               <label><message key="field.rr.rr_subcategory_cd.label.txt" /></label>
             </field>
         <field name="RR.CORRECTION_FLG" type="boolean" default="false">
                <label><message key="field.rr.x_correct_prior_rpt_flg.label.txt" />
       </label>
         </field>
         <field name="RR.RR_AGENCY_REF_ID" type="string" readonly="true">
                <label><message key="field.rr.rr_agency_ref_id.label.txt" /></label>
             </field>
         <field name="RR.X_PREPARED_DT" type="date" readonly ="true">
                <label><message key="field.rr.x_prepared_dt.label.txt" /></label>
             </field>
         <if test="RR.PARENT_OBJECT_NM = 'CASE'">
              <field type="component" name="RR.PARENT_OBJECT_RK"
component-name="ReportParentLink">
                  <label><message key="field.rr.parent_case.label.txt" /></label>
              </field>
         </if>
         <if test="RR.PARENT_OBJECT_NM = 'INCIDENT'">
               <field type="component" name="RR.PARENT_OBJECT_RK"
component-name="ReportParentLink">
                  <label><message key="field.rr.parent_incident.label.txt" /></label>
               </field>
         </if>
```

```
        </column>
      </column-layout>
    </tab>
```

The following is a sample History tab section:

```
<tab id="historyTab">
            <label><message key="tab.report.report.history.header.txt" /></label>
               <field type="component" component-name="ReportEventTable" />
</tab>
```

For more information about creating UI definitions, see "Overview of the Custom Page Builder" on page 98.

Configure the Report Workflow Definition

Workflow can be used to move the report from stage to stage until it is ready to be included in an e-file. After the data for the report has been collected, review and use the workflow component to mark the report as ready for managerial review by selecting **Submit for Review** from the drop-down menu under **Activity Status**. Your manager can then set the activity status to either **Return to Edit** or **Ready to Submit**.

Configure the E-file User-Defined Field

To maintain the transmitter information that will be incorporated into the e-file, you can define a user-defined field for each transmitter data element in an e-file record. The user-defined fields that can be used for FINCEN transmitter records are defined in efile_udf_def_fincen.csv, found in ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template**.

Configure the E-File UI Definition

E-files that contain reports must be configured to use a UI definition that contains the EFileReportTable component. This component allows for the updating of the report status. The UI definition also contains the **Generate E-file** button.

The e-file UI definition can be configured to automatically populate the transmitter information with a data loader. The ecmrr_efile_data_loader stored process is a data loader that is shipped with SAS Enterprise Case Management. It calls the ecmrr_efile_*government agency code*_trnsmttr_var macro to get the necessary information. The *government agency code* is specified in the FORM_AGENCY_CD column in the FORM_CONFIG table. Refer to "Define the Regulatory Report Form in FORM_CONFIG" on page 178 for more information about this field.

An example of ecmrr_efile_fincen_trnsmttr_var.sas can be found in ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\ucmacros**.

Note: The macro assumes that SAS Institute is the transmitter. Make sure that you edit the macro to include the correct transmitter information.

Enable Report Update in the E-File Stored Process

Based on certain regulatory requirements, during the e-file generation process the ECM web service might be called to update the reports under the e-file with e-file-related

information, such as transaction sequence number and signature date. With only the e-file default transmitter role, the user is not able to update any report. To resolve this issue, choose one of the following two options:

- Add the Edit Report capability to the e-file transmitter role and save the transmitter's user account and password to the SAS Metadata Server.
- Edit ecm_global_mvar.sas in ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\ucmacros** to provide a user ID and password that has the capability to edit reports and e-files. The parameters are ecm_webusername and ecm_webpassword.

Regulatory Report Form Configuration

Steps

As described earlier in this chapter, regulatory report forms must be defined before any regulatory reporting to the government can be done. The complete regulatory report form configuration involves two major setup processes:

1. paper report generation for regulatory report preview
2. electronic file generation for batch e-file submission

SAS Enterprise Case Management 6.2 is shipped with two forms as samples: FINCEN/SAR and FINCEN/NEWCTR. In summary, these are the basic steps for setting up a regulatory report form:

Note: All sample code referenced is based on the March 2013 FINCEN reports.

1. Define the regulatory report form configuration data in FORM_CONFIG.
2. Define the SAS macro to generate the regulatory report form for regulatory report PREVIEW.
3. Define the SAS program to transform the regulatory report data into regulatory report form data.
4. Define the regulatory report form template.
5. Define the regulatory report form section configuration data in FORM_PREVIEW_SECTION_CONFIG.
6. Define the regulatory report form field configuration data in FORM_PREVIEW_FIELD_CONFIG.
7. Define the SAS macro to format source data for form preview.
8. Define the program to generate the e-file
9. Define the SAS macro to transform the regulatory report data into regulatory report e-file-ready data.
10. Define the e-file record configuration in FORM_EFILE_RECORD_CONFIG.
11. Define e-file field configuration in FORM_EFILE_FIELD_CONFIG.

Define the Regulatory Report Form in FORM_CONFIG

A record in FORM_CONFIG is required for each regulatory report form. FORM_CONFIG contains the following columns:

FORM_CONFIG_RK
is the unique numeric number to identify the form.

FORM_AGENCY_CD
is the unique code to define the government agency that published the form.

FORM_TYPE_CD
is the unique code to define the government form published by the agency defined in FORM_AGENCY_CD.

Note: FORM_AGENCY_CD and FORM_TYPE_CD must be unique within the effective date range.

FORM_COUNTRY_CD
is the three-character country code of the form.

EFFECTIVE_FROM_DTTM
is the start date and time when the form is in use. This field is used for documentation purposes only. It is not used to determine whether the form is effective or not.

EFFECTIVE_TO_DTTM
is the end date and time when the form is in use. Use NULL to indicate that the form is the one currently in use.

Note: There can be only one record for FORM_AGENCY_CD and FORM_TYPE_CD with NULL as the EFFECTIVE_TO_DTTM.

FORM_DESC
is the form description.

PREVIEW_MIME_TYPE_CD
is the standard mimetype of the preview report.

PREVIEW_FILE_PREFIX
is the common prefix of all preview templates. For more information, see "Define the Regulatory Report Form Layout" on page 179.

PREVIEW_DRIVER_PGM_NM
is the SAS macro to be called by the ECMRR_PREVIEW stored process. For more information, see "Define the SAS Macro to Generate the Regulatory Report Form for Regulatory Report Preview" on page 179.

PREVIEW_PREPROCESS_PGM_NM
is the SAS macro to be called by PREVIEW_DRIVER_PGM_NM for transforming the regulatory report data into form-ready data. For more information, see "Define the SAS Macro to Generate the Regulatory Report Form for Regulatory Report Preview" on page 179.

EFILE_FILE_PREFIX
is the name of the output folder that will be used to keep the output e-files.

EFILE_DRIVER_PGM_NM
is the SAS macro to be called by ECMRR_EFILE_DRIVER for transforming the regulatory report data into e-file-ready data. For more information, see "Define the SAS Macro to Generate the E-File" on page 181.

EFILE_MAX_RR_CNT
is the maximum number of reports that can be added to an e-file without reaching the e-file size limit.

CREATE_USER_ID
is the ID of the user who added the record.

CREATE_DTTM
is the date and time when the record was added.

UPDATE_USER_ID
is the ID of the user who updated the record.

UPDATE_DTTM
is the date and time when the record was updated.

DELETE_FLG
is the flag indicating whether the record is active.

Define the SAS Macro to Generate the Regulatory Report Form for Regulatory Report Preview

This macro is the driver program that is called by the ECMRR_PREVIEW stored process. The two input parameters are FORM_CONFIG_RK and RR_RK. The output file must be _WEBOUT and a correct mime type must be set. The program should be written in a generic way so that it takes advantage of the information defined in Steps 3 through 6. See ecmrr_prvw_fincen_driver.sas for reference.

Define the SAS Macro to Transform Regulatory Report Data into Regulatory Report Form Data

The purpose of the program is to transform the regulatory report data into a format that is ready to be used for step 5. This works closely with the definition of the mapping defined in "Define the Regulatory Report Form Section Configuration Data in FORM_PREVIEW_SECTION_CONFIG". See ecmrr_prvw_fincen_sardi.sas for reference.

Note: You must use a valid folder name.

Define the Regulatory Report Form Layout

Define the regulatory report form template that controls the layout of the preview report. The form template should contain form field definitions that can be substituted with the actual regulatory report data values. For example, define a PDF form with PDF form fields, or define an HTML form with special field tags. Make sure that the file extension of the form template matches the mime type of the final output. For example, use .pdf for application/pdf. Then, save the template in ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template** and specify the file name of the template without the file extension in FORM_CONFIG.PREVIEW_FILE_PREFIX.

For the output involving multiple form templates, define one template for each output section and create with a common file prefix for all the form templates. Then specify only the common file prefix in FORM_CONFIG.PREVIEW_FILE_PREFIX. The file suffix of the form template is defined in FORM_PREVIEW_SECTION_CONFIG.SECTION_FILE_SUFFIX.

Define the Regulatory Report Form Section Configuration Data in FORM_PREVIEW_SECTION_CONFIG

FORM_PREVIEW_SECTION_CONFIG controls the sections of the output file. There should be one record for each regulatory report form template defined in "Define the Regulatory Report Form Layout". FORM_PREVIEW_SECTION_CONFIG contains the following columns:

FORM_CONFIG_RK
is a form key defined in FORM_CONFIG.

FORM_PREVIEW_SECTION_SEQ_NO
is the sequence number to uniquely identify the form templates of a form.

FORM_PREVIEW_SECTION_ID
is the ID to uniquely identify each form template.

SECTION_DESC
is the form template description.

SECTION_ORDER_NO
is the order in which the form templates are placed into the final output.

SECTION_FILE_SUFFIX
is the suffix of the file name of the form template.

SOURCE_TABLE_NM
is the name of the source table to be used to populate the form template. This source table should be a SAS table generated by the program defined in "Define the SAS Macro to Transform Regulatory Report Data into Regulatory Report Form Data" on page 179.

Define the Regulatory Report Form Field Configuration Data in FORM_PREVIEW_FIELD_CONFIG

FORM_PREVIEW_FIELD_CONFIG defines the mapping of the source data fields in FORM_PREVIEW_SECTION_CONFIG.SOURCE_TABLE_NM to the form fields in the form template defined in "Define the SAS Macro to Transform Regulatory Report Data into Regulatory Report Form Data" on page 179. There should be one record for each form field in the form template. FORM_PREVIEW_FIELD_CONFIG contains the following columns:

FORM_CONFIG_RK
is defined in FORM_PREVIEW_SECTION_CONFIG.

FORM_PREVIEW_SECTION_SEQ_NO
is defined in FORM_PREVIEW_SECTION_CONFIG.

FORM_PREVIEW_FIELD_SEQ_NO
is the unique sequence number to identify the form fields in the form template.

SOURCE_FIELD_NM
is the name of the field in FORM_PREVIEW_SECTION_CONFIG.SOURCE_TABLE_NM to be used for populating the form fields.

SOURCE_FIELD_NUM_FLG
indicates whether the source field or the result of the source field expression is numeric or not.

SOURCE_FIELD_EXP
is the expression to transform the source field.

TARGET_FIELD_NM
is the name of the form field defined in the form template.

TARGET_FIELD_TYPE_CD
is the code for each type of form field. A macro program should be written for each TARGET_FIELD_TYPE_CD to transform the data value into the appropriate output format.

TARGET_FIELD_TYPE_PARM_LIST
is the list of '|' separated parameters to be passed to the macros defined in "Define the SAS Macro to Format Source Data for Form Preview".

Define the SAS Macro to Format Source Data for Form Preview

A SAS macro should be written for each unique FORM_PREVIEW_FIELD_CONFIG.TARGET_FIELD_TYPE_CD. The list of parameters should be P1, P2, P3, and so on, depending on the number of parameters defined in FORM_PREVIEW_FIELD_CONFIG.TARGET_FIELD_PARM_LIST. The name of the macro depends on how the driver program is written. For example, in ECMRR_PRVW_FINCEN, the macro name is %ECMRR_FDF_<*target_field_type_cd*> because ECMRR_PRVW_FINCEN is generating fdf data for the PDF file. %ECMRR_FDF_PGM_GEN should be run whenever FORM_PREVIEW_FIELD_CONFIG is updated.

Define the SAS Macro to Generate the E-File

This is the driver program that is called by the ECMRR_EFILE stored process. The input parameter is EFILE_RK. The output file must be saved in the folder defined in FORM_CONFIG.EFILE_FILE_NM_PREFIX. The program should be written in a generic way that takes advantage of the information defined in Steps 10 and 11. See ecmrr_efile_fincen_driver.sas for reference.

Define the SAS Macro to Transform Regulatory Report Data into E-File-Ready Data

The purpose of the program is to transform the regulatory report data into a format that is ready to be used for Step 8. See ecmrr_efile_fincen_sardi.sas for reference.

Define the E-File Record Configuration Data in FORM_EFILE_RECORD_CONFIG

FORM_EFILE_RECORD_CONFIG controls the records of the output file. There should be one record for each record type defined by the government agency. FORM_EFILE_RECORD_CONFIG contains the following columns:

FORM_CONFIG_RK
is the form key defined in FORM_CONFIG.

FORM_EFILE_RECORD_SEQ_NO
is the sequence number to uniquely identify the record type of an e-file.

FORM_EFILE_RECORD_ID
is the ID to be used to uniquely identify the record type of an e-file.

FORM_EFILE_RECORD_DESC
is the e-file record description.

RECORD_SORT_ORDER_NO
is the order in which the records are placed into the e-file.

RECORD_LENGTH_NO
is the e-file record length.

SOURCE_TABLE_NM
is the name of the source table to be used to populate the e-file record.

PAGE_BY_FIELD_NM
is the name of the field for grouping the depending records together. For FINCEN, the page types are batch, institution, branch, and regulatory report. The fields to identify the page types are Z_<page_type>_SEQ_NO.

Define the E-File Field Configuration Data in FORM_EFILE_FIELD_CONFIG

FORM_EFILE_FIELD_CONFIG defines the mapping of the source data fields in FORM_EFILE_RECORD_CONFIG.SOURCE_TABLE_NM to the target fields in the e-file. FORM_EFILE_FIELD_CONFIG contains the following columns:

FORM_CONFIG_RK
is defined in FORM_PREVIEW_SECTION_CONFIG.

FORM_EFILE_RECORD_SEQ_NO
is defined in FORM_EFILE_RECORD_CONFIG.

FORM_EFILE_FIELD_SEQ_NO
is the unique sequence number to identify the data fields in the e-file.

TARGET_FIELD_START_POS_NO
is the start position of the e-file field published by the government agency.

TARGET_FIELD_END_POS_NO
is the end position of the e-file field published by the government agency.

TARGET_FIELD_NM
is the name of the e-file field published by the government agency.

TARGET_FIELD_LENGTH
is the length of the e-file field published by the government agency.

TARGET_FIELD_DESC
is the description of the e-file field published by the government agency.

SOURCE_FIELD_NM
is the name of the field in FORM_EFILE_RECORD_CONFIG.SOURCE_TABLE_NM to be used for populating the form fields.

SOURCE_FIELD_NUM_FLG
indicates whether the source field or the result of the source field expression is numeric or not.

SOURCE_FIELD_EXP
is the expression to transform source fields.

SAS_FORMAT_NM
is the name of the SAS format to be used to write the source field or the result of the source field expression. If the field is missing, the default format is used. That means

BEST for numeric fields and $ for character fields. The SAS formats used for FINCEN are defined in SAS macro ecmrr_fincen_sas_format.sas.

Support of Multiple Versions of Report Forms

Sometimes the government makes changes to the form submission requirements. If the changes involve more vigorous validation, the UI definitions should be updated and no new form is required. However, if the changes involve a new preview form or a new e-file layout, a new form should be created with the same FORM_AGENCY_CD and FORM_TYPE_CD.

SAS Enterprise Case Management looks up the FORM_CONFIG table by FORM_AGENCY_CD, FORM_TYPE_CD and EFFECTIVE_TO_DTTM to assign FORM_CONFIG_RK to a newly created report or e-file. All existing reports and e-files have the original FORM_CONFIG_RK. That means SAS Enterprise Case Management continues to generate PREVIEW reports and e-files following the old standard. To avoid mixing reports with different FORM_CONFIG_RK in the same e-file, it is a best practice to finish processing all outstanding reports before starting a new version of the government form.

To add new fields to a report form:

1. Edit the form template. Use an appropriate editor to add new form fields in the template. For a PDF template, Adobe Acrobat Pro software can be used. The template can be found in ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template**.
2. Add records to the FORM_PREVIEW_FIELD_CONFIG table to define the mappings of the source data fields to the new preview form fields.
3. Run %ECMRR_FDF_PGM_GEN.
4. Review or edit the macro for transforming regulatory report source data into regulatory report form data (for example, %ECMRR_PRVW_FINCEN_SAR_DATA for SAR) to make sure that the required source fields are derived properly.
5. Edit FORM_EFILE_FIELD_CONFIG to define the new mappings of the source fields to the e-file fields.
6. Review or edit the macro for transforming regulatory report source data into regulatory report e-file data (for example, %ECMRR_EFILE_FINCEN_SAR_DATA for SAR) to make sure that the required source fields are derived properly.
7. Run %ECMRR_EFILE_PGM_GEN (FORM_CONFIG_RK=<*form_config_rk*>) with the correct FORM_CONFIG_RK.

FINCEN

As described in the previous sections, SAS Enterprise Case Management is highly configurable to support different regulatory reports. Sample user-defined fields, UI forms, workflows, and SAS macros for FINCEN, SAR, and CTR are shipped in the

solution. The following is a summary of the sample files that are common in most FINCEN reports.

File	Location
ecmrr_config_fincen_new.sas	Windows: ***!SASROOT*\casemgmtmva\sasmisc\sample\config** UNIX: ***!SASROOT*/misc/casemgmtmva/sample/config**
ecmrr_config_fincen_new_base.sas	Windows: ***!SASROOT*\casemgmtmva\sasmisc\sample\config** UNIX: ***!SASROOT*/misc/casemgmtmva/sample/config**
ecmrr_prvw_fincen_*.sas	Windows: ***!SASROOT*\casemgmtmva\ucmacros** UNIX: ***!SASROOT*/ucmacros/casemgmtmva**
ecmrr_fincen_*.sas	Windows: ***!SASROOT*\casemgmtmva\ucmacros** UNIX: ***!SASROOT*/ucmacros/casemgmtmva**
ecmrr_efile_fincen_*.sas	Windows: ***!SASROOT*\casemgmtmva\ucmacros** UNIX: ***!SASROOT*/ucmacros/casemgmtmva**
case-rr-fincen-01.xml	Windows: ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions** UNIX: ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions**
efile-fincen-01.xml	Windows: ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions** UNIX: ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions**
fincenREPORT.xml	Windows: ***SAS-middle-tier-installation-directory*\deploy\sample\Workflows** UNIX: ***SAS-middle-tier-installation-directory*/deploy/sample/Workflows**

ecm_soap_config.xml	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**
efile_udf_def_fincen.csv	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**
2007NAICS.csv	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**
state_code_list.csv	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**
country_names_and_code_elements.txt	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**

Note: The file country_names_and_code_elements.txt contains ISO standard country code. The file state_code_list.csv contains U.S., Canadian, and Mexican state and province codes based on the FINCEN requirement. These location codes are used throughout SAS Enterprise Case Management, not limited to FINCEN reporting.

SAR

This section provides a summary of the sample files specifically for SAR and instructions for using the UI interface.

SAR Files

The following files are needed to process the SAR:

File	Location
ecmrr_config_fincen_sar.sas	Windows: `!SASROOT\casemgmtmva\sasmisc\sample\config\` UNIX: `!SASROOT/misc/casemgmtmva/sample/config`
ecmrr_prvw_fincen_sar_*.sas	Windows: `!SASROOT\casemgmtmva\ucmacros` UNIX: `!SASROOT/ucmacros/casemgmtmva`
ecmrr_efile_finen_sar_*.sas	Windows: `!SASROOT\casemgmtmva\ucmacros` UNIX: `!SASROOT/ucmacros/casemgmtmva`
rr-fincen-sar-01.xml	Windows: `SAS-middle-tier-installation-directory\deploy\sample\UIDefinitions` UNIX: `SAS-middle-tier-installation-directory/deploy/sample/UIDefinitions`
sar_*.pdf	`SAS-configuration-directory\Lev<num>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template`
sar_*.csv	`SAS-configuration-directory\Lev<num>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template`

SAR Report UI

SAS Enterprise Case Management is used to enter data into several objects required by the SAR form.

SAR requires institution and branch detail information to be reported. Defined in the sample rr-fincen-sar-01.xml, institution and branch data is available in look-up tables. See "User-Defined Generic Data Tables " on page 69 for a description of the definition of these tables.

Only the financial institutions and branches that were opened after the report container creation date are available for selection. To see the list of institutions, always save the new report container at least once. When an institution is selected, the institution detail is populated to the report. If a branch is involved, click **No** in the **If no branch is involved, choose Yes** field, to reveal all of the branch fields.

Due to the potential length of the branch list, rr-fincen-sar-01.xml uses the type-ahead feature for branch selection. That means you can enter the first few digits of the branch

code to shorten the list of available branches. When a branch is selected, the branch and contact information is populated. If necessary, change the contact information.

Note: Branch code is used only for branch look-up. It is not used for e-filing. A sequential number is assigned to each branch and contact combination as branch code when an e-file is generated.

To add suspects to a report, click **Add Subject**. A list of subjects associated with the report container is displayed at the top of the subject UI. Subject information will be populated to the subject UI. Enter the missing information or correct the populated content, and click **OK**.

Note: Subject information entered in the subject UI is saved in the report only. It does not affect the subject of the report container. If the subject is not part of the report container, enter the subject information manually.

To add an attachment to the SAR report for e-filing, follow these steps:

1. Click **Attachment** open the Attachments window.
2. Click **Browse** to locate the file that needs to be uploaded. FINCEN rejects any CSV file larger than one megabyte. If desired, enter the attachment description. Click **Commit Attachment**, and then click **Close** to close the window.
3. Click **Save** on the Report window to save the attachment to the report. You should see the attachment count go up.
4. On the **Report Detail** tab, select the CSV file from the **Attachment to E-File** menu. Then save the report again. This step only specifies the name of the CSV file to be used for e-filing. The actual content of the CSV file is extracted when the e-file is generated.
5. If you want to modify the attachment, repeat steps 1-3 to upload a modified version of the CSV file.
6. If you want to use a different attachment, repeat step 1-4 to upload a new CSV file.

CTR

The following is a summary of the sample files specifically for CTR and instructions for using the UI interface.

CTR Files

The following files are needed to process the CTR:

File	Location
ecmrr_config_fincen_newctr.sas	Windows: `!SASROOT\casemgmtmva\sasmisc\sample\config\` UNIX: `!SASROOT/misc/casemgmtmva/sample/config`

ecmrr_prvw_fincen_newctr_*.sas	Windows: ***!SASROOT*\casemgmtmva\ucmacros** UNIX: ***!SASROOT*/ucmacros/casemgmtmva**
ecmrr_efile_finen_newctr_*.sas	Windows: ***!SASROOT*\casemgmtmva\ucmacros** UNIX: ***!SASROOT*/ucmacros/casemgmtmva**
rr-fincen-ctr-02.xml	Windows: ***SAS-middle-tier-installation-directory*\deploy\sample\UIDefinitions** UNIX: ***SAS-middle-tier-installation-directory*/deploy/sample/UIDefinitions**
newctr_*.pdf	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**
newctr_*.csv	Windows: ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\form_template** UNIX: ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/form_template**

CTR Report UI

SAS Enterprise Case Management is used to enter data into several objects required by the CTR form.

Similar to SAR, CTR requires institution detail information to be reported. Defined in the sample rr-fincen-ctr-02.xml, institution data is available in look-up tables. See "User-Defined Generic Data Tables " on page 69 for a description of the definition of these tables.

Only the financial institutions that were opened after the report container creation date are available for selection. To see the list of institutions, always save the new report container at least once. When an institution is selected, the institution detail is populated to the report.

To add a person involved in the transaction, click **Add Person**. Enter the other information, and save the report.

Chapter 9

Related Items

Overview 189
Configuring Match Criteria 189

Overview

This chapter describes how to configure the match criteria that determine which cases and incidents are related to unassigned incidents (incidents not associated with a case). For a description of the process that users follow to find related items, see “Finding Items Related to Unassigned Incidents” in the *SAS Enterprise Case Management: User's Guide*.

By default, one match criterion (NATIONAL_ID) is defined. You can define additional match criteria when you configure SAS Enterprise Case Management.

Configuring Match Criteria

The match criteria for related items are defined in ECM_DB.RELATED_ITEM_CONFIG. The following table describes the columns for configuring related items:

Table 9.1 *Related Item Configuration Match Criteria*

Column	Description
RELATED_ITEM_RK	Unique key of the matching criterion.
RELATED_ITEM_ID	32-character field to name the matching criterion.
RELATED_ITEM_DESC	100-character fields to describe the matching criterion.
RELATED_PARTY_FIELD_NM	Name of the subject field that will be used to match the subjects. It can be any character field in PARTY_LIVE or defined in PARTY_UDF_DEF.

Column	Description
RELATED_INCIDENT_FIELD_NM	Name of the incident field that will be used to match the incidents. It can be any character field in INCIDENT_LIVE or defined in INCIDENT_UDF_DEF.
RELATED_CASE_FIELD_NM	Name of the case field that will be used to match the cases. This field is not currently supported.
CREATE_USER_ID	ID of the user who added this RELATED_ITEM_CONFIG record.
CREATE_DTTM	Date/time when this RELATED_ITEM_CONFIG record is added.
UPDATE_USER_ID	ID of the user who updated this RELATED_ITEM_CONFIG record.
DELETE_FLG	Flag indicating whether the record is active.

RELATED_ITEM_CONFIG contains one record with RELATED_PARTY_FIELD_NM=‘NATIONAL_ID’. Cases and incidents that have subjects with the same national ID as the subjects associated with an unassigned incident are considered to be related. Different match paths are taken when one or more of the RELATED_<ENTITY_NAME>_FIELD_NM field names are defined. The following table describes how different match paths are taken when PARTY_RELATED_FIELD_NM or INCIDENT_RELATED_FIELD_NM are specified.

Table 9.2 *Different Match Paths for Related Items*

Match Paths	Required Related Fields	Description
I-I_C	INCIDENT	1. Find related incident field values of the selected incident. 2. Return incidents with the same related incident field values. 3. Return associated cases of the related incidents.
I _P-P_C or I_P-P_I	PARTY	1. Find subjects associated with the selected incident. 2. Find related party field values of the associated subjects. 3. Find all subjects with the same party field values. 4. Return incidents and cases associated with the related subjects.
I-P_I or I-P_C	INCIDENT and PARTY	1. Find related incident field values of the selected incident. 2. Find subjects with related subject field values matching related incident field values in step one. 3. Return all incidents and cases associated with the related subjects.

Match Paths	Required Related Fields	Description
I_P-P_I-I or I_P-P_I-C	INCIDENT and PARTY	1. Find subjects associated with the selected incident. 2. Find related party field values of the associated parties. 3. Return incidents with the related incident field values matching related party field values in step two. 4. Return cases associated with the related incidents.

The underscores (_) and hyphens (-) in the match paths represent direct associations and indirect relationships, respectively. I, P, and C in the match paths represent incident, party, and case. If multiple records are defined in RELATED_ITEM_CONFIG, they are handled as an OR condition. In other words, the overall result is a union of the results from each criterion. The following examples show how you can configure a related item to define three match criteria.

Criterion 1: Subject full name match
: Relate the subjects who have the same full name as any subject of the selected incident. The subject full name is the core field PARTY_FULL_NM in PARTY_LIVE.

Criterion 2: Account number of incident match
: Relate all the incidents that contain transactions with the same account number as that of the selected incident. The account number is defined in INCIDENT_UDF_DEF as UDF_TABLE_NM=X_TRANSACTION and UDF_NM=X_TRANSACTION_ACCT_NO.

Criterion 3: Driver license number of incident and driver license number of subject match
: Relate all the incidents that contain transactions with the same driver license number as that of the selected incident. Also relate all the subjects that contain the same driver's license number as the driver license number of transactions of the selected incident. The subject driver license is defined in PARTY_UDF_DEF as UDF_TABLE_NM=PARTY and UDF_NM=X_DRIVER_LICENSE_ID. The driver license number of an incident is defined in INCIDENT_UDF_DEF as UDF_TABLE_NM=X_TRANSACTION and UDF_NM=X_TRANSACTION_DL.

The following table shows how the RELATED_ITEM_CONFIG might look:

Table 9.3 *RELATED_ITEM_CONFIG*

RELATED_ITEM_RK	RELATED_ITEM_ID	RELATED_ITEM_DESC	RELATED_PARTY_FIELD_NM	RELATED_INCIDENT_FIELD_NM
1	FULL_NAME	Subject full name	PARTY_FULL_NM	
2	ACCT_NO	Incident account number		X_TRANSACTION_ACCT_NO
3	DRIVER_LICENSE	Subject and incident driver license	X_DRIVER_LICENSE_ID	X_ACCOUNT_DL

Note that the UDF_TABLE_NM is not included in the definition. All UDF fields with the same UDF_NM are used for matching. If that is not what you want, the field should be renamed.

Chapter 10

Financial Items

Overview 193
Defining Financial Item Types in a Reference Table 193
Defining the UDF for Financial Transactions 194
Defining the UDF for Financial Summaries 194
Adding a FinancialItemsTable Component to a Case, Incident, or Subject User Interface 194
Defining the Financial Items User Interface 194
Customizing a SAS Stored Process to Compute Financial Summaries 195

Overview

A financial item is a dollar amount associated with a financial item type. SAS Enterprise Case Management displays the list of financial items within a case, incident, or subject and aggregates these amounts into financial summaries. This chapter discusses the steps needed to configure SAS Enterprise Case Management to support this feature.

Defining Financial Item Types in a Reference Table

All financial items are classified by the financial item types. Only the financial items with the same financial item type can be aggregated from transactional level to case or incident level. Financial item type can also affect the UI used for data entry. Therefore, it is critical that the right financial item types are defined up front.

Financial item types are specified in REF_TABLE_VALUE with REF_TABLE_VALUE='RT_FINANCIAL_ITEM_TYPE'. The program Load_post_install_data.sas provides an example for defining financial item type.

Defining the UDF for Financial Transactions

FINANCIAL_ITEM_* tables store the financial items at the transactional level. The core fields are included in FINANCIAL_ITEM_LIVE. New user-defined fields can be defined in FINANCIAL_ITEM_UDF_DEF.

Defining the UDF for Financial Summaries

Financial summaries aggregated from financial transactions should be stored with a case or an incident. Therefore, the numeric user-defined fields specific for total financial amount should be defined in CASE_UDF_DEF or INCIDENT_UDF_DEF, or both.

Adding a FinancialItemsTable Component to a Case, Incident, or Subject User Interface

For adding the FinancialItemsTable component to a case, incident, or subject, see the Custom Page Builder documentation found on the **Administration** menu of SAS Enterprise Case Management.

Defining the Financial Items User Interface

Sample UI definitions included with SAS Enterprise Case Management include fi-gen-01.xml and fi-sa-01.xml. These UI definitions are used when a user adds a financial item type to a case, incident, or subject. The FINANCIAL_ITEM_CONFIG table needs to be configured for each financial item type that you want to add to the system. The fi-gen-01.xml sample is a generic sample that includes base information for each financial item that is added to the database. You can choose to use the samples provided or configure your own files and associate them to the appropriate types. To configure your own, complete the following steps:

1. Enter the appropriate information in the FINANCIAL_ITEM_CONFIG table.
2. Enter the appropriate user-defined fields that will be referenced in the customized financial items UI definition type.
3. Enter the corresponding financial item types in the REF_TABLE_VALUE table.

Customizing a SAS Stored Process to Compute Financial Summaries

Financial summaries are computed by the ECM_FINANCIAL_SUMMARY_CALC stored process. The web UI passes the financial transactions as an XML file to the stored process, and the stored process returns the summaries as name-value pairs. For each name-value pair, the name is the name of the field in the UI definition that will be updated. The value must be either a numeric value or "." (the SAS missing value indicator). "." is interpreted as 0.0 in the summary. ECM_FINANCIAL_SUMMARY_CALC handles mainly the input and output formatting. The actual computation is done in the %ECM_FIN_SUM_DRIVER macro. This macro first sums up all the numeric columns and creates a table with the total of each financial item type. It then calls the %ECM_FIN_SUM_CUSTOM macro for more complex computation.

To customize the computation, copy ecm_fin_sum_custom.sas from one of the following locations:

- Windows: ***!SASROOT*\casemgmtmva\ucmacros**
- UNIX: ***!SASROOT*/ucmacros/casemgmtmva**

Paste it to ***SAS-configuration-directory*/Lev*<num>*/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros** with the same name. Then modify the section marked with 'Start custom code' and 'End custom code' in the new macro. All columns created in this macro are passed back to the web UI as name-value pairs. If the results are not populated properly on the screen, there might be a mismatch of column names in the macro and field names in the UI.

Chapter 11

Case Network Analysis

Overview 197
Case Network Analysis Process 197
Configuring Link Criteria 198
Configuring the Data Source 200
Configuring Displayed Data Fields and Link Filters 200
Configuring Display Labels 201
Case Network Analysis Logic 202

Overview

The Case Network Analysis graph provides a shallow analysis of the parties in the database, looking for parties that are related by demographic data such as national IDs, names, birth dates, and addresses. The Case Network Analysis component enables an investigator to identify a network of related parties and the cases, incidents, and reports that the parties have been involved in. The business motivation is to provide an investigator with a way of stepping back from an individual case and seeing broader patterns of behavior for a single party or a set of closely related parties.

The Case Network Analysis component enables users to generate a graph of a single selected party. After the initial graph has been rendered, an investigator can dynamically explore the network of parties, cases, incidents, and reports by expanding a node in the graph and “walking” to the other objects linked to that one.

This chapter discusses the configuration process for the Case Network Analysis component as well as how to configure the link criteria of the graph and the data fields to be displayed in the graph. Finally, it discusses the logic behind the analysis.

Case Network Analysis Process

The following steps show the process for defining the Case Network Analysis component:

1. The SAS Enterprise Case Management administrator defines the link criteria for Case Network Analysis.

2. The SAS Enterprise Case Management administrator defines the data fields and links to be displayed in the graph.
3. The user logs on to the SAS Enterprise Case Management web application to access the Case Network Analysis web component. Refer to the *SAS Enterprise Case Management: User's Guide* for instructions on how to do this.
4. The Case Network Analysis web component passes the surrogate key of the party to the SAS stored process getSocialNetwork.
5. The SAS stored process getSocialNetwork references the link criteria defined in step 1 and returns a list of nodes and links to the Case Network Analysis web component. The list also contains the node attributes such as node label and properties. The content of the property is the concatenation of the data field values of the node. The list of the data fields and other displayed content is configurable and is discussed in "Configuring Displayed Data Fields and Link Filters" on page 200.
6. The Case Network Analysis web component displays a Case Network Analysis graph with the resulting nodes and links.
7. If a Case Network Analysis graph is displayed, the user can refer to the *SAS Enterprise Case Management: User's Guide* to dynamically explore the graph. As the user explores the Case Network Analysis graph, there are two actions that will involve the SAS stored process to get more information from the database.

   Show details

   You can click on any node to get to its properties. Then, click **Show Details** to get the node detail. The Case Network Analysis web component passes the node key, note type, and user ID to the SAS stored process getSocialNetworkNodeDetail. getSocialNetworkNodeDetail then returns the node details, based on the user permission and the node type. The Case Network Analysis web component renders the node details in name-value pairs. The node details are displayed on the bottom pane. Only one node detail can be displayed at a time. The list of detail fields is configurable and is discussed in "Configuring Displayed Data Fields and Link Filters" on page 200.

   Expand the Case Network Analysis graph

   Users can expand the current Case Network Analysis graph by selecting a non-root party to run a new Case Network Analysis. To do that, the user can click on the plus sign (+) of the new party of interest. The Case Network Analysis web component then passes the surrogate key of the new party to the SAS stored process growSocialNetworkNode. growSocialNetworkNode uses the same logic used by getSocialNetwork to return a list of nodes and links. The Case Network Analysis web component renders the results by attaching the new nodes and links to the existing Case Network Analysis graph.

   Note: The + is always available for the leaf party node, regardless of whether the node can be further expanded or not.

Configuring Link Criteria

Two tables are used to configure Case Network Analysis link criteria: SNA_CONFIG_MASTER and SNA_CONFIG_DETAIL. SNA_CONFIG_MASTER is the master table for configuring the Case Network Analysis link criteria. Parties are considered as linked when one or many of these criteria are met. SNA_CONFIG_DETAIL is the detail definition of SNA_CONFIG_MASTER. It

contains one or many records of each SNA_CONFIG_MASTER record. A SNA_CONFIG_MASTER link criterion is considered as met when all of its associated SNA_CONFIG_DETAIL link criteria are met.

The SNA_CONFIG_MASTER table contains the following columns:

SNA_CONFIG_RK
is the unique key of the link criterion.

SNA_CONFIG_ID
is the 32-character field to name the link criterion.

SNA_CONFIG_DESC
is the 100-character field to describe the link criterion.

CREATE_USER_ID
is the ID of the user who added the SNA_CONFIG_MASTER record.

CREATE_DTTM
is the date and time when the SNA_CONFIG_MASTER record was added.

UPDATE_USER_ID
is the ID of the user who updated the SNA_CONFIG_MASTER record.

DELETE_FLG
is the flag indicating whether the record is active.

The SNA_CONFIG_DETAIL table contains the following columns:

SNA_CONFIG_RK
is the unique key of the link criterion.

SNA_CONFIG_SEQ_NO
is the secondary key to uniquely identify a SNA_CONFIG_DETAIL record.

FROM_PARTY_FIELD_EXP
is the expression to be used to define FROM_PARTY_FIELD_NM. If blank, FROM_PARTY_FIELD_NM is used for linking parties.

FROM_PARTY_FIELD_NM
is the name of the party field for linking parties. This is the 'link from' field.

FROM_PARTY_TABLE_NM
is the name of the table where FROM_PARTY_FIELD_NM is found.

TO_PARTY_FIELD_EXP
is the expression used to define TO_PARTY_FIELD_NM. If blank, TO_PARTY_FIELD_NM is used for linking parties.

TO_PARTY_FIELD_NM
is the name of the party field for linking parties. This is the 'link to' field.

TO_PARTY_TABLE_NM
is the name of the table where TO_PARTY_FIELD_NM is found.

The expression defined in FROM_PARTY_FIELD_EXP or TO_PARTY_FIELD_EXP must be a valid SAS expression. If you have very complicated logic, you might consider using PROC FCMP to create some user-defined functions. In SAS Enterprise Case Management, there is one record in SNA_CONFIG_MASTER and two associated records in SNA_CONFIG_DETAIL. The definition is for linking the parties when both the NATIONAL_ID_TYPE_CD and NATIONAL_ID fields match. Additional definitions can be found in the following locations:

Platform	Directory Path
Windows	*!SASROOT*`\casemgmtmva\sasmisc\sample\config\load_post_install_data.sas.`
UNIX	*!SASROOT*`/misc/casemgmtmva/sample/config/load_post_install_data.sas.`

Configuring the Data Source

Case Network Analysis involves the matching of subject data to the entire subject database. If your site has too many subjects, the browser may time out before the analysis is complete. To work around this performance problem, SAS Enterprise Case Management can be configured to use the report mart (such as tables in ECM_RPT) for subject matches. This is done by setting the global macro variable ECM_SNA_MATCH_LIB, which is defined in ecm_global_mvar.sas, to 'ECM_RPT'. This eliminates the need for transforming the live data into an analysis-ready format. This option affects the data source for matching the subject only, in order to speed up the match process. All labels and case, incident, subject, and report associations are still obtained from the live tables. Also, all data fields that are defined in SNA_CONFIG_DETAIL, as described in "Configuring Link Criteria " on page 198, have to be added to the report mart in SAS views. They are automatically created when the %ECM_REPORTING_DRIVER macro is executed. The views should be recreated by running the %ECM_REPORTING_XFORM macro whenever SNA_CONFIG_DETAIL is changed.

Configuring Displayed Data Fields and Link Filters

The SAS macro %ECM_SNA_GET_DETAIL_NODE_VARS defines the list of data fields to be used for node labels, node properties, and node details. It also defines the condition for filtering party relation types and member types to limit the scope of the graphs.

The following macro variables define the fields to be used as node labels in the graph:

- CASE_LABEL_VAR
- PARTY_LABEL_VAR
- INCIDENT_LABEL_VAR
- RR_LABEL_VAR

When the user has Write permission to the party, case, incident, and report record in the rest of the SAS Enterprise Case Management system, <*node_type*>_LABEL_VAR is used. If the user has only Read permission, <*node_type*>_ID is used. For information about how the user group permissions work, see "Configurations" on page 71, and refer to the appropriate subsection. The node label is limited to one field in <*node type*>_LIVE or user-defined fields in ECM_DB.<*node type*>_UDF_DEF. UDF_TABLE_NM equals <*node_type*>.

The following macro variables define the list of data fields to be included in the **Detail** tab. The order of the field list here is used for the initial display of the fields in the **Node Detail** tab. Users can change the fields to alphabetical order by clicking the column headings.

- PARTY_VAR_LIST_FULL
- INCIDENT_VAR_LIST_FULL
- CASE_VAR_LIST_FULL
- RR_VAR_LIST_FULL
- PARTY_VAR_LIST_SHORT
- INCIDENT_VAR_LIST_SHORT
- CASE_VAR_LIST_SHORT
- RR_VAR_LIST_SHORT

When the user has Write permission to the party, case, incident, or report record in the rest of the SAS Enterprise Case Management system, <*node_type*>_VAR_LIST_FULL is used. If the user has only Read permission, <*node_type*>_VAR_LIST_SHORT is used.

The following macro variables define the lists of data fields to be included in the party, incident, case, or report properties, respectively:

- PARTY_TOOLTIP_VAR_LIST
- INCIDENT_TOOLTIP_VAR_LIST
- CASE_TOOLTIP_VAR_LIST
- RR_TOOLTIP_VAR_LIST

Property fields are limited to the ECM_DB.PARTY_LIVE, ECM_DB.INCIDENT_LIVE, ECM_DB.CASE_LIVE, or ECM_DB.RR_LIVE tables respectively. Unlike other displayed fields, user permission is not checked for property content. Make sure that non-sensitive data fields are used in the properties.

The following macro variables define the WHERE clauses to be used to filter PARTY_X_PARTY, CASE_X_PARTY, and INCIDENT_X_PARTY tables:

- ECM_SNA_PXP_WHERE
- ECM_SNA_CXP_WHERE
- ECM_SNA_IXP_WHERE

The WHERE clauses have to use proper syntax and be properly quoted. Follow the examples in the macro to define the filters. The default condition is no filter.

Configuring Display Labels

Three types of labels are used in Case Network Analysis.

- Data column labels are defined in the custom.properties file and the content is populated into ECM_DB.FULL_ECM_COLUMN_LABEL_VIEW. These labels are used in the **Node Detail** tab.

- Match labels are defined in ECM_DB.FULL_REF_LABEL_TRANS with REF_TABLE_NAME="SNA_CONFIG_MASTER". Match labels are displayed as subject-to-subject link labels in the graph.
- **Detail** tab headers and column headings are defined in sashelp.entcm with key=GEN_<*various types*>_LABEL (for example, GEN_FIELD_LABEL and GEN_CASE_LABEL).

For information about updating the labels, see Chapter 14, "Internationalization," on page 211.

Case Network Analysis Logic

%ECM_SNA_DRIVER is the driver program for obtaining the nodes and links of the Case Network Analysis graph. The following process is used for the logic behind the analysis.

1. Include only the active link criteria in SNA_CONFIG_MASTER.
2. Include all parties for matching. To include only the parties that are associated with one or more cases, incidents, or parties, change the macro variable ASSOCIATED_PARTY_ONLY_YN in ECM_SNA_DRIVER from N to Y.
3. Transform the party data in step 2 into rectangle structure to include all core and UDF fields into one record.
4. Create views of the party data with derived fields defined in SNA_CONFIG_DETAIL.
5. For each active link criterion defined in SNA_CONFIG_MASTER, find the parties who are related to the root party by matching all data fields (that is, FROM_PARTY_FIELD_NM and TO_PARTY_FIELD_NM) defined in SNA_CONFIG_DETAIL.
6. Combine all the related parties found in step 5 to form the combined list of related parties. The graph is expanded to include PARTY_X_PARTY links. As a result, the related party list is not limited to parties that are found in step 5. It also includes the parties that are associated with the parties found in step 5.
7. Combine the list of associated cases of the related parties in step 6 with the list of associated cases of the related parties' associated incidents in step 6 to form the final list of case nodes.
8. Combine the list of the associated incidents of the cases in step 7 with the list of associated incidents of the related parties in step 5 to form the final list of incident nodes.
9. Obtain the list of associated parties of the cases and incidents in steps 7 and 8 and add this list of associated parties to the related parties list in step 6 to form the final list of party nodes.
10. Construct the links of the nodes based on their relationships found in steps 6, 7, 8, 9, and 10.

Chapter 12

Configuring Subject Search

Overview 203
Subject Search Process 203
Configuring Match Criteria for Subject Search 203
Subject Search Logic 204

Overview

This chapter describes how to configure and use the Subject Search component. Then it explains in detail how to configure the match criteria of the graph and the data fields to be displayed in the graph. Finally, it discusses the logic behind the search.

Subject Search Process

1. An administrator defines the match criteria for Subject Search.
2. A user passes an XML file with the subject information to the SAS Enterprise Case Management Subject Search web service.
3. SAS Enterprise Case Management Subject Search web service calls the ecm_subject_search stored process and reports the match results.

Configuring Match Criteria for Subject Search

There are two tables for configuring Subject Search match criteria. SUBJSRCH_CONFIG_MASTER is the master table for configuring the Subject Search match criteria. Subjects are considered as matched when one or many of these criteria are met. SUBJSRCH_CONFIG_DETAIL is the detail definition of SUBJSRCH_CONFIG_MASTER. It contains one or many records of each SUBJSRCH_CONFIG_MASTER record. A SUBJSRCH_CONFIG_MASTER match criterion is considered as met when all of its associated SUBJSRCH_CONFIG_DETAIL match criteria are met.

For details about SUBJSRCH_CONFIG_MASTER and SUBJSRCH_CONFIG_DETAIL, see the *SAS Enterprise Case Management: Data Dictionary*.

The expressions defined in FROM_PARTY_FIELD_EXP and TO_PARTY_FIELD_EXP must be valid SAS expressions. If you have very complicated logic, you might consider using PROC FCMP to create user-defined functions.

By default, there is one record in SUBJSRCH_CONFIG_MASTER and two associated records in SUBJSRCH_CONFIG_DETAIL. The definition is for matching subjects when both the NATIONAL_ID_TYPE_CD and NATIONAL_ID fields match. Additional definitions can be found in *!SASROOT*`\casemgmtmva\sasmisc\sample\config\load_post_install_data.sas` for Windows platforms or *!SASROOT*`/misc/casemgmtmva/sample/config/load_post_install_data.sas` for UNIX platforms. Eight matching criteria are defined in the sample.

Subject Search Configuration ID	Matched fields	Description
P_ADDRESS	PARTY.X_PRIMARY_ADDRESS_1_TXT +PARTY.X_PRIMARY_ ADDRESS_2_TXT, PARTY.X_PRIMARY_CITY_NM PARTY.X_PRIMARY_POSTAL_CD	Match by primary address. A SAS expression is used to concatenate two street address lines into one.
PASSPORT_ID	PARTY.X_PASSPORT_ID	Match by passport ID.
DRIVER_ID	PARTY.X_DRIVER_LICENSE_ID	Match by driver license ID.
HOME_PHONE	PARTY.X_HOME_PHONE_NO	Match by home phone number.
WORK_PHONE	PARTY.X_WORK_PHONE_NO	Match by work phone number.
CELL_PHONE	PARTY.X_CELL_PHONE_NO	Match by cell phone number.
EMAIL	X_PARTY_EMAIL.X_PARTY_EMAIL	Match by any e-mail addresses that party has.
L_NM_B_DT	PARTY.X_LAST_NM PARTY_X_BIRTH_DT	Match by party last name and birth date.

Subject Search Logic

The driver program %ECM_SUBJSRCH_DRIVER obtains the matches of subject search. Below is the summary of the logic behind it.

1. Include only the active match criteria in SUBJSRCH_CONFIG_MASTER.

2. Transform the party data in step 2 into rectangle structure to include all core and UDF fields into one record.
3. Create views of the party data with derived fields defined in SUBJSRCH_CONFIG_DETAIL.
4. For each active match criteria defined in SUBJSRCH_CONFIG_MASTER, find the parties who are related to the input parties by matching all data fields (for example, FROM_PARTY_FIELD_NM and TO_PARTY_FIELD_NM) defined in SUBJSRCH_CONFIG_DETAIL.
5. Return the list of parties and the SUBJSRCH_CONFIG_ID for each input party. If no match is found, return only the SEARCH_RK.
6. If the program aborts for any reason, return keyword ABORT.

Chapter 13

Report Mart

SAS Enterprise Case Management Report Mart 207

SAS Enterprise Case Management Report Mart

To facilitate user-defined columns and user-defined reference tables, much of the data for SAS Enterprise Case Management and configuration data is stored in “tall skinny” data tables. SAS Enterprise Case Management provides a facility to pivot “tall skinny” data tables into “short wide” data tables for easier reporting. The ECM_RPT library stores the normalized pivoted data. It also keeps the data that is specific for generating SAR e-files in batch.

Data tables in the ECM_RPT library can be created (or re-created) by running %ECM_REPORTING_DRIVER with ecm_autoexec.sas under ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source**. The data tables in this library contain only the current revision of each record (no historical revisions). You can run the %ECM_REPORTING_DRIVER macro nightly or whenever you need to run reports against this library.

%ECM_REPORTING_DRIVER is the driver program for six major macro calls. Calls in %ECM_PIVOT_DATATYPE create pivoted data tables for cases, parties, incidents, generic data, reports, and financial items respectively. %ECM_PIVOT_REF creates user-defined reference tables. A PROC COPY call copies all the ECM relationship tables from ECM_DB to ECM_RPT.

Note: In PROC COPY, the table names in the SELECT statement must be uppercase to facilitate the search of SQL Server tables.

Here is a summary of the derived tables in the ECM_RPT library that are generated by %ECM_REPORTING_DRIVER.

Table 13.1 *Derived Tables in the ECM_RPT Library*

Table	Description
CASE_PIVOT	This is a derived case table with user-defined columns added.

Table	Description
PARTY_PIVOT	This is a derived party table with user-defined columns added .
INCIDENT_PIVOT	This is a derived incident table with user-defined columns added.
FINANCIAL_ITEM_PIVOT	This is a derived financial item table with user-defined columns added.
RR_PIVOT	This is a derived regulatory report table with user-defined columns added.
CASE_X_PARTY	This is a direct copy of ECM_DB.CASE_X_PARTY.
CASE_X_USER_GROUP	This is a direct copy of ECM_DB.CASE_X_USER_GROUP.
INCIDENT_X_PARTY	This is a direct copy of ECM_DB.INCIDENT_X_PARTY.
INCIDENT_X_USER_GROUP	This is a direct copy of ECM_DB.INCIDENT_X_USER_GROUP.
PARTY_X_USER_GROUP	This is a direct copy of ECM_DB.PARTY_X_USER_GROUP.
<C/I/P/G/F/R>_X_	These are derived tables for all user-defined columns that can have more than one value selected or specified. • C = case • I = incident • P = party • G = generic data • F = financial item • R = reports For example, e-mail addresses of subjects will be P_X_PARTY_EMAIL.

Table	Description
X_RT_	These are derived tables for all user-defined reference tables, such as X_RT_ID_TYPE. Derived tables for all in-product reference tables include the following: • RT_CASE_CATEGORY • RT_CASE_STATUS • RT_CASE_TYPE • RT_PARTY_CATEGORY • RT_PARTY_TYPE • RT_INCIDENT_CATEGORY • RT_INCIDENT_TYPE • RT_SOURCE_SYSTEM

SAS has a limit of 32,767 characters for column width. Since the LNGVARCHAR UDF field can be longer than 32,767 characters, the pivot macro reads the <*data_object*>_UDF_LGCHAR_VIEW instead of <*data_object*>_UDF_LGCHAR_VALUE to get the field values. In the view, the LNGVARCHAR UDF field is broken into two 32,760–character fields to accommodate UTF-8 codes. The field value columns are called UDF_VALUE_1 and UDF_VALUE_2. In the resulting table, the field names are suffixed with '_1' and '_2'. For example, in the sample SAR UDF definition, there is an LNGVARCHAR field called X_ACTIVITY_DESC_LONG_TXT. It is broken into X_ACTIVITY_DESC_LONG_TXT_1 and X_ACTIVITY_DESC_LONG_TXT_2 in the ECM_RPT.RR_PIVOT table.

Chapter 14

Internationalization

Overview 211
Specify the Database Character Encoding 211
SAS Session Encoding Consideration and DBCS Support 212
Default Encoding for Databases Supported by SAS Enterprise Case Management 213
Restricting the Maximum Length of VARCHAR Fields 213
Naming Conventions for Locales 214
Create and Use Custom Translated Messages 214
Localizing Custom Table Labels and Column Labels 215
Localizing Reference Tables 216
Localizing Workflow Activities and Statuses 216

Overview

The default language used in SAS Enterprise Case Management is English. This chapter discusses the processes related to configuring SAS Enterprise Case Management for use with other languages.

Specify the Database Character Encoding

This section applies only if you are using SAS Enterprise Case Management with an Oracle or PostgreSQL database. If you are using a Microsoft SQL Server database, the database will automatically use a UTF-16 encoding for all string data.

Before you install SAS Enterprise Case Management, you must decide which database character encoding to use for your environment. Determining an appropriate encoding to use for your SAS Enterprise Case Management database is dependent upon the following:

- the languages that the application needs to support now.
- the languages that the application needs to support in the future.

- consideration of the performance implications associated with choosing a database character set. For example, a single-byte character set provides better performance when compared with multi-byte character sets. Single-byte character sets also tend to take up less space in your environment. However, they offer only restricted multilingual support.

The character set that you choose affects what type of encoding scheme is used. For example:

Scenario 1
: If you need to support English, French, and Portuguese languages, then single-byte, 8-bit encoding schemes are appropriate because they define up to 256 characters and can often support a group of related languages. One example is the ISO 8859-1 character set, which supports many Western European languages. For Oracle databases, you could also use the WE8ISO8859P1 character set. When you use a character set that supports a group of languages, your database has restricted multilingual support.

Scenario 2
: If you need to support double-byte character languages (for example, Japanese, Chinese, or Korean), then you can use legacy ANSI-based double-byte character set (DBCS) encodings such as shift-jis, gbk, krc, or big5. By using these encodings, you can use one DBCS language and English (for example, Japanese and English).

Scenario 3
: If you need to accommodate data for multiple DBCS languages (for example, Japanese and Korean), DBCS languages with European languages (for example, Chinese with French), or Western and Central European languages (for example, German and Polish), then you must use Unicode (UTF-8) encoding. If you are using an Oracle database, then you can also use the AL32UTF8 character set, which is based on the Unicode UTF-8 character set.

SAS Session Encoding Consideration and DBCS Support

If your database encoding supports multiple DBCS languages, then you must also use UTF-8 for the SAS session encoding. For example, if you use SAS to produce reports that contain data for multiple languages, then you must specify UTF-8 for the SAS session encoding. If you need to support one double-byte language and English, then you must also use a DBCS encoding for the SAS session encoding.

Although DBCS encoding is supported in SAS, SAS tables are still ASCII based. That means any column defined in SAS tables needs to be doubled in length. For example, if X_BRANCH_ADDRESS_TXT is defined as follows in ECM_DB.CASE_UDF_DEF, the field length of X_BRANCH_ADDRESS_TXT in the SAS table should be 200.

```
 UDF_TABLE_NM='CASE'
UDF_NM= 'X_BRANCH_ADDRESS_TXT'
UDF_TYPE_NM= 'VARCHAR'
UDF_DESC= 'Branch address where activity occurred'
MAX_CHAR_CNT= 100
```

This requirement affects the creation of the SAS Enterprise Case Management report mart and stored processes that process data in SAS tables. SAS Enterprise Case Management does not handle this situation properly. As a result, data can be truncated in the ECM_RPT tables and in the Case Network Analysis node detail UI. To work around

this limitation, a proper length multiplier has to be set. The global macro variable ecm_charMultiplier is defined for this purpose. Refer to the comment in the ecm_global_mvar macro to set its correct value.

Default Encoding for Databases Supported by SAS Enterprise Case Management

The *!SASROOT*`\casemgmtmva\sasmisc\sample\dbscript` directory, for Windows platforms, contains database-specific subdirectories that include scripts for the following databases:

Note: For UNIX platforms, the directory is *!SASROOT*`/misc/casemgmtmva/sample/dbscript`.

- Oracle
- PostgreSQL
- SQL Server

Within these database-specific directories, database scripts are provided that enable you to create and initialize your SAS Enterprise Case Management database. The Oracle and SQL Server scripts create the schema instead of the database, so an encoding is not specified in these scripts. By default, the PostgreSQL scripts use UTF-8 encoding. If you are using PostgreSQL on Windows for double-byte character languages, such as Japanese, Chinese, or Korean, then you might need to update the character set encoding value that is used in the PrepareDatabase script. For example, to specify a character set encoding to use the extended UNIX code for the Korean language, you can customize the PostgreSQL scripts as follows:

1. Open the PrepareDatabase script in a text editor. This file is located in one of the following directories, depending on your platform:
   - Windows: *!SASROOT*`\casemgmtmva\sasmisc\sample\dbscript\PostgreSQL`
   - UNIX: *!SASROOT*`/misc/casemgmtmva/sample/dbscript/PostgreSQL`
2. Change the -E UNICODE option to -E EUC_KR.

Restricting the Maximum Length of VARCHAR Fields

If you are using a multi-byte character set encoding, it is recommended that you restrict the maximum length of any VARCHAR fields in your Custom Page Builder UI definition files to 1000 characters. The recommended maximum field length that you should set for the VARCHAR fields property is as follows:

Database	Maximum Field Length for VARCHAR fields property
Oracle	If your Oracle database is using AL32UTF8 encoding, then the CHAR data type can hold up to 4 bytes. The SAS Enterprise Case Management tables use the VARCHAR2 data type. SAS Enterprise Case Management specifies the number of characters, and then Oracle handles how to translate the number of bytes. However, Oracle has a maximum limit of 4000 bytes on the VARCHAR data type. Therefore, it is recommended that you restrict VARCHAR field lengths to 1000. If you are using English or other single-byte character sets (for example, WE8ISO8859P1), then you can extend the size to 4000.
PostgreSQL	4000 characters.
SQL Server	1000 characters.

Naming Conventions for Locales

When the translation of an object is loaded, the locale that is associated with that translation must follow the standard Java naming convention for locales. The naming convention for locales requires that the language code must be a lowercase two-letter code from the ISO 639 specification and that the country code must be an uppercase two-letter code from ISO 3166. For example, fr is the language code for French; en is the language code for English. For more information, see the following website: **http://java.sun.com/developer/technicalArticles/J2SE/locale.**

The macro variables LOCALE_SAS and LOCALE_DEF in ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/control/ecm_autoexec.sas** are used to define the SAS session locale and the default locale when text in the SAS session locale is not available. These macro variables are initialized as follows:

```
%let locale_sas=%SYSFUNC(getpxlocale()) ;

  %let locale_default=%substr(&locale_sas,1,2);
```

If you want to set the SAS session locale to German, change

```
'%let locale_sas=%SYSFUNC(getpxlocale()) ; '
```

to

```
' %let locale_sas=de_DE ;'
```

Create and Use Custom Translated Messages

All SAS macros and stored processes in SAS Enterprise Case Management make use of the SASMSG function to retrieve translated log messages based on the locale of the server. To localize these strings, you can use the %SMD2DS macro to add messages that can be used by the SASMSG function. For the syntax of the %SMD2SD and SASMSG

functions, see Appendix 3, "SASMSG and %SMD2DS," on page 251. The message file that SAS Enterprise Case Management uses is sashelp.entcm.

To set the locale of the ECM SAS server, change the value of the LOCALE_SAS macro variable to the desired locale in ***SAS-configuration-directory*\Lev<*num*>\Applications\SASCaseManagementServerCfg\6.2\Source\control\ecm_autoexec.sas**.

For example:

```
%let locale_sas=DE; /* for German */
```

Restart the SAS object spawner to put this into effect.

Localizing Custom Table Labels and Column Labels

SAS Enterprise Case Management stores table labels and columns labels in the case management database to guarantee consistent terminology across the web application and SAS code. The data stored in the label translation tables is generated by a batch process that reads the application's standard Java resource bundles and the custom resource bundles, and then stores the appropriate data in the label translation tables.

Labels for table and column names are determined by a translation key naming convention. In most cases, labels can be defined for a standard or custom table or column by naming the translation key table.tableNm.label.txt or field.tableNm.columnNm.label.txt, respectively. For example, table.x_party_alias.label.txt provides the label for the X_PARTY_ALIAS user-defined table and field.x_party_alias.x_alias_nm.label.txt provides the label for the X_PARTY_ALIAS_NM column in that table.

There is also support for differentiating the labels based on the entity type. For example, if a situation arises where tables and columns with the same name need to have different labels or translations for cases and incidents, then the naming convention is extendable to include the entity name in the key name. For example, if both case and incident tables define a custom table named TAGS with a field named TAG_NM, they can have different table labels by defining table.case.tags.label.txt = Case Tags and table.incident.tags.label.txt = Incident Tags.

Anytime a custom resource bundle is uploaded with a new translation for one of the table or column labels, the values in the label translation tables need to be recomputed. To recompute those tables, an application administrator should perform the following steps:

1. Log on to SAS Enterprise Case Management.
2. Click the **Administration** tab.
3. Click **Refresh Report Mart Labels**.

For more information about custom resource bundles, see "Custom Resource Bundles" on page 77.

Localizing Reference Tables

REF_TABLE_VALUE is the main table for defining code description in the default locale. The following tables are for supporting multiple languages.

REF_TABLE_TRANS
holds the translations for all the codes in REF_TABLE_TRANS.

ECM_LOCALE
holds the list of supported locales along with instructions for which locale to use if there is no translation for that locale.

FULL_REF_TABLE_TRANS is a view that corresponds to a cross product of the REF_TABLE_VALUE table and the ECM_LOCALE table. Each row would hold the proper translation for that supported locale. The following is an example of a SAS program to set the priority look-up table in German:

```
proc sql;

   insert into ecm_db.ecm_locale values ('de','def');

   delete from ecm_db.ref_table_trans
   where ref_table_nm='X_RT_PRIORITY' and locale='de';


   insert into ecm_db.ref_table_trans values ('X_RT_PRIORITY', 'H', 'de', ' hoch');
   insert into ecm_db.ref_table_trans values ('X_RT_PRIORITY', 'M', 'de', ' mittler');
   insert into ecm_db.ref_table_trans values ('X_RT_PRIORITY', 'L', 'de',’ niedrig’);
```

Localizing Workflow Activities and Statuses

In SAS Enterprise Case Management, the names of workflow activities and statuses can be localized by performing the following steps:

1. From the New Data Object dialog box, or the Edit dialog box of an existing object, click ... next to either the **Data Object Label** or **Description** field.
2. Specify the resource bundle key in the **Localization Key** field, and click **OK**.
3. Include translations for that key in the appropriate custom properties files and upload the modified custom properties file to the server (for example, ecm.sample.workflow.status.open.txt = Open).

If a key is specified in the workflow, but no translation is found, the name of the activity or status will be used.

Chapter 15

Adding Custom SAS Code

Adding Custom SAS Code 217

Adding Custom SAS Code

An existing SAS macro can be overridden by adding a SAS macro program in ***SAS-configuration-directory*/Lev<*num*>/Applications/SASEnterpriseCaseManagement/6.2/Source/ucmacros** with the same filename. You can also add a new macro and save it in the same location. Here are some tips for writing your own code.

1. The following statement should be added to the beginning of the program if it is not being called by any calling program.

   ```
   %inc 'SAS-configuration-directory/Lev<num>/Applications/SASEnterpriseCaseManagement/
   6.2/Source/control/ecm_autoexec.sas';
   %ecm_db_connect;
   ```

   To run %ECM_DB_CONNECT, you have to be a valid SAS Enterprise Case Management user. If you are the programmer testing the program in SAS Display Manager, make sure you log on to the SAS server machine with a valid user ID.

2. The %ECM_PIVOT_DATATYPE_SUBSET macro can be used to convert the ECM data from its native format into rectangular structure. For example, if you want to retrieve case records with all the core and UDF fields for CASE_TYPE='FIN' , you can use the following code:

   ```
   data case_subset;
   set ecm_db.case_live (keep=case_rk);
   where case_type='FIN';
   run;
   %ecm_create_label_fmt;
   %ecm_pivot_datatype(in_lib=ecm_db,out_lib=work,datatype=CASE,
          subset_dsn=case_subset,include_lgchr_YN=N,table_wh="CASE");
   ```

   The output tables are WORK.CASE_PIVOT for the case table and WORK.C_X_<UDF_TABLE_NM> for the case subtables. <UDF_TABLE_NM> is the value of the UDF_TABLE_NM column defined in CASE_UDF_DEF. These tables contain only the most current data. Therefore, the value of the DATA_OBJECT_RK column without VALID_FROM_DTTM can be used to join the tables.

3. To place a generic data table in rectangular structure, use the following code:

```
%let table_nm=X_BRANCH ;  /* specific generic table name */
%ecm_pivot_datatype(datatype=GENERIC_DATA,in_lib=ECM_DB,
out_lib=WORK, table_wh="&table_nm");
```

The output table is WORK.G_X_BRANCH.

Chapter 16

Event Logging

Overview 219
Currently Supported Events 219
Creating a Batch Load Event 222

Overview

SAS Enterprise Case Management enables you to log events for entities. The event logs provide a history of activities performed for audit purposes. All events include a timestamp that indicates when the event happened and who performed the event unless otherwise noted. This chapter describes the events that are logged for entities.

Currently Supported Events

All events include a timestamp that indicates when the event happened and the user who performed the action that created the event unless otherwise noted.

Event Type	Event Description
Save entity event	A save entity event is logged when an entity is saved. The version number of the saved record is stored in the description column.
Load entity event	A load entity event is logged when an entity is loaded into the system from an ETL (extract, transform and load) process or web service call. For information on saving load events in an ETL process, see "Creating a Batch Load Event" on page 222 .
Lock entity event	A lock entity event is logged when a user locks an entity. The user who locked the entity is saved in the description column.

Event Type	Event Description
Unlock entity event	An unlock entity event is logged when a user unlocks an entity. The user who unlocked the entity is saved in the description column.
Status change event	A status change event is logged when a user changes the status of a workflow activity from the Edit Case page. The original status and the new status are stored in the description column.
New status event	A new status event is logged when the status of a case has changed as a result of a change within the associated workflow. The Created By column is blank because these events are triggered from the associated workflow and might not necessarily be caused by a user action (for example, a status change could result from an expired timer). If the new status event immediately follows a status change event, you can deduce that the new status was caused by the status change user action.
Assign case owner event	An assign case owner event is logged when a case is assigned to a new owner. The user who now owns the case is saved in the description column.
Assign incident owner event	An assign incident owner event is logged when an incident is assigned to a new owner. The user who now owns the incident is saved in the description column.
Assign report owner event	An assign report owner event is logged when a report is assigned to a new owner. The user who now owns the report is saved in the description column.
Add comment event	An add comment event is logged when a comment is added to an entity. The comment subject is saved in the description column.
Edit comment event	An edit comment event is logged when an existing comment has been edited. The comment subject is saved in the description column.
Delete comment event	A delete comment event is logged when a comment is deleted from an entity. The comment subject is saved in the description column.
Add attachment event	An add attachment event is logged when an attachment is added to an entity. The attachment file name is saved in the description column.
Delete attachment event	A delete attachment event is logged when an attachment is deleted from an entity. The attachment file name is saved in the description column.

Event Type	Event Description
Add incident event	An add incident event is logged when an incident is added to a case. The key for the incident is saved with the event, and the ID and source system code of the incident is shown in the description.
Delete incident event	A delete incident event is logged when an incident is removed from a case. The key for the incident is saved with the event, and the ID and source system code of the incident is shown in the description.
Add financial item event	An add financial item event is logged when a new financial item is added to an incident or a case. The key for the financial item is saved with the event, and the ID and the source system code of the financial item is shown in the description.
Edit financial item event	An edit financial item event is logged when a financial item is modified. The key for the financial item is saved with the event, and the ID and the source system code of the financial item is shown in the description.
Delete financial item event	A delete financial item event is logged when a financial item is removed from an incident or a case. The key for the financial item is saved with the event, and the ID and the source system code of the financial item is shown in the description.
Add party relationship event	An add party relationship event is logged when a party is linked to an incident, case, or party with a given relationship. The key for the party is saved with the event, and the ID and source system code of the party is shown in the description.
Remove party relationship event	A remove party relationship event is logged when a party relationship is unlinked from an incident, case, or party. The key for the party is saved with the event, and the ID and source system code of the party is shown in the description.
Add associated case event	An add associated case event is logged when one or more cases are associated with a case. The key for the associated case is saved with the event, and the ID and source system code of the case is shown in the description.
Remove associated cases event	A remove associated case event is logged when one or more associated cases are removed from a case. The key for the associated case is saved with the event, and the ID and source system code of the case is shown in the description.
Add identical party event	An add identical party event is logged when one or more parties are identified as identical parties. The key for the identical party is saved with the event, and the ID and source system code of the party is shown in the description.

Event Type	Event Description
Remove identical party event	A remove identical party event is logged when one or more parties are removed from an identical parties list. The key for the identical party is saved with the event, and the ID and source system code of the party is shown in the description.
Submit regulatory report event	A submit regulatory report event is logged when a regulatory report is submitted from the Case Detail page by clicking the **Submit Report** button.
Add report event	An add report event is logged when a report is added to a case or incident. The key for the report is saved with the event, and the ID and source system code is shown in the description.
Activate case workflow event	An activate case workflow event is logged when a workflow is activated for the case.
Terminate case workflow event	A terminate case workflow event is logged when workflow is terminated from the case.
Activate report workflow event	An activate report workflow event is logged when a workflow is activated for the report.
Terminate report workflow event	A terminate report workflow event is logged when workflow is terminated from the report.

Creating a Batch Load Event

Load events are logged when an entity is loaded into the system from an extract, transform and load (ETL) process or web service call. ETL processes must manually add this event when loading an entity. The following SAS program provides an example of how to insert an ETL load event for a case.

```
/********** SETUP LIBNAME **********/
%let DB_SERVICE = …;
%let DB_SCHEMA = …;
%let DB_USER = …;
%let DB_PASSWORD = …;
libname ecm_db oracle
path="&DB_SERVICE" user="&DB_USER" password="&DB_PASSWORD" schema="&DB_SCHEMA";
/********** GET NEXT CASE KEY AND NEXT CASE EVENT KEY **********/
proc sql noprint;
connect to oracle (
path="&DB_SERVICE" user="&DB_USER" password="&DB_PASSWORD" connection=global);
select * into :CASE_KEY from connection to oracle
(select &DB_SCHEMA..case_rk_seq.nextval from dual);
select * into :CASE_EVENT_KEY from connection to oracle
(select &DB_SCHEMA..event_rk_seq.nextval from dual);
```

```
disconnect from oracle;
quit;
%let CASE_KEY = %trim(&CASE_KEY);
%let CASE_EVENT_KEY = %trim(&CASE_EVENT_KEY);
/********** GET CURRENT DATE/TIME **********/
%let CURRENT_DATETIME = %sysfunc(datetime(), datetime);
%let CURRENT_DATETIME_SQL = "&CURRENT_DATETIME"dt;
/********** INSERT CASE **********/
…
/********** COPY TO CASE_VERSION TABLE **********/
…
/********** INSERT USER DEFINED FIELD VALUES **********/
…
/********** INSERT GROUP PERMISSIONS **********/
…
/********** INSERT ETL CASE EVENT **********/
proc sql noprint;
insert into ecm_db.case_event values (
&CASE_EVENT_KEY,
&CASE_KEY,
'LOADEN',
'event.etl.load.txt',
null,
```

In the preceding example, you use sequences to get the next record key (CASE_RK_SEQ, INCIDENT_RK_SEQ, PARTY_RK_SEQ, RR_RK_SEQ, or EFILE_RK_SEQ) and event key (EVENT_RK_SEQ). LOADEN is the event-type code for load events. event.etl.load.txt is the resource bundle property key defined in AppResources.properties for the ETL load event description. Null is the user ID (you can load an actual user ID instead of null). The final value in the insert statement is the timestamp indicating when the event took place. For web service loads, the load event is automatically created if the source system of the loaded record is not SASECM.

Chapter 17

Additional Tasks

Case Routing Configurations for SAS Enterprise Case Management: Regional Manager Setup 225
Add the Region Case User-Defined Field 225
Create the Region User-Defined Reference Table 225
Create a Group in SAS Management Console for Each Region 226
Create the Regional Manager Group Case User-Defined Field 226
Add the Region User-Defined Field to the User Interface Definition 227
Derive the Regional Manager Group Name from Region 227
Use the Regional Manager Group Field in the Workflow 227
Add a Root-Level Set Process Participant Policy to the Workflow 228
Upload the User Interface Definition and Workflow 229
Test Your New Configuration 229

Setting Up Data Management Jobs 230

SAS Enterprise Case Management – Backup Requirements 230

Metadata Clustering 231

Case Routing Configurations for SAS Enterprise Case Management: Regional Manager Setup

This chapter describes how to configure SAS Enterprise Case Management to support routing the review capability of cases to regional managers, based on region.

Add the Region Case User-Defined Field

This field is used to store the region code.

```
insert into ecm_db.case_udf_def values (
   'CASE', 'X_REGION_CD', 'VARCHAR', 'Region code', 3);
```

Create the Region User-Defined Reference Table

This reference table contains all possible regions.

```
insert into ecm_db.ref_table_value values (
   'X_RT_REGION', 'N', 'North', null, null, 0);
insert into ecm_db.ref_table_value values (
   'X_RT_REGION', 'S', 'South', null, null, 0);
```

```
insert into ecm_db.ref_table_value values (
   'X_RT_REGION', 'E', 'East', null, null, 0);
insert into ecm_db.ref_table_value values (
   'X_RT_REGION', 'W', 'West', null, null, 0);
```

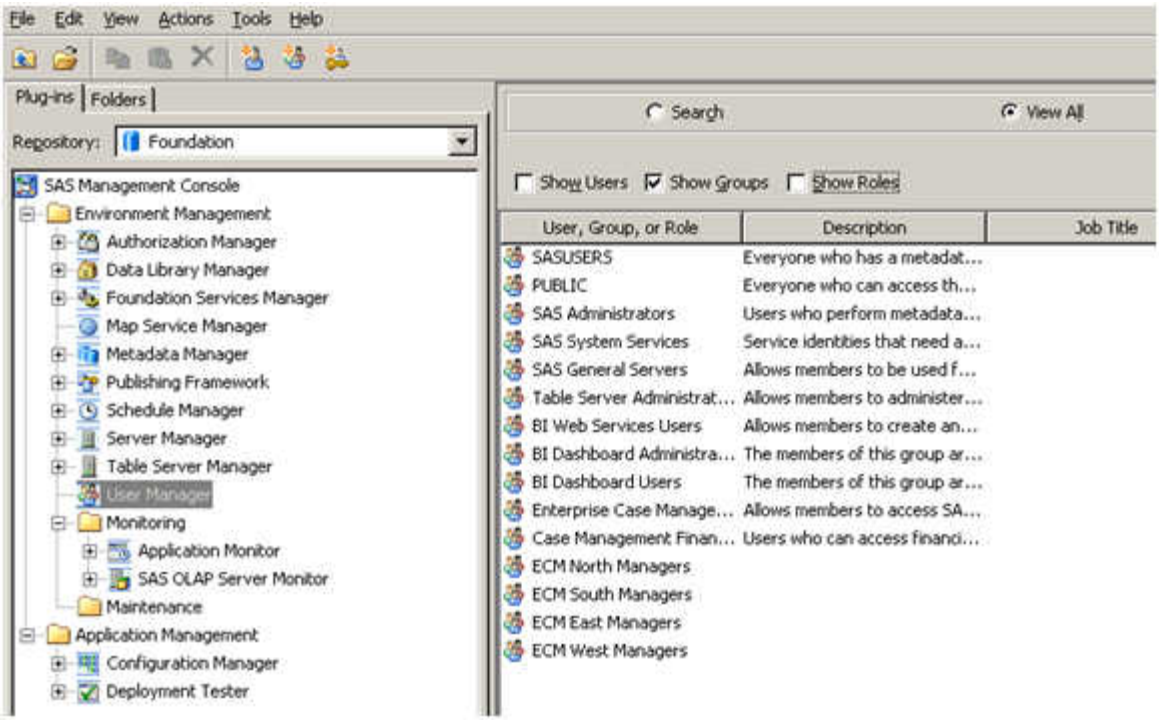
Create a Group in SAS Management Console for Each Region

Each group contains the managers assigned to that region. Here is a list of those managers:

- SAS Enterprise Case Management North managers
- SAS Enterprise Case Management South managers
- SAS Enterprise Case Management East managers
- SAS Enterprise Case Management West managers

In the following display, the previously listed groups were created to correspond to each region.

Display 17.1 *Create a Group – SAS Management Console*



Create the Regional Manager Group Case User-Defined Field

This field is used to store the regional manager group name assigned to review the case. This field is derived from the region user-defined field.

```
insert into ecm_db.case_udf_def values (
   'CASE', 'X_MANAGER_GROUP_NM', 'VARCHAR', 'Manager group name', 60);
```

Add the Region User-Defined Field to the User Interface Definition

You can now prompt for the region on the Case Detail page by adding the following code to the case user interface definition.

```
<field name="CASE.X_REGION_CD" type="dropdown" required="true"
       values="GetLabelValues('X_RT_REGION')">
   <label>Region:</label>
</field>
```

Derive the Regional Manager Group Name from Region

The case user interface definition should be updated to derive the regional manager group name from region in the finalize section of the Case Detail page as follows:

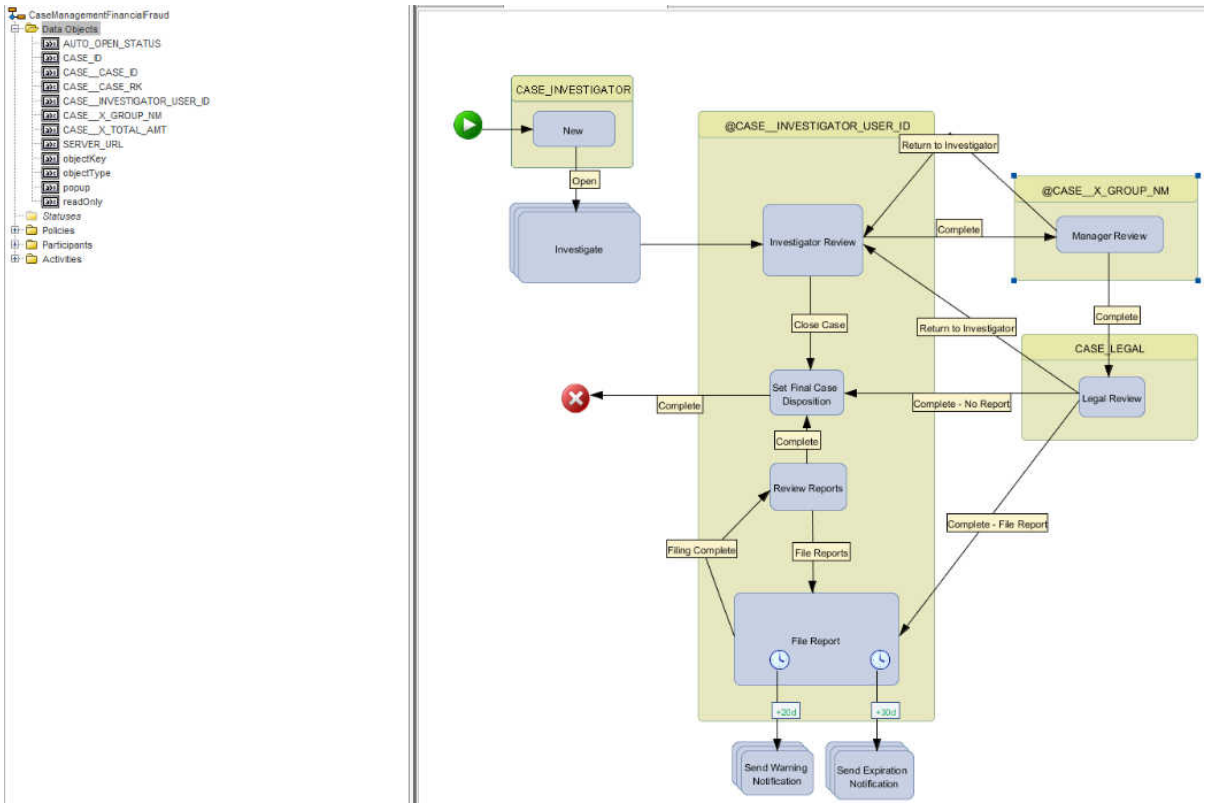
```
<finalize>
   <set name="CASE.X_MANAGER_GROUP_NM"
        value="if(CASE.X_REGION_CD = 'N', 'ECM North Managers',
                  CASE.X_MANAGER_GROUP_NM)"/>
   <set name="CASE.X_MANAGER_GROUP_NM"
        value="if(CASE.X_REGION_CD = 'S', 'ECM South Managers',
                  CASE.X_MANAGER_GROUP_NM)"/>
   <set name="CASE.X_MANAGER_GROUP_NM"
        value="if(CASE.X_REGION_CD = 'E', 'ECM East Managers',
                  CASE.X_MANAGER_GROUP_NM)"/>
   <set name="CASE.X_MANAGER_GROUP_NM"
        value="if(CASE.X_REGION_CD = 'W', 'ECM West Managers',
                  CASE.X_MANAGER_GROUP_NM)"/>
</finalize>
```

Whenever the case is saved, the regional manager group user-defined field is derived from the region.

Use the Regional Manager Group Field in the Workflow

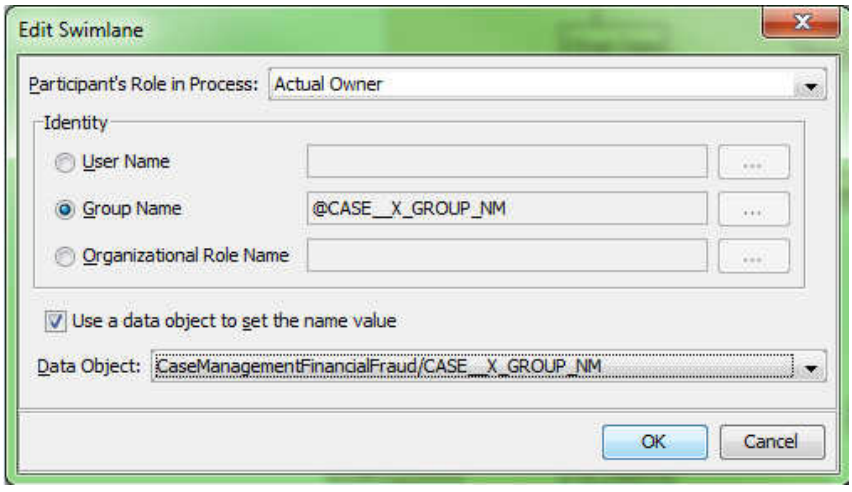
Add a data object for the regional manager group user-defined field (CASE__X_MANAGER_GROUP_NM) in SAS Workflow Studio. Use the value of this field as the actor (also known as swimlane) for the Manager Review activity. This allows the value of this field to determine which group can perform the Manager Review activity. The following display shows a workflow diagram in SAS Workflow Studio.

Display 17.2 *SAS Workflow Studio*



In addition, the **Use a data object to set the name value** check box must be selected on the Edit Swimlane dialog box. The following figure shows the Edit Swimlane dialog box in SAS Workflow Studio.

Display 17.3 *Edit Swimlane Dialog Box*

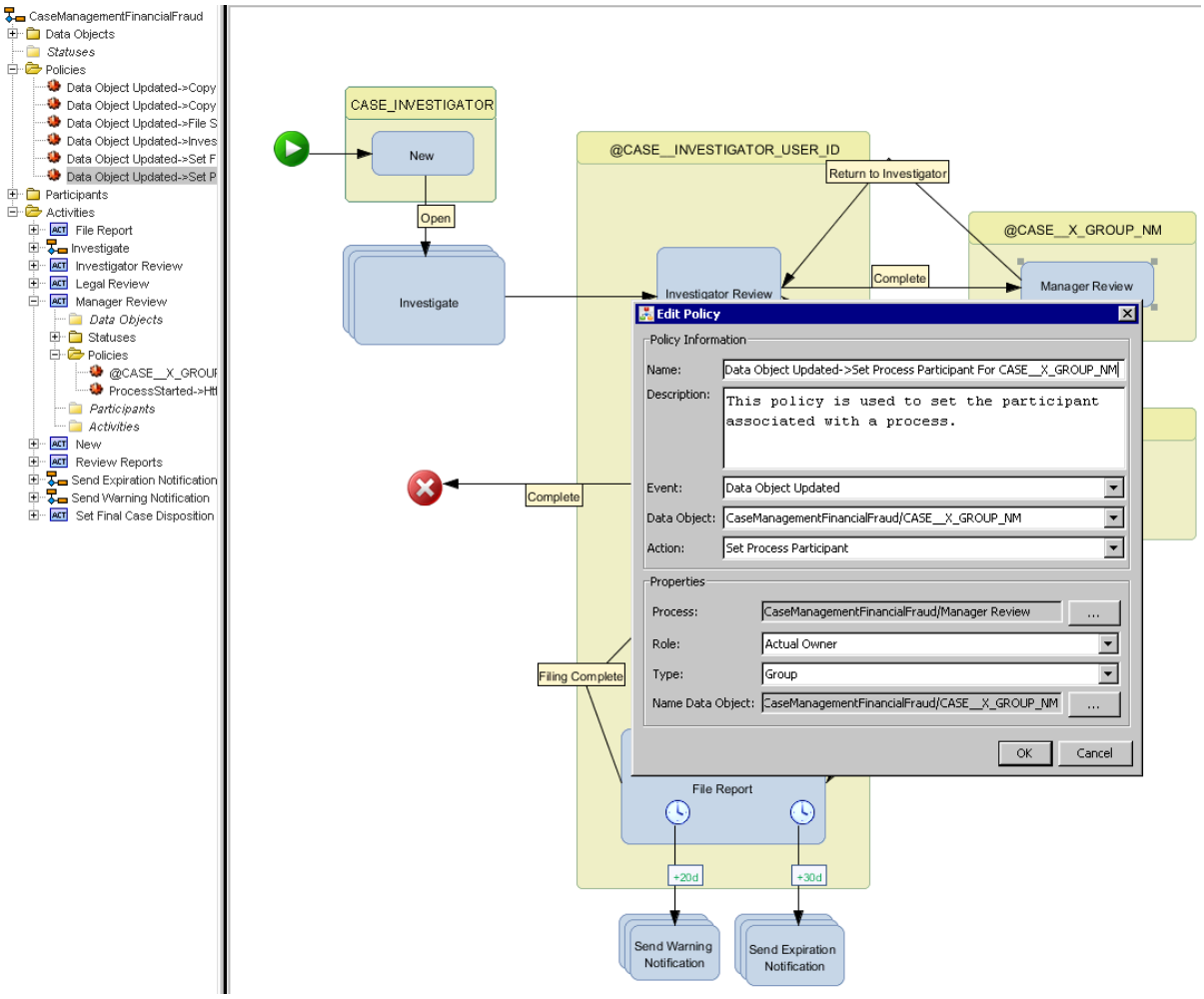


Add a Root-Level Set Process Participant Policy to the Workflow

The Set Process Participant policy allows the access control entries to be updated for the Manager Review activity whenever the CASE__X_MANAGER_GROUP_NM data

object is updated. This enables the Manager Review activity to be reassigned to a different regional manager group if the region is changed after the activity has started. The following display shows the Edit Policy dialog box in SAS Workflow Studio.

Display 17.4 *Edit Policy – SAS Workflow Studio*



Upload the User Interface Definition and Workflow

Upload the user interface definition from the SAS Enterprise Case Management **Administration** tab. Upload the workflow template from SAS Workflow Studio.

Test Your New Configuration

1. Create a new case, setting the region user-defined field.
2. Move through the workflow until you reach the Manager Review activity.
3. Verify that only the managers in the corresponding region can perform the activity.
4. Log on as someone who can edit the case.
5. Change the region to another region.
6. Verify that managers from the new region can perform the activity.

Setting Up Data Management Jobs

It is recommended that programs for refreshing the SAS Enterprise Case Management report mart should be set up to run regularly as batch jobs. There are many ways to set up scripts to run SAS programs in batch. This section provides a simple example for Windows.

1. Create a directory where the job script and SAS batch code will be stored. For example:

   ***SAS-configuration-directory*\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\jobs**

2. Create a SAS program to call the SAS Enterprise Case Management macro (for example, ecm_reporting_driver_job.sas). Add the following line to the program.

   ```
   %ecm_reporting_driver;
   ```

3. Create a Windows command file to call SAS (for example, ecm_job_generate_batch_efile.cmd). Use the following statement as a reference and define the content of the command file with your site information.

   ```
   "SAS-installation-directory\SASFoundation\9.4\sas.exe"
   -config "SAS-configuration-directory\
   Lev<num>\SASApp\WorkspaceServer\sasv9.cfg"
   -autoexec "SAS-configuration-directory\Lev<num>\Applications\
   SASEnterpriseCaseManagement\6.2\Source\control\autoexec.sas" -SYSIN
   "SAS-configuration-directory\Lev<num>\Applications\SASEnterpriseCaseManagement\
   6.2\Source\jobs\ecm_reporting_driver_job.sas" -log
   "SAS-configuration-directory\Lev<num>\Applications\SASEnterpriseCaseManagement\
   6.2\Source\jobs\ecm_reporting_driver_job.log" -nodms
   ```

   Note:

   - *SAS-installation-directory* should be the path where SAS is installed.
   - *SAS-configuration-directory* should be the path where SAS Enterprise Case Management is configured.
   - Make sure that there are no line breaks in the command program.

SAS Enterprise Case Management – Backup Requirements

To ensure the integrity of the SAS Enterprise Case Management system, you should establish a formal, regularly scheduled backup process. It is important to back up all of the following items at the same time so that related information will be synchronized if a restore becomes necessary:

SAS Metadata
: SAS Metadata contains ECM server and middle-tier configuration information, user or group capabilities, and more. The instructions for backing up all SAS metadata can be found in the topic “Backing up and Restoring Your System” in the *SAS 9.4 Intelligence Platform: System Administration Guide*.

SAS Content Server
: All UI definition files, custom properties files, and attachments to cases or incidents are stored in the SAS Content Server in ***SAS-configuration-directory/*** **Lev<*num*>/AppData/SASContentServer**. Instructions for backing up the SAS Content Server can be found at the following location:

http://support.sas.com/documentation/cdl/en/bisag/60945/HTML/default/a003133703.htm#a003266477

SAS Shared Service Database
: SAS Enterprise Case Management uses SAS Web Infrastructure Platform to manage workflows, attachments, alerts, and more. The database associated with SAS Web Infrastructure Platform should be backed up regularly.

SAS Enterprise Case Management Database
: The SAS Enterprise Case Management database contains all cases, incidents, subjects records, reports, e-files, reference tables, and configuration data of various SAS Enterprise Case Management components. This database should be backed up regularly.

SAS Social Network Analysis Database
: SAS Enterprise Case Management uses SAS Social Network Analysis to generate and annotate case network analysis graphs. The database associated with SAS Social Network Analysis should be backed up regularly.

SAS Enterprise Case Management Configuration Directory
: ***SAS-configuration-directory*****\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2** contains SAR e-file data and any custom code defined at your site. The complete directory should be backed up regularly.

Metadata Clustering

Metadata server clusters provide high availability and performance scaling options. Information on how to set up your environment to support metadata server clustering can be found in "Configuring a Metadata Server Cluster" in the *SAS 9.4 Intelligence Platform: System Administration Guide*.

To take advantage of the metadata server clustering in the SAS Enterprise Case Management SAS server environment, some adjustment to the SAS programs must be made. In ecm_autoexec.sas, found in ***SAS-configuration-directory*****\Lev<*num*>\Applications\SASEnterpriseCaseManagement\6.2\Source\control**, the following SAS options statement is included to allow easy access to the metadata server.

```
options metaserver="<metadata server>"  metaport=<metadata port number>
metarepository="<metadata repository name>";
```

In the SAS metadata server cluster environment, **<metadata server>** is the first node of the cluster. The existence of this statement prevents SAS Enterprise Case Management from connecting to the other nodes when the first node is not available. Therefore, it defeats the purpose of clustering. Because of this, the statement should be commented out. Removing this statement allows SAS Enterprise Case Management stored processes to connect to any available metadata server node. However, it affects the experience of SAS Display Manager users and SAS batch processes because SAS prompts for metadata connection information. To provide the metadata connection information without specifying a particular node name, the **-metaprofile=** option can

be added to the command for starting up SAS. The location of the profile is in ***SAS-configuration-directory*\Lev<*num*>\metadataConfig.xml**. Refer to SAS 9.4 documentation for more information about SAS system options for metadata.

Appendix 1

SAS Enterprise Case Management Web Service

Introduction 233
A Sample Request 234

Introduction

SAS Enterprise Case Management provides a web service to allow external systems to load data into the system. The web service accepts these types of requests:

Configuration requests
: Retrieve the list of type, category, and subcategory configurations for one or more entity types. The results can be filtered by which types of entities the caller can create. A configuration request can also return the list of user-defined fields and the localized label for each field.

Search requests
: Search for subjects in the system based on a configurable set of criteria and return any matching subjects along with the list of matching criteria.

Lookup requests
: Retrieve one or more objects in the system by ID and source system code and return all the standard and user-defined fields associated with the objects that are found. The request can also return a list of all the entities related to the one returned.

Create requests
: Create new entities and link them together. Existing objects can be updated as well. This request can load live or historic data. A create request can be run in test mode, which processes the request — running all data integrity checks — but does not save the results in the database.

Note: Create and update requests do not validate data against a UI definition. It is the responsibility of the calling process to validate the data before submitting the request.

The web service uses the Simple Object Access Protocol (SOAP). Authentication is required and all requests are processed with the visibility and capabilities of the authenticated user.

A Sample Request

The following code shows a simple example of how to call the web service from within SAS using PROC SOAP.

```
filename infref "!SASROOT\casemgmtmva\sasmisc\sample\webservice\ping-request.xml";
filename outfref "c:\temp\ping-response.xml";

%ecm_ws_get_url;

proc soap IN=infref
          OUT=outfref
          SRSURL="&ecm_ws_srsurl"
          URL="&ecm_ws_url"
          wssusername="THE USERNAME"
          wsspassword="THE PASSWORD"
SOAPACTION="http://sas.com/solutions/casemgmt/webservice/
            CaseManagementServiceInterface/create";
run;
```

PROC SOAP automatically handles the authentication. The SRSURL directs PROC SOAP to the service registry, which it uses to find the SAS Security Token Service. PROC SOAP passes the WSSUSERNAME and WSSPASSWORD to the Security Token Service to authenticate the user. PROC SOAP submits the request to the web service only after authentication succeeds. If you prefer not to hardcode the password in the PROC SOAP call, you can remove the WSSUSERNAME and WSPASSWORD parameters. In that case, PROC SOAP authenticates the user login with the SAS metadata server. Then, it generates a one time password to call the web service. Since SAS Enterprise Case Management web services checks for user permission and capabilities, it is important that the user has the right permission to perform the web service actions.

Additional sample files can be found in *!SASROOT***\casemgmtmva\sasmisc\sample\webservice** for Windows platforms or *!SASROOT***/misc/casemgmtmva/sample/webservice** for UNIX platforms. The WSDL is available from the SAS Enterprise Case Management web application found at **http://<host>:<port>/SASEntCaseManagement/service/CaseManagementService.wsdl**.

Appendix 2

Troubleshooting

Workflow Status Updates 236

Report Workflow Transition Problems 236

Unable to Open SAS Workflow Studio 236

Database Error Warnings and SAS Deployment Wizard 237

Post-installation Database Steps Required after Unsuccessful SAS Deployment Wizard Database Installation 237

ODBC Database Transcode Error 238

Case Network Graph Stops Working 238

Case Network Analysis Web Service Not Created 239

Special Characters Are Missing in Case Network Analysis and Report Mart . . . 240

Specifying the Version Number for SAS Enterprise Case Management 241

DBMS JAR File and Multiple Machine Installations 241

Assigning the Primary Owner to a Case 241

Adding the Custom Column Type VARCHAR 241

Locking and Unlocking an Entity 242

Using the Search Functionality in SAS Enterprise Case Management 242

Search Options 242

Search Panel 242

Search Box 243

Search Box Error Messages 243

SAS Enterprise Case Management Time-Out 244

Returning to SAS Workflow Administrator after Timing Out 244

Financial Items Warning Message 245

Transaction Sequential Number Is Not Assigned After an E-File Is Generated . 245

Incorrect or Missing Translations 245

Field Definition Changes and Search Page Errors 246

Warning While Uploading a UI Definition with Required Fields 246

SAS Lock Down State Not Supported 246

Uploading Large Attachments 247

Main Navigation Menu Errors When Logging in 247

Application Is Not Finding Data in the Database 247

Notifications are Not Being Received 248

Maintaining Database Statistics 248
Overview 248
Maintaining Statistics in Oracle 249
Maintaining Statistics in DB2 249
Maintaining Statistics in SQL Server 249
Maintaining Statistics in PostgreSQL 249

Browser Not Supported Message in Internet Explorer 9 249

Enabling Right-to-Left Page Layout 249

Workflow Status Updates

After you select and open a case in SAS Enterprise Case Management, you can update the task status for that case on the Action Items panel. In the **Task Transition** column, select the needed task status for a task. Then, when you select **Save**, the task status changes are saved. The time that the task status was changed is then listed in the **Completed Date** column.

Note: The time that is displayed in the **Completed Date** column is the SAS Enterprise Case Management server time.

Report Workflow Transition Problems

The sample report workflow, FINCENReport.xml, is designed to work with reports that have been created through the user interface. The workflow assumes the report starts in draft status when the workflow is activated. Reports that are loaded through the web service as ready to e-file do not propagate through that workflow process properly. That situation can result in the workflow having the **In Process** and **Manager Review** tasks both active. If regulatory reports that are already set to file are to be loaded, they should either be created with no workflow or with a custom workflow specific to that situation.

Unable to Open SAS Workflow Studio

Access to SAS Workflow Studio is controlled by permissions. If you have an administrative account that cannot access workflow, follow these steps to give the account access by giving it the needed permissions:

1. In a browser, navigate to `<server>/SASAdmin`.
2. Expand **SAS Web Administration Console** ⇨ **Environment Management** ⇨ **Authorization** ⇨ **Assign Roles**.
3. Click **Assign Roles**.
4. Select whether to assign the new role to a user or group and click **Next**.

5. Choose the group name and click **Next**,
6. Select the workflow roles required and click **Finish**.

After the user or group has the proper role, SAS Workflow Studio should be available.

Database Error Warnings and SAS Deployment Wizard

When you are using the SAS Deployment Wizard to install SAS Enterprise Case Management, you might encounter possible warnings when configuring your database. If a yellow check is listed during the configure step in the SAS Deployment Wizard, a warning was encountered during your configuration. This is most likely a database error and is received for any of the following reasons:

- The database doesn't exist.
- The user name (schema) does not exist on the database.
- The user name and database exist but the tables have already been created.

In addition, if you receive the warning message **`SAS Enterprise Case Management Server-Tier Configuration Failed to Deploy Successfully because of an invalid database connection`**, you should be aware that on some platforms, SAS programs will fail if the relational database version does not match the default SAS/ACCESS configuration. On UNIX platforms, SAS/ACCESS needs to be configured to the correct version of the relational database.

Note: Errors have been reported when Oracle schema users don't have the authority to create views. In this case, the sequences, tables, and indexes are created, but the views are not. Resolving this problem might require manually dropping and recreating the files using the scripts in ***`SAS-configuration-directory`***`/Lev<`***`num`***`>/Applications/SASEnterpriseCaseManagement/6.2/Source/sasmisc/install/Oracle`.

Post-installation Database Steps Required after Unsuccessful SAS Deployment Wizard Database Installation

Note: In steps 1 and 2, specify your applicable database.

Note: These steps should be run using the SAS Enterprise Case Management user created during the pre-installation steps.

1. Run the following from your native database client, depending on your platform:
   - Windows platforms: ***`SAS-installation-directory`***`\SASFoundation\9.4\casemgmtmva\sasmisc\install\ddl\`***`dbname`***`\drop_ddl.sql`
   - UNIX platforms: ***`SAS-installation-directory`***`/SASFoundation/9.4/misc/casemgmtmva/install/ddl/`***`dbname`***`/drop_ddl.sql`
2. Run the following from your native database client, depending on your platform:

- Windows platforms: ***SAS-installation-directory*\SASFoundation\9.4\casemgmtmva\sasmisc\install\ddl*dbname*\load_ddl.sql**
- UNIX platforms: ***SAS-installation-directory*/SASFoundation/9.4/misc/casemgmtmva/install/ddl/*dbname*/load_ddl.sql**

  Note: If you are running the script in the SQL Server Management Studio, replace each semicolon (;) with the word "go" to prevent errors.

3. Run the following from your native database client, depending on your platform:
   - Windows platforms: ***SAS-installation-directory*\SASFoundation\9.4\casemgmtmva\sasmisc\install\config\load_install_data.sql**
   - UNIX platforms: ***SAS-installation-directory*/SASFoundation/9.4/misc/casemgmtmva/install/config/load_install_data.sql**
4. Return to the steps in "Loading the SAS Enterprise Case Management Configuration Tables" on page 40.

ODBC Database Transcode Error

If you see the following error message in a SAS log, SAS/ACCESS to ODBC might have a problem with transcoding the non-ASCII data stored in the ECM_DB library:

```
Unable to transcode data to/from UCS-2 encoding
```

If this is the case, you should either upgrade the ODBC driver on the SAS server to support non-ASCII characters, or use SAS (English) as the default SAS server. To set the default SAS server to SAS (English), edit the -CONFIG option of ***SAS-installation-directory*/SASV9.CFG** to use ***SAS-installation-directory*/nls/en/SASV9.CFG**.

For example:

```
-CONFIG "C:\Program Files\SAS\SASFoundation\9.4\nls\en\SASV9.CFG"
```

Case Network Graph Stops Working

If a valid entity table name other than "PARTY" is entered in the UDF_TABLE_NM column of the PARTY_UDF_DEF table, the case network will stop working. An example of a valid entity name is "CASE." If you customize this table and column, you need to comply with the following proper naming conventions:

- The length must be 3–30 characters.
- The first two characters must be "X_" .
- The characters following "X_" can be any combination of uppercase letters, numbers, and underscores.
- The name must be unique with respect to all other static and user-defined table names.

Case Network Analysis Web Service Not Created

After SAS Enterprise Case Management is installed and configured, the SAS stored processes getSocialNetwork, getSocialNetworkNodeDetails, and growSocialNetworkNodes should be registered as a web service so that the Case Network Analysis web component can use them. The registration can be verified by looking at the list of registered web services. You can see the list by browsing to **`http://WebApplicationServerHostName:SASServer1PortNumber/SASBIWS/services/ECMSocialNetworkAnalysis.wsdl`**. Look for a link called "ECMSocialNetworkAnalysis." If the link is there, the web service has been correctly registered. If the ECMSocialNetworkAnalysis web service has not been registered, follow these steps:

1. In SAS Management Console, use the **Folders** tab to navigate to the folder that was just imported to the **`system/applications/SAS Enterprise Case Management/Ent Case mgmt Configuration 6.2/Application SAS code`** folder.
2. Select the **Application SAS code** folder. The **Stored Process** icons appear in the right pane.
3. Hold down the CTRL key and click to select the following stored process icons:
   - **getSocialNetwork**
   - **getSocialNetworkNodeDetails**
   - **growSocialNetworkNode**
4. Right-click one of the selected icons and select **Deploy As Web Service**. The Deploy As Web Service wizard starts.
5. On the Web Service Information window, confirm the following:
   - Use the default value for **Web Service Maker URL**.
   - Use **`ECMSocialNetworkAnalysis`** for **New Web Service Name**.
   - Select **Next.**

   Note: The choice of credentials to use does not matter.
6. On the Web Service Keywords and Namespace window, provide the following value for the Namespace field: **`http://sas.com/sso/fraud/sna`**.
7. Select **Next**.
8. Confirm the settings and then select **Finish**. Open the following page in a web browser and view the available services:

   `http://WebApplicationServerHostName:WebServerPortNumber/SASBIWS/services/ECMSocialNetworkAnalysis.wsdl`

   The new web service ECMSocialNetworkAnalysis is listed. If you click on it, you can see its WSDL, which is similar to the following.

Display A2.1 *WSDL*


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="ECMSocialNetworkAnalysis" targetNamespace="http://sas.com/sso/fraud/sna" xmlns:tns="http://sas.com/sso/fraud/sna"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:typesns="http://sas.com/sso/fraud/sna"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
  - <types>
    - <xs:schema elementFormDefault="qualified" targetNamespace="http://www.sas.com/sso/fraud/sna/score"
        xmlns="http://www.sas.com/sso/fraud/sna/score" xmlns:od="urn:schemas-microsoft-com:officedata"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
      - <xs:element name="TABLE">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" ref="COLORMODEL" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="COLORMODEL">
        - <xs:complexType>
          - <xs:sequence>
            - <xs:element minOccurs="0" name="FMTNAME" od:jetType="text" od:sqlSType="nvarchar">
              - <xs:simpleType>
                - <xs:restriction base="xs:string">
                    <xs:maxLength value="32" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            - <xs:element minOccurs="0" name="START" od:jetType="text" od:sqlSType="nvarchar">
              - <xs:simpleType>
                - <xs:restriction base="xs:string">
                    <xs:maxLength value="16" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            - <xs:element minOccurs="0" name="END" od:jetType="text" od:sqlSType="nvarchar">
              - <xs:simpleType>
                - <xs:restriction base="xs:string">
                    <xs:maxLength value="16" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            - <xs:element minOccurs="0" name="LABEL" od:jetType="text" od:sqlSType="nvarchar">
              - <xs:simpleType>
                + <xs:restriction base="xs:string">
                </xs:simpleType>
              </xs:element>
            - <xs:element minOccurs="0" name="TYPE" od:jetType="text" od:sqlSType="nvarchar">
```


If the ECMSocialNetwork web service does not show up in the SAS Business Intelligence Web Services application, go to SAS Management Console and check whether the same service exists by clicking on the **Plug-ins** tab and then navigating to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **BI Web service for Java 9.4** ⇨ **WebServiceMaker**. If it exists in SAS Management Console but not in the SAS Business Intelligence Web Services application, the web service is not configured properly. Delete the web service from this location and redeploy the stored processes again.

Special Characters Are Missing in Case Network Analysis and Report Mart

If you are using Oracle UTF-8 database on the SAS server machine where the Oracle client is used, ensure that the NLS_LANG setting is for UTF8. For American English, it should be american_america.AL32UTF8.

Specifying the Version Number for SAS Enterprise Case Management

If the SAS Enterprise Case Management version number is not specified in the SAS Enterprise Case Management database, the SAS Enterprise Case Management web application will not execute correctly when you attempt to log on.

DBMS JAR File and Multiple Machine Installations

In a multiple-machine installation, the SAS Deployment Wizard prompts you for the name of the DBMS JAR file that is used on the middle tier. However, this file might not be available on the middle tier because the DBMS JAR file is installed on the server machine. If that is the case, you need to transfer this file to the middle-tier machine before the middle-tier installation.

Note: The JDBC JAR files should be copied to a secure location where they will be kept for the life of the application.

Assigning the Primary Owner to a Case

When a case is created in SAS Enterprise Case Management, a user is assigned as the Primary Owner of the case. The Primary Owner is determined by settings for CASE_CONFIG. If there is a Primary Owner configured in CASE_CONFIG for the case type, category, and subcategory, then that user will be designated as the Primary Owner. SAS Enterprise Case Management automatically sets the Primary Owner for a case if there is not a Primary Owner configured for the case and the Primary Owner is not assigned through the UI definition when a case is created. In this scenario, the first person to edit the case after it has been saved automatically becomes the Primary Owner.

You can also assign the Primary Owner to a case if you are currently the Primary Owner. On the **Results** panel of the Search Cases window, select the **Actions** menu for a case. Select **Set Primary Owner**. The Set Primary Owner dialog box appears. You can now select an owner from the **New owner** drop-down list. Select **OK** to save the change.

Adding the Custom Column Type VARCHAR

When adding a custom column of type VARCHAR, make sure the **`max_char_cnt`** is a number greater than 0, preferably the maximum possible size for your custom column.

Locking and Unlocking an Entity

In SAS Enterprise Case Management, you can lock a case for restricted use or unlock a case to enable another user to edit. You can access the **Lock** and **Unlock** options for a case from the case **Actions** menu. If you do not have access to these options for a case, they will be inactive.

To lock a case, select **Lock** from the **Actions** menu. Locking a case disables the **Edit** and **Unlock** options for that case for other users. You can, however, view the case. If you try to edit a case that is locked, and you do not have access to the case, a message appears, stating that the case is locked by another user. If a case is already locked by you or another user, the **Lock** option is disabled.

If you have access to a locked case, you can unlock the case. To unlock a case, select **Unlock** from the **Actions** menu. When you unlock a case, a message will state that the case is unlocked. The **Unlock** option is disabled if the case is already unlocked or you do not have access to unlock it.

Another way to lock a case is to edit a case. You can select the **Edit** option for a case from the **Actions** menu, or you can select the case from the **Case ID** column in the case Results panel. This automatically locks the case and opens the case for editing. If the case is locked by another user, the **Edit** option is disabled in the **Actions** menu.

When you are finished editing and leave a case, it automatically becomes unlocked. The case also becomes unlocked if your session times out before you leave the case.

A case can have other cases associated with it. An associated case is identified by the **Case ID** on the **Associated Cases** tab of the **Case Information** panel. If the associated case is unlocked, clicking on the linked **Case ID** locks the associated case and opens it for editing.

Using the Search Functionality in SAS Enterprise Case Management

Search Options

When working in SAS Enterprise Case Management, you might need to search for existing entities. There are two options for searching in SAS Enterprise Case Management. You can use the Search panel associated with a specific entity to search within that entity. If it is enabled, you can also search across all entities by using the search box at the top of the page.

Search Panel

The Search panel is available for each specific entity. It contains three options that enable you to modify the search criteria and results that are displayed. You can select from the available search fields and then select one of the following options:

Search

The **Search** option enables you to search for existing cases, incidents, or subjects based on the search fields that are selected. It is possible that one or more search

fields are selected by default. Enter the search criteria needed in the available search fields. Select **Search**. Any existing records that match the search criteria are displayed in the Results panel.

Clear
The **Clear** option clears all search field selections and any records that are displayed in the Results panel. This includes any search fields that have been selected by default.

Reset
The **Reset** option resets the search fields to their initial value. This option can be used to reset the search fields to their initial value *before* a search is performed. Fields that are selected by default and records that are displayed in the Results panel are not affected by the **Reset** option. Select **Reset.** Any search fields that were changed *before* a search is executed are reset to their initial value.

Note: If the **Clear** option is used, the **Reset** option does not reset any fields that were select by default. Those fields remain cleared.

Search Box

If it is enabled, the search box is always visible in the SAS Enterprise Case Management banner. The search box enables you to do a broad search across entities based on the keyword you enter. Type a query in the box and submit it. The results are shown in a list. Click on the entity to see its details

The search box in the banner is visible when all three of the following criteria are met:

- ECM.Search.Enabled is set to `true` in SAS Management Console.
- At least one valid search configuration has been uploaded.
- Solr is running and accessible to search.

ECM.Search.URL must be set to the URL you are configuring search for in SAS Management Console. It typically looks like http://<*server*>:<**port**>/solr/<*collection_name*>.

The search capability is used to restrict access to re-indexing of the search data. The Bulk Load Index role is associated with Case Management: Advanced and Case Management: Administrator groups by default. Without this capability, you cannot clear or bulk load the index from the HTTP services.

The upload search configuration capability is used to restrict access to search configuration files. The Upload Search Configuration role is associated with Case Management: Advanced and Case Management: Administrator groups by default. Without this capability, you cannot upload or delete search configurations.

Search Box Error Messages

Here are the common error messages you might receive if a problem arises with your search, along with their meaning and possible resolutions:

`There is a problem with the search configuration. The search configuration was not saved.`
You tried to upload a bad search configuration.

`Search is currently unavailable. Please contact your system administrator.`

This could be caused by any of the following:

- You tried to access the search page when there are no search configurations.
- You tried to access the search page when Solr could not be contacted.
- You tried to access the search page when search is disabled.

It is recommended that the system administrator check the server logs for more details. Make sure search is enabled in SAS Management Console, a valid search URL has been set in SAS Management Console, and at least one search configuration has been uploaded.

SAS Enterprise Case Management Time-Out

When you are working in SAS Enterprise Case Management, your session can expire if a time-out period has elapsed with no user activity. The default time-out period is five minutes before the actual session time-out occurs. Click **Return to Application** to log on again.

When your HTTP session times out, the system can automatically unlock objects that you had locked—either the objects that were locked during the current session or all objects that you had locked. The unlocking of objects is controlled by the following SAS Enterprise Case Management properties set in metadata:

ECM.Policy.AutoUnlockOnLogout
: The value is **`true`** or **`false`**. This property controls whether any unlock process is run during logout or session time-out.

ECM.Policy.AutoUnlockStrategy
: The value is **`session`** or **`all`**. This property controls which approach the system uses. The default approach is to unlock only the objects locked during the particular session. The other option removes all of your locks. Do not set this property to **`all`** if users access the system concurrently from multiple browsers or devices.

To modify these metadata properties, open SAS Management Console. On the **Plug-ins** tab, go to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Ent Case Mgmt Mid-Tier 6.2**. Right-click **Ent Case Mgmt Mid-Tier 6.2**, select **Properties**, and click the **Advanced** tab. Edit the properties as necessary.

Those changes take place the next time the SAS Enterprise Case Management web application, typically on SASServer8, is restarted.

Returning to SAS Workflow Administrator after Timing Out

After timing out of SAS Enterprise Case Management, you might be redirected to the SAS Workflow Administration application. This can occur even when you select **Continue** from the SAS Enterprise Case Management time-out warning dialog box. In this scenario, the SAS Enterprise Case Management session timer is being reset. However, the SAS Workflow Administration session timer is not being reset.

You can set the SAS Workflow Administration session time-out value to some number of minutes or hours that would keep it from timing out during a user's normal working conditions.

To change this value, edit SAS Workflow Administrator's web.xml file found at ***web application server*\SASServer7_1\sas_webapps\sas.workflow.adminconsole.war\WEB-INF\web.xml**.

```
<session-config>
      <session-timeout>31</session-timeout>
</session-config>
```

The value represents the number of minutes before time-out.

Financial Items Warning Message

The following warning message appears in the log if there are financial items associated with a case, but the UI definition is not configured to display financial items:

Warning: A case that is not configured with financial items may have had financial items copied from an associated incident. These financial items must be configured to be visible. Refer to the SAS Enterprise Case Management Administrator's Guide for details.

You can ignore this message if the case does not need to show financial items. There is an association being saved between the case and the financial items that can be displayed at any time. For information about how to display financial items associated with cases, see the FinancialItemsTable component in the Custom Page Builder documentation, found on the **Administration** menu of SAS Enterprise Case Management.

Transaction Sequential Number Is Not Assigned After an E-File Is Generated

If an e-file is generated successfully and the transaction sequence numbers of the reports under the e-file are not updated, the user might not have permission to update reports. To fix this, see "Enable Report Update in the E-File Stored Process" on page 176.

Incorrect or Missing Translations

You might experience either of the following scenarios:

- A new custom resource bundle has been loaded, and the report mart labels have been refreshed, but the application is not showing the latest translations.
- New values have been loaded into the REF_TABLE_TRANS table, but the values are not being displayed in the application drop-down menus.

If so, consider the following possible causes:

- The custom resource bundle is not following Java naming conventions for the language and country that it supports. See "Custom Resource Bundles" on page 77 for more information on the file naming convention.
- The ECM_LOCALE table does not have an entry for the specified locale. This table is used only for translations stored in the database, such as reference table translations as well as table and column label translations.
- The application has cached values of the old translations. In this case, go to the **Administration** menu and select **Clear Cache**.

Field Definition Changes and Search Page Errors

SAS Enterprise Case Management can have errors on the search page if the type of a user-defined field changes after data has been loaded for that field. For example, suppose a field is originally created with UDF_TYPE_NM = 'VARCHAR' and records are created for that field. If the UDF_TYPE_NM field definition record is then changed to 'DATE', the search page will show an error when those records are part of the returned results. This is because the existing string values cannot be properly converted to date values.

The recommended solution is to avoid the problem by never updating the UDF_TYPE_NM of a field definition that has existing values. If you need to change the type of data being stored for a field, create a new field with the proper configuration. If this problem is happening in your environment, contact technical support for help with cleaning up the obsolete data.

Warning While Uploading a UI Definition with Required Fields

If you create a UI definition that displays a required field as ReadOnly, you will get a warning similar to the following when attempting to upload the UI definition: `Field "CASE.CASE_ID is required by the data model but marked "readonly" in the screen definition.`

This warning is harmless and can be ignored. However, you can also remove the warning by adding the field a second time to the UI definition, as a hidden field. You can see this technique being used for entity ID fields in many of the SAS Enterprise Case Management sample UI definitions.

The following is an example of how to avoid the warning when uploading:

```
<field type="hidden" name="EFILE.EFILE_ID" readonly="false"/>
<field type="string" length="32" name="EFILE.EFILE_ID" readonly="true"></field>
```

SAS Lock Down State Not Supported

The reach and activities of a SAS server can be limited by putting the server in a locked-down state. However, the lock down feature is not supported by SAS Enterprise Case

Management. For more information, see the *SAS Intelligence Platform: Security Administration Guide*.

Uploading Large Attachments

You might receive an error message in your browser when attempting to upload large files as attachments. Many browsers have a limit to the size of files that they will upload, such as 4GB for Internet Explorer 9 and 10. If you have a large file that needs to be added as an attachment, use the web service to work around this limitation.

Main Navigation Menu Errors When Logging in

When you upload a new menu definition, validation is done to ensure that the basic structure of the menu XML is correct and that any functions used are defined. If any problems are detected, an error message is displayed that gives the location of the error(s), and the user is given the choice of re-uploading or cancelling the operation.

If a version of MainNavigationMenu.xml is uploaded with an undetectable problem (for example, calling a function that throws a runtime exception), then the menu definition uploads successfully, but fails to load when a user logs in. Users receive the following error message:

```
An error occurred loading the main navigation menu. Contact your
System Administrator.
```

In this scenario, SAS Enterprise Case Management loads and uses a default navigation menu. This menu gives administrators the ability to upload menu definitions so that they can fix the problem. Non-administrative users will not have access to the system until a corrected menu definition is uploaded.

Application Is Not Finding Data in the Database

The database queries that are issued by the SAS Enterprise Case Management web application include an explicit database schema name. Depending on how the database account is configured, there can be situations where Enterprise Case Management appears to be unable to see data in the database. For example, when configuring a PostgreSQL environment it is common for the user to read and write to the schema named “public.” If Enterprise Case Management is trying to read and write to a schema that matches the user name but Postgres is expecting to read and write from the public schema, the web application will not find the database tables that were created during configuration.

To change the database schema for the web application complete the following steps in SAS Management Console:

1. From the Plug-ins tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Ent Case Mgmt Mid-Tier 6.2**, and right-click on **Properties**.
2. Click the **Advanced** tab, and edit the value for DB.Schema.

3. Restart the middle tier.

Notifications are Not Being Received

If your notifications are not being received, review the following settings.

- Make sure that notifications are configured as you expect. In SAS Management Console, under the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**. Click **Properties**. On the **Settings** tab, click **Notifications**. Make sure the appropriate type of notification is in the **Selected** box.
- If you are expecting e-mail notifications, check the **Advanced** tab of the same dialog box mentioned previously and check the values for Email.Host and Email.Port.
- Each person must also review his or her SAS preferences to confirm that the notifications are being delivered as expected. From SAS Enterprise Case Management, click and select **Preferences**. Select **Notifications**, and review the list of selected notification types.

Maintaining Database Statistics

Overview

For SAS Enterprise Case Management to perform efficiently, database systems must maintain statistical information about the structure and distribution of data stored in each table and index. Without accurate information, the database is unable to determine the most efficient execution path for SQL statements executed by the application. If this information is missing or has become outdated because of significant changes to the data, then performance will suffer dramatically. Therefore, these statistics should be updated as part of your regular database maintenance process.

The process for monitoring and updating optimizer statistics is specific to the database in use. Since statistics-gathering can be very time consuming for large databases, the database systems provide a number of options and utilities. The best strategy for solving problems with statistics depends on factors such as the database system, the number of tables and indexes affected, and the volume of data in the affected tables. SAS Enterprise Case Management provides sample scripts to update database statistics. However, further assistance from a qualified DBA is recommended.

The sample scripts for updating database statistics can be found here:

- Windows platforms: *SAS-installation-directory*`\SASFoundation\9.4\casemgmtmva\sasmisc\sample\dbscript\<`*`dbname`*`>`
- UNIX platforms: *SAS-installation-directory*`/SASFoundation/9.4/misc/casemgmtmva/sample/dbscript/<`*`dbname`*`>`

Maintaining Statistics in Oracle

For more information about maintaining statistics in an Oracle database, see *Oracle Database Performance Guide and Reference*, which is available on the Oracle web site.

Maintaining Statistics in DB2

For more information about maintaining statistics in a DB2 database, see *IBM DB2 Universal Database Administration Guide: Performance*, which is available on the IBM web site.

Maintaining Statistics in SQL Server

For more information about maintaining statistics in a SQL Server database, see *SQL Server Reference Manual*, which is available on the SQL Server web site.

Maintaining Statistics in PostgreSQL

For information about maintaining statistics in a PostgreSQL database, see chapter 23 of *PostgreSQL Routine Database Maintenance Tasks*.

Browser Not Supported Message in Internet Explorer 9

If **Compatibility View** is turned on in Internet Explorer 9, it might cause SAS Enterprise Case Management to display an error stating that the browser is not supported. To resolve this issue, make sure that **Compatibility View** is turned off.

Enabling Right-to-Left Page Layout

To enable right-to-left page layout, add a property named RTL.SupportedLocales to the SAS Enterprise Case Management middle-tier application object. The value for the property should be a comma separated list of locales to render right to left. The list should include the exact locales the browser will be sending (for example, “ar-SA,ar-DZ,ar,he”).

To add or update that metadata property, open SAS Management Console. On the **Plug-ins** tab, go to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Ent Case Mgmt Mid-Tier 6.2**. Right-click **Ent Case Mgmt Mid-Tier 6.2**, select **Properties**, and click the **Advanced** tab. Edit the properties as necessary.

Those changes take place the next time the SAS Enterprise Case Management web application, typically on SASServer8, is restarted.

Appendix 3

SASMSG and %SMD2DS

How Does SAS Enterprise Case Management Use %SMD2DS and SASMSG? . 251
About the SASMSG Function 251
Formatting 252
Open Code Macro Statements 252
The %SMD2DS Macro 252
Example: Add a Message and Its Translation to Be Used by SASMSG 253
Example: Message Substitutions 254

How Does SAS Enterprise Case Management Use %SMD2DS and SASMSG?

SAS Enterprise Case Management uses the SASMSG function to retrieve translated strings based on the locale of the client. For user-defined reports, it is also possible to use the %SMD2DS macro to add messages that can be used by the SASMSG function.

About the SASMSG Function

The SASMSG function returns a message from a specified data set. The message that is returned is based on the current locale of the client and a specified key. SASMSG uses the following syntax:

```
SASMSG("BASENAME", "KEY", <<"QUOTE"|"DQUOTE"|"NOQUOTE">
 <, "substitution 1", ..., "substitution 7">>)
```

BASENAME is the name of the data set where the message is located, and KEY is the message key. If a key name is specified for a key that does not exist, the key name is returned.

Other parameters include an option to indicate the type of quotation marks added to the message text and strings that are used as substitutions. The default quoting option that is used is DQUOTE.

The SAS message data set must be a 7-bit ASCII data set. Any character that cannot be represented in the 7-bit ASCII encoding is represented in the Unicode escape format,

\uxxxx, where *xxxx* is the base-10 numeric representation for the Unicode value of the character.

The message that is returned is based on the LOCALE system option. The LOCALE option has a value of the form *ll_RR*, where *ll* represents the 2-letter language code and *RR* represents the 2-letter region code. The function does the following:

- If a match is not found, then the function searches for a match with the language only.
- If the pair LOCALE/KEY still is not found, then the function defaults to use the English language (en).
- If the KEY does not exist for English (en), then the KEY name is returned.

Formatting

String substitution is allowed using the format code %S. A maximum of seven string substitutions are allowed.

In some cases, the translation of a message to a language other than English might require that you change the order of the string substitutions.

To change the order of string substitutions, insert an argument number specification, *#nn*, within a formatted string. *nn* is the number of the argument in the substitution list. For example, the following substitution returns a message of "My cat. Your dog."

```
msg = sasmsg("nls.mymsg","IN_CD_LOG","noquote","cat","dog") ;
IN_CD_LOGINFO = My %#1s. Your %#2s
```

However, if you change the order of the arguments as follows, then the message that is returned is "My dog. Your cat."

```
IN_CD_LOGINFO = My %#2s. Your %#1s
```

Open Code Macro Statements

You can use SASMSG in the Open Code Macro with the %SYSFUNC macro function.

Note: Arguments that are passed to a function called by the %SYSFUNC macro must not be in quotation marks. However, arguments that are passed to SASMSG outside of the %SYSFUNC macro must be quoted.

When the SASMSG function is used with the %SYSFUNC macro function, the returned string is wrapped with the %NRBQUOTE function.

The %SMD2DS Macro

The %SMD2DS macro is available in the autocall library. You can use it to create SAS message data sets from .smd files. This macro uses the following syntax:

```
%SMD2DS(DIR=,BASENAME=,LOCALE=,LIB=)
```

The arguments used in this macro are defined as follows:

DIR
is the directory where the .smd files are located.

BASENAME
is the base name of the file to process.

LOCALE
(optional) is the list of included locales separated by a blank. basename.smd is the default file that is processed.

LIB
(optional) is the library where the data set will be created. The default library is WORK.

Note that the parameters DIR and BASENAME are required.

Example: Add a Message and Its Translation to Be Used by SASMSG

This example demonstrates how you can create an English and German version of the same message for SASMSG to use.

To create customized translated messages for SASMSG:

1. Create the new directory **`C:\MyORMsmd`**.
2. In the new directory, create a file and name it new_msg.smd. This file defines the English translations with the following line:

   ```
   my_new_sasmsg1 = Process start time
   ```

3. In the new directory, create a file and name it new_msg_de.smd. This file defines the German translations with the following line:

   ```
   my_new_sasmsg1 = Startzeit f\u00fcr Prozess
   ```

4. Start SAS and submit the following code:

   ```
   /* The libname statement contains the installation specific path. */
   /* Change this path as necessary. */
   libname ormhelp 'C:\Program Files\SAS\SASFoundation\9.4\ormonitormva\sashelp';
   %smd2ds(DIR=C:\MyORMsmd, BASENAME=new_msg, LIB=ormhelp, LOCALE=de);
   ```

5. Use SASMSG and the option LOCALE= to verify that you can use the new messages:

   ```
   options locale=English;
   %put %sysfunc(sasmsg(sashelp.new_msg, my_new_sasmsg1, NOQUOTE)) ;

   options locale=German;
   %put %sysfunc(sasmsg(sashelp.new_msg, my_new_sasmsg1, NOQUOTE)) ;
   run;
   ```

For English, the following string is returned:

```
Process start time
```

For German, the following string is returned:

```
Startzeit für Prozess
```

Example: Message Substitutions

This example demonstrates how to use message substitutions.

To use message substitutions:

1. Add a new message in the new_msg.smd file, as follows:

```
new_msg_with_parms =
  This is the first substitution %#1s and this is the second %#2s
```

2. Pass the following parameters to the new message:

```
options locale=English;
%let parm1=Test1;
%let parm2=Test2;
data _null_;
a= sasmsg("sashelp.new_msg", "new_msg_with_parms",
          "NOQUOTE", "&parm1", "&parm2");
put a=;
run;
```

The following string is returned:

```
This is the first parameter Test1 and this is the second Test2
```

Index

Special Characters

<help-text> 122
%SMD2DS macro 251, 252

A

actors
  defining 80
  defining static actors 80
  dynamically determined 80
Administrative User account
  defining 38
alert notifications
  controlling from SAS Preferences Manager 56
alerts 55
anonymous web access
  disabling 17

B

backup requirements 230
batch jobs 230
batch load events 222

C

cache, clearing 40
capabilities 43
  associating 43
  associating with roles 36
  case capabilities 43
  general 47, 50
  incident capabilities 44
  party capabilities 45
  relational 47
  user interface impact 47
case capabilities 43
  user interface and 47
case configurations 71
case management 1
case network
  graph stops working 238
case network analysis 197
  configuring display labels 201
  configuring displayed data fields 200
  configuring link criteria 198
  logic 202
  process for defining 197
  web service not created 239
case report notifications 54
case routing
  configurations for regional manager setup 225
case routing configuration
  testing 229
case UI definitions
  e-filing 174
case user interface
  adding FinancialItemsTable component to 194
cases
  assigning primary owner to 241
  locking and unlocking 242
character encoding 211
clearing the cache 40
client tier 15
code
  adding custom SAS code 217
column header labels
  search panels 92
column labels
  localizing custom labels 215
columns
  custom type VARCHAR 241
components, custom 132
configuration directory
  backing up 231
configuration tables
  loading 40
  users and groups referenced in 37
configurations

case configurations 71
case routing for regional manager setup 225
customizing 71
display labels 201
displayed data fields 200
e-filing 173
incident configurations 72
link criteria for case network analysis 198
match criteria 189
match criteria for subject search 203
party configurations 73
PostgreSQL database, for multi-tier installation 11
reference tables in database 82
SAS Information Delivery Portal 55
SAS Spelling Correction 59
SQL Server ODBC connection 10
testing case routing configuration 229
user-defined fields in database 67
user-specified, for search panels 87, 90, 91
web service 56
create requests 233
Custom Action component 149
custom column type VARCHAR 241
custom components 132
custom functions 131
custom help 122
Custom Page Builder 98
assigning permissions 100
components 148
custom help 122
customizable user interfaces 99
examples 125
expressions and functions 123
static component field formatters 149
user interface definitions 100
valid XML elements and descriptions for user interface definitions 101
custom properties
uploading 41
custom resource bundles 77
custom SAS code
adding 217
custom table and column labels
localizing 215
custom translated messages 214
examples 253, 254
formatting 252
message substitutions 254
customizable search panels 66
customizable user interfaces 99
customizing 62
configurations 71
customizable search panels 66
data security 76
error messages 126
reference tables 65, 82
resource bundles 77
search panels 87
stored processes, to compute financial summaries 195
user interface definition files 125
user interface definitions 62, 70
user-defined fields 66
user-defined generic data tables 69
workflows 64, 78
customizing installation 39

D

data column labels 201
data fields, displayed
configuring 200
data management jobs
setting up 230
data objects 78
adding 80
types supported in SAS Workflow Studio 79
using regional manager group field in workflow 227
data security
customizing 76
field level 76
record level 76
data tables
customizing user-defined generic tables 69
database character encoding 211
database creation scripts 14
databases
backing up 231
configuring reference tables in 82
configuring user-defined fields in 67
default encoding for supported databases 213
error warnings and SAS Deployment Wizard 237
pre-installation database information 5
SAS Web Infrastructure Platform requirement 7
steps required after unsuccessful SAS Deployment Wizard database installation 237
dates
validating 127
DBCS encoding 212
DBMS jar file 241
decimal digits

specifying number of 127
default file locations 28
definitions
uploading 40
deployment 16
deployment plan 5
detail tab headers 202
display labels
configuring 201
displayable fields
search panels 91
displayed data fields
configuring 200
downloading software 5
drop-down lists
groups and roles as 35
specifying 127
dynamic conditional logic
in user interface definition files 129
dynamically determined actors 80

E

e-files
generating 230
e-filing 170
case UI definitions 174
configuring 173
process 172
user-defined fields 174
encoding 211
default encoding for supported databases 213
restricting maximum length of VARCHAR fields 213
SAS session encoding and DBCS support 212
enterprise case management 1
error messages
customizing 126
error warnings
databases and SAS Deployment Wizard 237
ETL processes
creating batch load events 222
event alert notifications 52
event logging 219
creating batch load events 222
currently supported events 219
expressions 123

F

field labels 91
search panels 89
field-level data security 76
fields
customizing user-defined fields 66
displayable 91
filterable 90
hiding required fields 126
required and non-required 125
restricting maximum length of VARCHAR fields 213
searchable 88
specifying read-only fields 126
file locations, default 28
filters
filterable fields 90
search filters 90
financial items 193
adding FinancialItemsTable component to case or incident user interface 194
defining item types in reference tables 193
defining UDF for financial summaries 194
defining UDF for financial transactions 194
defining user interface 194
warning message 245
financial summaries
customizing stored precesses to compute 195
defining UDF for 194
financial transactions
defining UDF for 194
FinancialItemsTable component
adding to case or incident user interface 194
functions 123
custom 131

G

general capabilities 47, 50
generic data tables, user-defined
customizing 69
groups 34
as drop-down lists 35
creating, for each region 226
defining 34
defining in SAS Management Console 38
in workflow definitions 35
referenced in configuration tables 37
referenced in user interface definitions 37

H
help, custom 122
hiding required fields 126
HttpRequest policy 81

I
incident capabilities 44
  user interface and 48
incident configurations 72
incident user interface
  adding FinancialItemsTable component to 194
installation 15
  See also post-installation
  See also pre-installation
  customizing 39
  DBMS jar file and multiple machine installations 241
  default file locations 28
  disabling anonymous web access 17
  installed SAS products 16
  reviewing Instructions.html file 28
  SAS Deployment Wizard tasks 16
  SAS Spelling Correction 57
  selecting single- or multi-tier 15
  updating SAS SID file 30
Instructions.html file 28
internationalization 211
  creating and using custom translated messages 214
  default encoding for supported databases 213
  localizing custom table and column labels 215
  localizing reference tables 216
  localizing workflow activities and statuses 216
  naming conventions for locales 214
  restricting maximum length of VARCHAR fields 213
  SAS session encoding and DBCS support 212
  specifying database character encoding 211

J
jar file 241
JDBC drivers 8
jobs
  setting up data management jobs 230

L
labels
  column header labels 92
  configuring display labels 201
  creating, for user-defined fields 77
  field labels 91
  localizing custom table and column labels 215
languages
  See internationalization
license error 30
link criteria
  for case network analysis 198
load events
  creating batch load events 222
loading configuration tables 40
locales
  naming conventions for 214
localizing
  custom table and column labels 215
  reference tables 216
  workflow activities and statuses 216
locking cases 242
logging events
  See event logging
lookup requests 233

M
match criteria
  configuring 189
  configuring for subject search 203
match labels 202
message substitutions 254
messages
  creating and using custom translated messages 214
middle tier 15
multi-tier installation 15
  configuring PostgreSQL database for 11
multiple machine installatinos
  DBMS jar file and 241

N
naming conventions
  for locales 214
notifications 51, 55
  HttpRequest policy 81

O
ODBC
  configuring SQL Server connection 10
Open Code Macro statements 252
operating system requirements
  verifying 4

Oracle database
creating Oracle users 9
installing 9
pre-installation 9

P

party capabilities 45
user interface and 49
party configurations 73
permissions
Custom Page Builder 100
setting user permission to report mart directory 39
post-installation 32
database steps required after unsuccessful SAS Deployment Wizard installation 237
PostgreSQL
testing database access 12
PostgreSQL database
configuring for multi-tier installation 11
creating users 11
installing 11
pre-installation 11
pre-installation 4
database information 5
installing PostgreSQL database 11
JDBC drivers 8
Oracle database 9
sample database creation scripts 14
SQL Server database 10
process instance
See workflow instance
process templates
See workflow definitions
properties
uploading 40
uploading custom properties 41

R

radio buttons
specifying 127
read-only fields 126
record-level data security 76
reference tables
adding user-defined tables 85
configuring in database 82
customizing 82
customizing and 65
defining 82
defining financial item types in 193
defining static tables 83
localizing 216
rules and conventions for database fields and values 86
region case user-defined field 225
region user-defined field
adding to user interface definition 227
region user-defined reference table 225
regional manager group case user-defined field 226
regional manager group field
using in workflow 227
regional manager group name
deriving from region 227
regional managers
case routing configurations 225
regions
deriving regional manager group name from 227
related items 189
configuring match criteria 189
relational capabilities 47
user interface and 50
renewal 30
report mart 207
refreshing 230
report mart directory
setting user permission to 39
resource bundles
customizing 77
roles 34
See also actors
as drop-down lists 35
associating capabilities with 36
defining 34, 39
in workflow definitions 35
referenced in user interface definitions 37
routing the review capability of cases
regional manager setup 225

S

SAR (Suspicious Activity Report)
files 186
reports 186
UI 186
SAR reports 170
SAR-DI (Suspicious Activity Report by Depository Institutions) 170
SAS code
adding custom code 217
SAS Content Server
backing up 231
SAS Deployment Wizard 16
database error warnings and 237

database steps required after unsuccessful database installation 237
SAS Download Manager 5
SAS Enterprise Case Management
backing up configuration directory 231
backing up database 231
installation 15
version number 241
SAS Information Delivery Portal
configuring 55
notifications and 55
SAS Management Console
creating a group for each region 226
defining 38
defining users 37
User Manager 35
SAS Metadata
backing up 230
SAS Metadata Repository
search panel properties 92
SAS Preferences Manager
controlling alert notifications from 56
SAS session encoding 212
SAS SID file
updating 30
SAS Spelling Correction 56
SAS Web Infrastructure Platform
backing up database 231
database requirement 7
SAS Workflow Studio
data object types supported 79
uploading user interface definition and workflow 229
SASMSG function 251
example 253
scripts
sample database creation scripts 14
search criteria 87
wildcards and 88
search filters 90
Search function 242
search panels 89
column header labels 92
customizable 66
customizing 87
displayable fields 91
field labels 89, 91
filterable fields 90
SAS Metadata Repository properties 92
search criteria 87
search filters 90
search results 91
searchable fields 88
user interface controls 89
user-specified configurations 87, 90, 91
search requests 233
search results 91
searchable fields
for search panels 88
searches
See subject search
security
See data security
server tier 15
session time-out warning message 244
Set Process Participant policy
adding to workflow 228
SID file 5
updating 30
single-tier installation 15
software, downloading 5
spell checking 56
SQL Server database
configuring ODBC connection 10
creating users 10
installating 10
installing SAS/ACCESS for ODBC 10
installing SAS/ACCESS for SQL Server 10
pre-installation 10
testing database access 11
static actors
defining 80
static component field formatters 149
static reference tables
defining 83
status
localizing workflow statuses 216
workflow status updates 236
workflows 80
stored processes
customizing to compute financial summaries 195
subject search 203
configuring match criteria for 203
logic 204
process 203
subscriptions 51
Suspicious Activity Report
See SAR reports
Suspicious Activity Report by Depository Institutions (SAR-DI)
See SAR-DI (Suspicious Activity Report by Depository Institutions)
swimlanes
See also actors
using regional manager group field in workflow 227

T

table labels
  localizing custom labels 215
task list
  alerts 55
  notifications 54
  reminder timer interval 55
task reminders 51
testing
  access to PostgreSQL database 12
  case routing configuration 229
  SQL Server database access 11
text area
  specifying 127
text fields
  specifying 127
time-out warning message 244
translated messages
  creating and using custom messages 214
translations
  incorrect or missing 245
troubleshooting
  adding custom column type VARCHAR 241
  assigning primary owner to cases 241
  case network analysis web service not created 239
  case network graph stops working 238
  database error warnings and SAS Deployment Wizard 237
  DBMS jar file and multiple machine installations 241
  financial items warning message 245
  incorrect or missing translations 245
  locking and unlocking cases 242
  Search function 242
  session time-out warning message 244
  unsuccessful SAS Deployment Wizard database installation 237
  version number 241
  workflow status updates 236

U

UDF
  defining for financial summaries 194
  defining for financial transactions 194
UI definition files
  See user interface definition files
unlocking cases 242
uploading
  custom properties 41
  definitions and properties 40
  user interface definitions 41
  workflow definitions 42
user accounts
  creating 4
user interface
  adding FinancialItemsTable component to case or incident user interface 194
  capabilities and 47
  defining financial items user interface 194
user interface controls 89
user interface definition files 70, 98
  customizing 125
  dynamic conditional logic in 129
  required 70
user interface definitions
  adding regioin user-defined field to 227
  customizing 62, 70
  deleting 101
  editing 100
  groups, roles, and users referenced in 37
  updating 70
  uploading 41, 100, 229
  valid XML elements and descriptions for 101
  viewing 100
user interfaces, customizable 99
User Manager
  SAS Management Console 35
user permission
  setting to report mart directory 39
user-defined field tables 66
user-defined fields
  configuring in database 67
  customizing 66
  data types 67
  e-filing 174
  example 68
user-defined generic data tables
  customizing 69
user-defined reference tables
  adding 85
user-specified configurations
  search panels 91
users 34
  creating Oracle users 9
  creating PostgreSQL users 11
  creating SQL Server users 10
  defining 34
  defining in SAS Management Console 37
  in workflow definitions 35
  referenced in configuration tables 37
  referenced in user interface definitions 37

V

validating dates 127
VARCHAR custom column type 241
VARCHAR fields
  restricting maximum length of 213
version number 241

W

warning messages
  financial items 245
  session time-out 244
web access, anonymous 17
web service 233
  configuring 56
  not created for case network analysis 239
  sample request 234
wildcards 88
workflow definitions 78
  groups, roles, and users in 35
  uploading 42, 229
workflow instance 78
workflow templates
  modifying 82
workflows
  adding set process participant policy to 228
  customizing 78
  customizing and 64
  data object 78
  defining 78
  defining actors 80
  HttpRequest policy 81
  localizing activities and statuses 216
  status 80
  status updates 236
  using regional manager group field in 227

X

XML elements and descriptions
  for user interface definitions 101