SAS® Marketing Automation 6.5: Administrator’s Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Marketing Automation 6.5: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Marketing Automation 6.5: Administrator's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

October 2016

6.5-P1:maag

Contents

Accessibility

Accessibility Notice

For information about the accessibility of this product, see *SAS Marketing Automation: User's Guide*.

1

Architecture

Architecture of the SAS Intelligence Platform

The SAS Intelligence Platform is designed to efficiently access large amounts of data, while simultaneously providing timely intelligence to a large number of users. The Platform uses an *n*-tier architecture that enables you to distribute functionality across computer resources, so that each type of work is performed by the resources that are most suitable for the job.

A *tier* in the SAS architecture represents a conceptual category of software components that perform similar types of computing tasks and that require similar types of resources. Different tiers do not necessarily represent separate computers or groups of computers. More than one computer can be used for a specified tier as well.

You can modify the SAS architecture to meet the demands of your workload. For a large company, the architecture can easily consist of many computers with different operating environments. For prototyping, demonstrations, or very small enterprises, the components for all of the tiers can be installed on a single computer.

The architecture of the SAS Intelligence Platform consists of the following tiers.

Clients	The client tier provides users with access to intelligence data and to functionality through web-based interfaces. For more information about the client tier for SAS Customer Intelligence solutions, see “Client Tier” on page 8.

Middle tier

The middle tier enables users to access intelligence data and functionality via a web browser. This tier provides web-based interfaces for report creation and information distribution, while passing analysis and processing requests to the SAS servers.

For more information about the middle tier for SAS Customer Intelligence solutions, see “Middle Tier” on page 10.

SAS servers

SAS servers perform SAS processing on your enterprise data. Several types of SAS servers are available to handle different workload types and processing intensities. The software distributes processing loads among server resources so that multiple client requests for information can be met without delay.

For more information about the SAS server tier for SAS Customer Intelligence Solutions, see “SAS Server Tier” on page 14.

Data sources

Data sources store your enterprise data. All of your existing data assets can be used, whether your data is stored in relational database management systems, SAS tables, or enterprise resource planning system (ERP) tables.

For more information about the data tier for SAS Customer Intelligence solutions, see “Data Tier” on page 26.

For more information about the SAS Intelligence Platform, see *SAS Intelligence Platform: Overview* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Customer Intelligence Solutions

Overview of SAS Customer Intelligence Solutions

SAS Customer Intelligence is a suite of marketing solutions that can be installed in different combinations. Your SAS Customer Intelligence installation includes the solutions that are specified by your contract. For a comprehensive list of the SAS components that were installed at your site, see the plan file that was emailed to you by the sas.com Planning Application. The following SAS solutions might be part of your installation of SAS Customer Intelligence.

SAS Digital Marketing

SAS Digital Marketing uses mobile, SMS text message, and email technologies to integrate multimedia messaging capabilities into the broader business marketing strategy. SAS Digital Marketing primarily consists of the SAS platform, several web applications, including a server to interact with an email server and a tracking tier (bess), and SAS Visual Analytics Administration and Reporting. Each implementation of SAS Digital Marketing requires a Mail Transfer Agent (MTA) Server to deliver email to the customer.

The default location for SAS Digital Marketing data is **<*config*>/Lev1/Applications/SASCustomerIntelligence/DigitalMarketing**

SAS Marketing Automation

SAS Marketing Automation creates marketing campaigns quickly while automating processes and providing tracking capabilities. SAS Marketing Automation includes SAS Enterprise Business Intelligence, SAS Data Integration, SAS Enterprise Miner, and SAS Visual Analytics Administration and Reporting.

The default location for SAS Marketing Automation campaign data is **`<config>/Lev1/Applications/SASCustomerIntelligence/CampaignManagement/data`**

SAS Marketing Optimization

SAS Marketing Optimization optimizes marketing campaigns and determines the optimal set of customers to be targeted. Like SAS Marketing Automation, SAS Marketing Optimization also includes SAS Enterprise Business Intelligence and SAS Visual Analytics Administration and Reporting. SAS Enterprise Miner is not included

The default location for SAS Marketing Optimization data is **`<config>/Lev1/Applications/SASCustomerIntelligence/MarketingOptimization`**

SAS Real-Time Decision Manager

SAS Real-Time Decision Manager capitalizes on customer interactions by coordinating marketing campaigns across channels and making real-time recommendations to the customer based on a customized decision flow.

Operating Environments

For system requirements for SAS Customer Intelligence products, see http://support.sas.com/resources/sysreq/index.html. For third-party software requirements, see http://support.sas.com/resources/thirdpartysupport/index.html.

Recommendations for Designing SAS Customer Intelligence Solutions Architecture

Overview of Designing SAS Customer Intelligence Solutions Architecture

SAS Customer Intelligence can be configured in a variety of ways, depending on customer requirements and data volume. These components can be combined on the same physical server. In most cases, the components are spread across multiple servers for better load balancing, availability, or performance.

When you design the architecture for SAS Customer Intelligence solutions at your site, there are several factors that you should consider.

- SAS Web Application Server is the only supported web application server.
- The recommended deployment for the SAS Digital Marketing tracking tier is a third-party reverse proxy server outside a firewall. The tracking tier (Bulk Email Server Servlets or BESS) is deployed inside a firewall. The reverse proxy server must be the SAS Web Application Server. For more information, see *SAS Digital Marketing: Administrator's Guide*.
- SAS Digital Marketing grid-node nodes are not supported. The SAS Digital Marketing application (sdm.war) is now a web application that is deployed within SASServer6_1. It is not eligible for middle-tier clustering. Consider broadcast volumes and future growth when you size a machine for this workload.
- The non-distributed version of SAS Visual Analytics Administration and Reporting (VAAR) is included with SAS Marketing Automation, SAS Marketing Optimization, SAS Digital Marketing, and SAS Real-Time Decision Manager. Allocate a separate machine for this component. If the VAAR middle tier and SAS server tiers are placed on separate machines, the operating system (Windows or Linux) must be the same on both machines.
- When you deploy SAS Real-Time Decision Manager, configure at least two instances of the SAS Real-Time Decision Manager operational environment. Each instance includes a SAS Decision Services engine server, a SAS Federation Server, and a SAS Authentication Server. This configuration provides high availability while balancing the workload to ensure acceptable response times.
- Consider creating one or more unique server contexts for SAS Customer Intelligence. SAS Marketing Optimization, in particular, can take advantage of this configuration because of a demanding workload on SAS server tier resources. For more information, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.
- Select a database carefully. Your choice of databases might be limited, based on the combination of SAS Customer Intelligence solutions at your site. The selection of a database should take into account the possibility of adding a solution in the future to the currently licensed solutions.

SAS Deployment Manager

You can use the SAS Deployment Manager to add additional nodes for SAS Metadata Server and the middle tier. These nodes are automatically defined when the deployment plan is created.

During configuration, the SAS Deployment Manager typically deploys SAS Customer Intelligence web applications to individual managed SAS Web Application Server instances. A deployment that includes all four SAS Customer Intelligence solutions configures six managed servers.

Platform Suite for SAS

SAS Customer Intelligence does not use any grid components from the Platform Suite for SAS. Tasks such as email broadcasts, optimization data uploads, and campaign executions usually require a scheduler. Standard operating system scheduling capabilities might not be flexible enough to meet more demanding tasks. In order to provide robust scheduling capability, you should add Platform Suite for SAS to your components.

SAS Visual Analytics Administration and Reporting

A non-distributed version of SAS Visual Analytics Administration and Reporting is included with each solution. This new addition is represented as an individual server, although it is possible to separate the middle tier from the SAS server tier. Install this component on a separate machine.

You can license the distributed version of SAS Visual Analytics Administration and Reporting or the full SAS Visual Analytics product to integrate with any of the solutions. The full product can be distributed or non-distributed. The distributed version runs only on Linux. The non-distributed versions run on Linux or Windows.

A minimum of 16 cores is recommended for implementation of non-distributed versions. Implementation of distributed versions should include 4 servers and 64 cores. If you license the distributed version of SAS Visual Analytics Administration and Reporting, install the component on a separate collection of machines.

SAS Marketing Automation Topology

The most common practice for SAS Marketing Automation is to designate a separate machine for the SAS server tier, middle tier, client tier, and for SAS Metadata Server. The middle tier and SAS Metadata Server can be clustered. It is possible for all of the tiers to be deployed on a single machine.

The Customer Intelligence LASR Analytic Server can be installed on the SAS server tier or on a separate machine.

For improved performance, the SAS Marketing Automation Solution LASR Analytic Server should be installed on a separate machine. A planning purpose named "Solution LASR Analytic Server" is available in the planning tool on the **Define Machines** tab. If you check this purpose in the planning tool, the SAS Marketing Automation Solution LASR Analytic Server is distributed to the desired machine.

SAS Real-Time Decision Manager Topology

SAS Real-Time Decision Manager consists of two distinct phases of processing.

Architecture for the design phase contains servers on the middle tier and SAS server tier. These servers are configured to manage the design process. During the design process, a campaign is created with a decision flow. The decision flow is designed to make a recommendation based on the eligibility of a customer to receive a particular offer.

When the design phase is complete, the campaign is promoted to the operational environment. The operational environment is separate from the design environment. The separation does not need to be a physical or network separation, although in some cases it might be.

Workload for the operational environment should be load balanced across two or more servers to accommodate high volumes of requests and provide acceptable response time and high availability. The SAS Decision Services Engine Server, SAS Authentication Server, and SAS Federation Server are key components of the operational servers

SAS Authentication Server and SAS Federation Server are typically installed on each middle-tier node.

Load Balancer for SAS Real-Time Decision Manager

Production workloads for SAS Real-Time Decision Manager must be split across two or more operational servers that contain a SAS Decision Services engine server, SAS Federation Server and SAS Authentication Server. In order to balance incoming requests, a load balancer might be necessary. If a load balancer is already installed at a site, configure this component to direct traffic to the operational servers.

Storage Architecture

The storage for SAS Customer Intelligence components is organized into the following categories:

- SAS installation files and binaries
- SAS configuration files and configuration data
- SASWORK
- Customer data
- Common data model

The first three areas are typical of any standard SAS deployment. Many SAS Customer Intelligence processes are driven by stored processes and large customer data sets. It is important to size the SAS work space. Typically, customer data already exists. Storage parameters for those databases are defined. The common data model is stored in a relational database. Consider the space that must accommodate the expected volume of data such as contact history, response history, and presented treatment history tables.

Estimating Size

If all four SAS Customer Intelligence solutions are deployed, there are several factors to consider when you estimate the size of an environment.

SAS servers and middle-tier servers are key components for SAS Marketing Automation and SAS Marketing Optimization. These solutions do not interact directly with a customer or external requests. Consider the number of users, complexity of campaigns, and amount of data.

For SAS Digital Marketing, it is important to consider the SAS Digital Marketing server and the tracking tier. Because the SAS Digital Marketing server cannot be clustered, estimate the size of the web application server to accommodate future growth.

When you plan for SAS Real-Time Decision Manager, consider the volume of transactions and complexity of decision flows that are processed by the SAS Decision Services engine. You should also plan for the number of customer or model processes that are handled by SAS Federation Server.

High Availability

SAS 9.4 improves the availability of the SAS platform. The areas of improved availability are SAS Metadata Server and middle-tier servers. These servers can be clustered with the SAS Deployment Wizard. The planning application automatically generates additional server node instances that can be used to create a cluster. For more information, see *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Not all web applications are candidates for clustering. The SAS Digital Marketing application is on the middle tier, but it is not capable of being clustered. The same is true for SAS Decision Services Monitor.

SAS Federation Server, SAS Authentication Server, SAS Decision Services engine, SAS Customer Intelligence Reporting, and related SAS servers can be replicated to distribute workload for SAS Real-Time Decision Manager while providing higher availability for those components. Each operational server should be configured with a co-located SAS Federation Server, SAS Authentication Server, and SAS Decision Services engine server.

Server Hardware

The choice of server hardware depends on the SAS Customer Intelligence solutions, volume of customer data and broadcasts, complexity of communications and campaigns, and expected response time. Larger volumes of data might require additional servers to ensure acceptable response time. A separate server is recommended for default SAS Visual Analytics Administration and Reporting. If clustering of the middle tier servers or SAS Metadata Server is configured, each clustered node equates to a physical server. A customer who has licensed all four solutions might have ten or more servers in the production environment if clustering is configured. In many cases, development environments might fit on a single server.

Client Tier

Overview of the Client Tier

The SAS Customer Intelligence client tier consists of a web browser such as Internet Explorer, Firefox, or Chrome, and several SAS components. The following topics list the SAS components on the client tier for each solution.

SAS Digital Marketing Client Tier

The following SAS components are installed on the SAS Digital Marketing client tier:

- Digital Marketing Studio
- SAS Management Console
- SAS Workflow Studio

- SAS Enterprise Guide

SAS Mobile BI is installed on SAS Visual Analytics mobile clients.

SAS Marketing Automation Client Tier

The following SAS components are installed on the SAS Marketing Automation client tier:

- SAS Add-In for Microsoft Office
- SAS BI Lineage Plug-in for SAS Management Console
- SAS Data Integration Studio
- SAS Enterprise Guide
- SAS Enterprise Miner Client
- SAS Enterprise Miner Plug-in for SAS Management Console
- SAS Esri Map Manager Plug-in for SAS Management Console
- SAS Information Map Studio
- SAS Management Console
- SAS Metadata Bridges for General Industry Standards
- SAS Metadata Bridges Plug-in for SAS Data Integration Studio
- SAS Metadata Bridges Plug-in for SAS Management Console
- SAS OLAP Cube Studio
- SAS OLAP Server Monitor
- SAS Workflow Studio

SAS Mobile BI is installed on SAS Visual Analytics mobile clients.

SAS Marketing Optimization Client Tier

The following SAS components are installed on the SAS Marketing Optimization client tier:

- SAS Add-In for Microsoft Office
- SAS BI Lineage Plug-in for SAS Management Console
- SAS Data Integration Studio
- SAS Enterprise Guide
- SAS Esri Map Manager Plug-in for SAS Management Console
- SAS Information Map Studio
- SAS Management Console
- SAS Metadata Bridges for General Industry Standards
- SAS Metadata Bridges Plug-in for SAS Data Integration Studio
- SAS Metadata Bridges Plug-in for SAS Management Console
- SAS OLAP Cube Studio
- SAS OLAP Server Monitor

- SAS Workflow Studio

SAS Real-Time Decision Manager Client Tier

The following SAS components are installed on the SAS Real-Time Decision Manager client tier:

- SAS Data Management Studio Clients
  - Base SAS
  - SAS Data Management Studio Platform
- SAS Real-Time Decision Manager Design Clients
  - SAS Add-In for Microsoft Office
  - SAS Information Map Studio
  - SAS Management Console
  - SAS BI Lineage Plug-in for SAS Management Console
  - SAS Data Integration Studio
  - SAS Metadata Bridges
  - SAS Metadata Bridges Plug-in for SAS Data Integration Studio
  - SAS Metadata Bridges Plug-in for SAS Management Console
  - SAS Workflow Studio
  - SAS Enterprise Guide
- SAS Decision Services
  - SAS Decision Services Plug-ins for SAS Management Console
  - SAS Management Console
  - SAS Web Infrastructure Client
  - SAS Drivers for Federation Server

SAS Mobile BI is installed on SAS Visual Analytics mobile clients.

Middle Tier

Overview of the Middle Tier

The SAS Customer Intelligence resources on the middle tier provide the analytical processing power for the applications that perform marketing activities. Marketing activities include segmenting customers as well as creating, scheduling, and executing campaigns.

In development environments or very small environments, the SAS Customer Intelligence middle tier might be deployed on the same machine as the SAS servers. In cases where there are large volumes of data and transactions, or there are a large number of users, place the middle tier on a separate host. If high availability or additional capacity is required, the SAS Web Application Server should be horizontally clustered. In a horizontal cluster, multiple instances of managed web application servers are deployed across multiple hosts.

For more information, see *SAS Intelligence Platform: Middle-Tier Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Web Application Server

The middle tier is supported only on SAS Web Application Server. SAS Web Application Server is based on VMware vFabric tc Server. External third-party application servers are neither required nor supported.

Reverse Proxy Server

A standard SAS Customer Intelligence deployment now also includes the SAS Web Server. The SAS Web Server is an HTTP, HTTPS reverse-proxy server that is configured as a single connection point for SAS web applications. If you cluster the SAS Web Application Server, the SAS Web Server is configured automatically to perform load balancing. Secure Sockets Layer (SSL) can be enabled during configuration with the SAS Deployment Wizard to provide secure communication.

This server can be deployed in the DMZ portion of the network or in the domain with the remaining SAS server tier and middle-tier servers. The SAS Deployment Wizard can automatically configure the SAS Web Server to provide secure communications (HTTPS).

During SAS Deployment Wizard configuration, a tracking tier application is generated for SAS Digital Marketing. The application can be deployed inside the firewall along with other web applications. It can be deployed inside the firewall with a proxy version deployed outside the firewall. It can be deployed inside the firewall with a reverse proxy server outside the firewall. The preferred method for deployment is in a reverse proxy configuration with a reverse proxy server deployed outside a firewall and the tracking tier (bess.ear) for SAS Digital Marketing deployed inside the firewall. The reverse proxy server must be the SAS Web Application Server. The SAS Customer Intelligence resources on the middle tier provide the SAS Web Application Server on which SAS Digital Marketing Server (sdm.war) and bess.war are deployed. The bess_proxy.war should be deployed on a separate SAS Web Application Server.

SAS Environment Manager

SAS Environment Manager is a web application that can monitor, track, and manage SAS resources, including SAS Foundation servers and the SAS Web Application Server. SAS Environment Manager can be used to build dashboards that present performance, resource consumption, and availability data from a variety of services. It also provides the capability to monitor logs and trigger alerts based on pre-defined conditions. Making effective use of this tool requires an agent to be deployed on each tier.

SAS Studio

SAS Studio is a development application for SAS that you access through your web browser. With SAS Studio, you can access your data files, libraries, and existing programs, and you can write new programs. You can also use the predefined tasks in SAS Studio to generate SAS code. When you run a program or task, SAS Studio connects to a SAS server to process the SAS code. The SAS server can be a hosted server in a cloud environment, a server in your local environment, or a copy of SAS on your local machine. After the code is processed, the results are returned to SAS Studio in your browser.

In addition to writing and running your own SAS programs, you can use the predefined tasks that are included with SAS Studio to analyze your data. The tasks are based on SAS System procedures and provide access to some of the most commonly used graph and analytical procedures. You can also use the default task template to write your own tasks.

SAS Customer Intelligence Solutions and Middle-Tier Architecture

The SAS Web Server serves as the proxy, or entry point, to the SAS Web Application Servers. The Web Infrastructure Platform and related services, SAS Deployment Backup, SAS Studio, and standard EBI applications reside in SASServer1 and SASServer2. SASServer6 holds SAS Marketing Automation, SAS Marketing Optimization and SAS Digital Marketing. The JVM for SASServer7 houses the SAS Real-Time Decision Manager applications. SAS Enterprise Miner is deployed to SASServer11_1 and applications for SAS Visual Analytics Administration and Reporting are located in SASServer12_1. The SAS Digital Marketing tracking tier can be deployed in one of the existing JVMs, but it is recommended that an isolated JVM server be created for this application. Although the tracking tier application is generated automatically, it must be deployed manually.

Middle-Tier Components for SAS Digital Marketing

The following SAS components are installed on the SAS Digital Marketing middle tier:

- SAS Digital Marketing Web Studio and Web Reporting Infrastructure
- SAS Digital Marketing Web Studio
- SAS Customer Intelligence Reporting Middle-Tier
- SAS Customer Intelligence Studio
- SAS Studio Middle-Tier
- SAS Visual Analytics Administration and Reporting Middle-Tier
- SAS Web Infrastructure Platform
- SAS Web Application Server
- SAS Web Server
- SAS Web Infrastructure Database JDBC Drivers

Middle-Tier Components for SAS Marketing Automation

The following SAS components are installed on the SAS Marketing Automation middle tier:

- SAS BI Dashboard
- SAS BI Portlets
- SAS Customer Intelligence Core Middle-Tier
- SAS Customer Intelligence Reporting Middle-Tier
- SAS Customer Intelligence Studio
- SAS Enterprise Miner Middle-Tier
- SAS Information Delivery Portal Web Application
- SAS Studio Middle-Tier
- SAS Visual Analytics Administration and Reporting Middle-Tier
- SAS Web Application Server
- SAS Web Infrastructure Platform
- SAS Web Report Studio
- SAS Web Parts for Microsoft SharePoint
- SAS Web Server

Middle-Tier Components for SAS Marketing Optimization

The following SAS components are installed on the SAS Marketing Optimization middle tier:

- SAS BI Dashboard
- SAS BI Portlets
- SAS Customer Intelligence Reporting Middle-Tier
- SAS Customer Intelligence Studio
- SAS Marketing Optimization Middle-Tier
- SAS Information Delivery Portal Web Application
- SAS Studio
- SAS Visual Analytics Administration and Reporting Middle-Tier
- SAS Web Application Server
- SAS Web Infrastructure Platform
- SAS Web Parts for Microsoft SharePoint
- SAS Web Report Studio
- SAS Web Server

Middle-Tier Components for SAS Real-Time Decision Manager

In addition to the SAS Customer Intelligence web applications, the SAS Real-Time Decision Manager deployment requires the SAS Authentication Server, SAS Federation Server and SAS Federation Server Manager to configure, manage, authenticate, and execute real-time decisions. SAS Federation Server Manager is the web-based component that uses wizards and dialog boxes to define accounts and database connections in place of using DDL and FedSQL.

The following SAS components are installed on the SAS Real-Time Decision Manager middle tier:

- SAS Business Rules Web Manager
- SAS Customer Intelligence Core Middle-Tier
- SAS Customer Intelligence Reporting Middle-Tier
- SAS Customer Intelligence Studio
- SAS Decision Manager Middle-Tier
- SAS Decision Services Design Middle-Tier
- SAS Decision Services Engine Server
- SAS Decision Services Monitor
- SAS Drivers for JDBC
- SAS Environment Manager
- SAS Environment Manager HQ Plug-in for SAS Decision Services
- SAS Environment Manager HQ Plug-in for SAS Decision Services Engine Administration
- SAS Environment Manager Middle-Tier
- SAS Federation Server Manager
- SAS Studio Middle-Tier
- SAS Visual Analytics Administration and Reporting Middle-Tier
- SAS Web Application Server
- SAS Web Infrastructure Database JDBC Drivers
- SAS Web Infrastructure Platform
- SAS Web Parts for Microsoft SharePoint
- SAS Web Report Studio
- SAS Web Server

SAS Server Tier

Overview of the SAS Server Tier

SAS Customer Intelligence resources on the SAS server tier include several types of SAS servers that handle different workload types and different processing intensities.

Platform Suite for SAS

Platform Suite for SAS includes the following applications from Platform Computing, Inc., that are supported in order to schedule SAS jobs.

- Platform Process Manager for SAS
- Platform LSF (Load Sharing Facility)
- Platform Grid Management Service

Platform Process Manager enables users to schedule SAS Marketing Automation jobs (for example, the recurring executions of either communications or entire campaigns). You send jobs to be scheduled to Platform Process Manager either from SAS Customer Intelligence Studio, or through the Schedule Manager in SAS Management Console.

The following software is required to support job scheduling for SAS Customer Intelligence:

- SAS Customer Intelligence Studio
- Launcher For more information, see "Execute Campaigns and Manage Information Maps" on page 191.
- SAS Data Integration Studio

Configuration files for Platform LSF are located in <*LSF-installation-dir*>`\conf`. Log files are located in <*LSF-installation-dir*> `\logs` .

See also:

- *Scheduling in SAS* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html
- Schedule Manager Help in SAS Management Console

Object Spawner

The object spawner is a process that runs on the SAS server host and listens for requests, including requests from the SAS Customer Intelligence application server. When a request is received, the object spawner accepts the connection and sends a job to the workspace server or to the stored process server.

The object spawner requires the following software to be installed on the computer that is assigned to the SAS server tier:

- SAS 9 Software or later
- SAS Integration Technologies
- SAS Metadata Server

Note: SAS/SECURE is an optional component.

The object spawner uses the following files:

Executable
: ObjectSpawner.bat (Windows) or ObjectSpawner.sh (UNIX) is typically located in <*config-dir*> `Lev1\ObjectSpawner` on the SAS server computer.

Configuration

metaConfig.xml is located in *<config-dir>*`\Lev1\ObjectSpawner`. The configuration file contains the name of the metadata server and the login credentials for the SAS Trusted User.

Log

Objspawn.log is located in `\ObjectSpawner\Logs`. You can view the object spawner log file in order to troubleshoot connections between the client and server computers. The object spawner log file is not created by default. To create the object spawner log file, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

The object spawner must be refreshed to include any changes that are made to autoexec_usermods.sas and sasv9.cfg. For more information, see *SAS Intelligence Platform: Application Server Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Workspace Server

The SAS Workspace Server enables client applications to submit SAS code to a SAS session that uses an application programming interface (API). Your environment can include one or more workspace servers. When IT professionals process a campaign , SAS code is generated by the computation engine on the middle tier, and is submitted to a workspace server.

SAS Workspace Server uses the following files:

Executable

You can find the name and location of your workspace servers by using the SAS Server Manager in SAS Management Console.

Log

Workspace servers are not initially configured to produce log files. For troubleshooting purposes, workspace server logs can be helpful by capturing calls that are made to the database. In these situations, you can use the alternative logging configuration file (logconfig.trace.xml) that is provided in each workspace server's configuration directory. If a log is created, the default location is *<config-dir>*`\Lev1\SASApp\WorkspaceServer\Logs` .

For information about managing logs for the workspace server, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Stored Process Server

The SAS Stored Process Server executes and delivers results from SAS stored processes in a multi-client environment. SAS stored processes are SAS programs that are stored centrally and that can be executed by business users and by client programs on demand.

The SAS Stored Process Server performs work that is similar to the workspace server. For information about the difference between the two servers, see *SAS Intelligence Platform: Application Server Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Stored processes are used in SAS Customer Intelligence to perform computations.

For information about stored process software requirements, see *SAS Stored Processes: Developer's Guide* at http://support.sas.com/documentation/onlinedoc/inttech/index.html.

The SAS Stored Process Server uses the following files:

Executable

In SAS Management Console, the initial stored process server is configured as a load-balancing server named SASApp-Stored Process Server. Find the name and location of the SAS Stored Process Server by using the Server Manager in SAS Management Console.

Configuration

sasv9.cfg is located in *<config-dir>* `\Lev1\SASApp\StoredProcessServer` and calls sasv9.cfg that is located in SASApp.

Log

The stored process server log is located in *<config-dir>*`\Lev1\SASApp\StoredProcessServer\logs` . Log parameters are specified in sasv9_StorProcSrv.cfg. The log file for the stored process server is useful for troubleshooting the execution of Split nodes, Limit nodes, and exports.

For information about managing logs, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Each stored process server process handles multiple users, and by default each server uses multiple server processes. A load-balancing algorithm distributes client requests between the server processes.

By default, the object spawner starts the processes of the stored process server by authenticating the SAS Spawned Servers user ID, `sassrv`.

For information about load balancing, see *SAS Intelligence Platform: Application Server Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Pooled Workspace Server

Stored processes run on the SAS Pooled Workspace Server. For more information, see "Configure Stored Processes for Pooled Workspace Server" on page 165.

SAS Metadata Server

The SAS Metadata Server controls access to a central repository of metadata that is shared by all of the applications that run as part of SAS Customer Intelligence.

Metadata for the following objects is generated and stored for SAS Customer Intelligence:

- information maps that display the physical source data as business data that is used by business users to query a data warehouse
- intermediate tables that are generated when you create campaigns by using SAS Customer Intelligence
- contact history and response history tables

- authorization rules for objects (such as information maps, business contexts, and campaigns) that are used by SAS Customer Intelligence
- user IDs and passwords for SAS Customer Intelligence that are used to access a third-party server (for example, an Oracle database server)
- SAS stored processes that update and execute nodes
- reports about SAS Customer Intelligence that are accessible by using SAS Management Console

The SAS Metadata Server uses the following files:

Configuration

sasv9_MetadataServer.cfg is located in *<config-dir>* `\Lev1\SASMeta\MetadataServer` and points to Sasv9.cfg in *<config-dir>*`\Lev1\SASMeta`. Sasv9.cfg points to the default SAS configuration file at `C:\Program Files\SASHome\SASFoundation\9.4\sasv9.cfg`.

The metadata configuration file contains information about accessing a metadata server. The spawner uses the information that is contained in the configuration file to connect to a metadata server and to read the appropriate server definitions. To enable the spawner to connect to and read the appropriate metadata from a metadata server, you must specify the appropriate login information in the metadata configuration file.

Log

MetadataServer#d#b#y.log is located in *<config-dir>*`\Lev1\SASMeta\MetadataServer\logs`. This log file contains information about users who connect to the SAS Metadata Server . For example, this log might show the status of the SAS Metadata Server and might indicate whether a request from a client application was successful. The log entries also contain information that is useful for diagnosing server start-up problems, diagnosing authorization failures, and debugging method calls.

According to the SAS Metadata Server options that you specify at invocation, the SAS Metadata Server can write the following categories of information to the SAS log:

- the start and stop information for the SAS Metadata Server:
  - the user ID
  - the SAS long version number
  - the SAS Metadata Model version
  - the directory where the SAS Metadata Server was started
  - configuration file options
- user connection and disconnection events
- creation and deletion of a repository
- opening, closing, pausing, resuming, and refreshing events
- errors, such as these:
  - task and thread exceptions
  - memory allocation errors
  - I/O errors
  - application logic errors

  - authentication errors
  - authorization failures
- authentication events
- XML input strings and XML output strings
- traces that are invoked by the debugging options in omaconfig.xml

For detailed documentation, see http://support.sas.com/documentation/onlinedoc/metadatasrvr/index.html.

SAS Federation Server

SAS Federation Servers run the SAS Activities and score code that are based on DS2. DS2 is used by SAS Real-Time Decision Manager.

SAS Digital Marketing Server

If you installed SAS Digital Marketing as an optional client with SAS Marketing Automation, then your site has installed a SAS Digital Marketing server on the SAS server tier that you can access through your network.

In SAS Customer Intelligence Studio, a campaign that includes an email broadcast (that is defined within a Communication node) is routed to the SAS Digital Marketing server. The SAS Digital Marketing server executes the job by applying any filters if applicable (such as opt-out requests) to the list of recipients. The SAS Digital Marketing server tracks the replies. The SAS Digital Marketing server uses Customer Intelligence Common web service to update the common data model with the replies.

The SAS Web Application Server is the application server component of the SAS middle tier. This application server is used for running SAS Java web applications on the SAS 9.4 platform. The SAS Marketing Automation and SAS Digital Marketing integration uses SAS data sets for recipient list.

The SAS Digital Marketing ebtool log statements are available in the SAS Digital Marketing log file. For more information, see *SAS Digital Marketing Administrator’s Guide*.

SAS Digital Marketing can be used with SAS Marketing Automation when both products are installed with SAS Customer Intelligence Studio. SAS Digital Marketing uses the SAS Digital Marketing ebtool utility to execute broadcasts. SAS Digital Marketing updates the common data model that is used by SAS Marketing Automation.

For information about displaying SAS Digital Marketing information in SAS Customer Intelligence Studio, see “View Digital Marketing Information” on page 45. For information about capabilities and SAS Digital Marketing, see Table 3.2 on page 54. For information about user permissions and SAS Digital Marketing, see “Assigning User Permissions for Sites That Use SAS Digital Marketing” on page 62.

For more information about SAS Digital Marketing administration, see *SAS Digital Marketing: Administrator’s Guide* at http://support.sas.com/documentation/solutions/ci/index.html. To access this page, log on as user ID *sas*, and use password *CIadmin123*.

SAS Customer Intelligence Integration Utilities

Use the SAS Customer Intelligence Integration Utilities to achieve the following objectives:

- extract, back up, and restore campaigns (including any associated communications and export definitions) to the SAS Metadata Repository
- import new SAS Customer Intelligence metadata into the SAS Metadata Repository

These utilities are also referred to as SAS Marketing Automation Integration Utilities. They are typically located on the SAS server tier and are outside the SAS Application Server.

SAS Customer Intelligence Integration Utilities were developed to import and extract metadata to and from the SAS Customer Intelligence. This metadata is used for reporting and for other uses by third-party software.

A typical user is a SAS consultant or third-party software integration specialist.

The Integration Utilities use the following files:

Executable

The typical location of the executable file is `C:\Program Files\SASHome\SASMarketingAutomationIntegrationUtilities\6.5`.

The processing logic for the sasmaextract and sasmaimport utilities is held within sas.analytics.crm.ma.auxdataio.jar, a single-client JAR file in `C:\Program Files\SASHome\SASVersionedJarRepository\eclipse\plugins`. The code within the JAR file acts as a client interface to the SAS Application Server.

Log

View the log file with a text editor (for example, Notepad under the Windows operating environment).

For information about the log files for SAS Customer Intelligence Integration utilities, see *SAS Customer Intelligence Integration Utilities: User's Guide*.

To use the SAS Customer Intelligence Integration Utilities to back up campaigns, see "Using SAS Integration Utilities to Back Up and Restore Campaigns" on page 208.

To convert XML files to another form (for example, a SAS data set) so that SAS reports can be created, use tools such as SAS XML Mapper or Extensible Stylesheet Language Transformations (XSLT). Consult the product documentation for the tool that you want to use.

Promotion is the process of moving data from a development environment to a testing environment, and then from a testing environment to the production environment. For details about SAS Promotion Tools, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

SAS Server Tier Components for SAS Digital Marketing

The following SAS components are installed on the SAS server tier for SAS Digital Marketing:

- SAS Digital Marketing
  - Base SAS
  - SAS/ACCESS engines
  - SAS Customer Intelligence Common Public Type Metadata
  - SAS Customer Intelligence Server
  - SAS Digital Marketing Server Components
  - SAS Digital Marketing Public Type Metadata
  - SAS/GRAPH
  - SAS LASR Analytic Server Access Tools
  - SAS Management Console
  - SAS/SECURE
  - SAS/SHARE
  - SAS Stored Process Server
  - SAS Studio
  - SAS Workspace Server
- SAS Visual Analytics Administration and Reporting Server
  - SAS/ACCESS engine
  - Base SAS
  - SAS LASR Analytic Server
  - SAS Visual Analytics Administration and Reporting
- SAS Metadata Server
  - Base SAS
  - SAS Metadata Server

SAS Server Tier Components for SAS Marketing Automation

The following SAS components are installed on the SAS server tier for SAS Marketing Automation:

- SAS Marketing Automation
  - Base SAS
  - Platform Suite for SAS (optional)
  - SAS/ACCESS engines
  - SAS/ACCESS to PC Files
  - SAS BI Lineage Plug-in for SAS Management Console
  - SAS BI Lineage Scheduling Server
  - SAS/CONNECT
  - SAS Customer Intelligence Common Public Type Metadata
  - SAS Customer Intelligence Processes

  - SAS Customer Intelligence Public Type Metadata
  - SAS Customer Intelligence Server
  - SAS/ETS
  - SAS/GRAPH
  - SAS LASR Analytic Server Access Tools
  - SAS Management Console
  - SAS Marketing Automation Integration Utilities
  - SAS Marketing Automation Launcher
  - SAS Metadata Bridges
  - SAS OLAP Server
  - SAS/OR
  - SAS/SECURE
  - SAS/SHARE
  - SAS/STAT
  - SAS Stored Process Server
  - SAS Studio
  - SAS Workspace Server
- SAS Visual Analytics Administration and Reporting Server
  - Base SAS
  - SAS/ACCESS engine
  - SAS LASR Analytic Server
- SAS Solution LASR Analytic Server
  - Base SAS
  - SAS Solution LASR Analytic Server
  - SAS Visual Analytics Administration and Reporting
  - Standard SAS/ACCESS engine of choice
- SAS Enterprise Miner Server
  - Base SAS
  - SAS Enterprise Miner Server
  - SAS/SECURE
  - SAS/STAT
  - SAS Studio
- SAS Metadata Server
  - Base SAS
  - SAS Metadata Server

SAS Server Tier Components for SAS Marketing Optimization

The following SAS components are installed on the SAS server tier for SAS Marketing Optimization:

- SAS Marketing Optimization
  - Advanced Computing Library
  - Base SAS
  - Platform Suite for SAS (optional)
  - SAS/ACCESS engines
  - SAS/ACCESS to PC Files
  - SAS BI Lineage Plug-in for SAS Management Console
  - SAS BI Lineage Scheduling Server
  - SAS/CONNECT
  - SAS Customer Intelligence Common Public Type Metadata
  - SAS Customer Intelligence Public Type Metadata
  - SAS Customer Intelligence Server
  - SAS/GRAPH
  - SAS High-Performance Server
  - SAS LASR Analytic Server Access Tools
  - SAS Management Console
  - SAS Marketing Optimization Launcher
  - SAS Marketing Optimization Procedure
  - SAS Marketing Optimization Public Type Metadata
  - SAS Marketing Optimization Server
  - SAS Metadata Bridges
  - SAS OLAP Server
  - SAS/OR
  - SAS/SECURE
  - SAS/SHARE
  - SAS/STAT
  - SAS Studio
  - SAS Threaded Kernel Extensions for Advanced Analytics
  - SAS Threaded Kernel Extensions for High-Performance Analytics
  - SAS Threaded Kernel Extensions for Marketing Optimization
  - SAS Threaded Kernel Extensions for Operations Research
  - SAS Threaded Kernel Extensions for Statistics
  - SAS Thread-Safe Compiler and Symbolic Differentiator

  - SAS Workspace Server
- SAS Visual Analytics Administration and Reporting Server
  - Base SAS
  - SAS/ACCESS engine
  - SAS LASR Analytic Server
- SAS Solution LASR Analytic Server
  - Base SAS
  - SAS Solution LASR Analytic Server
  - SAS Visual Analytics Administration and Reporting
  - Standard SAS/ACCESS engine of choice
- SAS Metadata Server
  - Base SAS
  - SAS Metadata Server

SAS Server Tier Components for SAS Real-Time Decision Manager

The following SAS components are installed on the SAS server tier for SAS Real-Time Decision Manager:

- SAS Real-Time Decision Manager
  - Base SAS
  - SAS/ACCESS engines
  - SAS BI Lineage Plug-in for SAS Management Console
  - SAS BI Lineage Scheduling Server
  - SAS Customer Intelligence Common Public Type Metadata
  - SAS Customer Intelligence Processes
  - SAS Customer Intelligence Public Type Metadata
  - SAS Customer Intelligence Server
  - SAS Data Integration Server
  - SAS/GRAPH
  - SAS LASR Analytic Server Access Tools
  - SAS Management Console
  - SAS Marketing Automation Launcher
  - SAS/SECURE
  - SAS/STAT
  - SAS Pooled Workspace Server
  - SAS Stored Process Server
  - SAS Studio
  - SAS Workspace Server

- SAS Decision Services Server
  - Base SAS
  - SAS/ACCESS engine
  - SAS Decision Services Server Components
  - SAS Environment Manager Agent
  - SAS Integration Technologies
  - SAS Object Spawner
  - SAS ODBC Driver for the Web Infrastructure Platform Data Server
  - SAS Pooled Workspace Server
  - SAS/SECURE
  - SAS Stored Process Server
  - SAS Studio
  - SAS Web Infrastructure Platform Data Server
  - SAS Web Infrastructure Platform Scheduling Services
  - SAS Workspace Server
- SAS Business Rules Manager Server
  - Base SAS
  - SAS/SECURE
  - SAS STUDIO
  - SAS/ACCESS engine
  - SAS Business Rules Manager
  - SAS LASR Analytic Server Access Tools
- SAS Federation Server
  - Base SAS
  - SAS/SECURE
  - SAS Studio
  - SAS Federation Server
  - SAS LASR Analytic Server Access Tools
  - SAS Federation Server Driver
- SAS Authentication Server
  - Base SAS
  - SAS/SECURE
  - SAS Studio
  - SAS Authentication Server
  - SAS LASR Analytic Server Access Tools
  - SAS Federation Server Driver
- SAS Visual Analytics Administration and Reporting Server

  - Base SAS
  - SAS/ACCESS engine
  - SAS LASR Analytic Server
- SAS Solution LASR Analytic Server
  - Base SAS
  - SAS Solution LASR Analytic Server
  - SAS Visual Analytics Administration and Reporting
  - Standard SAS/ACCESS engine of choice
- SAS Metadata Server
  - Base SAS
  - SAS Metadata Server

Data Tier

Overview of the Data Tier

Campaign managers use SAS Information Map Studio to build information maps. The maps identify the data sources (represented on the data tier) that campaign managers use to formulate their campaign queries. Campaign managers also use SAS Customer Intelligence Studio to choose their target markets from the data sources on the data tier. History tables, response tables, and the results of marketing campaigns might also be stored on your data tier, depending on your environment.

The database that is selected depends on the solution and database version as well as other factors. If all SAS Customer Intelligence solutions are deployed, the number of available choices are limited.

Data Sources

Customer data is typically stored in a supported relational database management system (RDMS). It is likely that this data already exists and is available before a deployment. The second key data source, the SAS Customer Intelligence common data model, is also stored in a relational database. This might be in the same database system or in a system provided by a separate provider. To prevent cross-database joins and performance problems, store the common data model in the same database as the customer data. The common data model is created after the software deployment. Initially, it contains no data. If the data provider is a SQL Server database, the hardware must support the Windows operating system.

Components of the Data Tier

Your data tier might include several sources of data for SAS Customer Intelligence processes. The sources of data are determined by the objectives of your organization and by company resources.

You can access almost any type of data source by using SAS LIBNAME statements.

Note: Some databases, such as PostgreSQL, perform a case-sensitive comparison of table or column names when the names are in quotation marks. For these databases, set the following options to `No`:

- `Preserve the column name as in the database management system`
- `Preserve the table name as in the database management system`

SAS can access SAS data sets, which are analogous to relational database tables.

SAS/ACCESS provides direct access to the following relational databases for SAS Marketing Automation:

- DB2
- Microsoft SQL Server on UNIX through SAS/ACCESS for SQL Server
- Microsoft SQL Server on Windows through SAS/ACCESS for ODBC or SAS/ACCESS for OLEDB
- Oracle
- PostgreSQL
- SAP HANA
- Teradata

SAS/ACCESS provides access to the following data warehouse appliances:

- Greenplum

  Note: To avoid errors in processing Greenplum tables, set the following Greenplum server configuration value:

  ```
  SET standard_conforming_strings = 'ON'
  ```
- Netezza

SAS/ACCESS provides access to Amazon Redshift through PostgresSQL and SAS/ACCESS for ODBC. For more information, see Step into the Cloud: Ways to Connect to Amazon Redshift with SAS/ACCESS and *SAS/ACCESS for Relational Databases: Reference* at http://support.sas.com/documentation/onlinedoc/access/index.html.

For the specific versions of each database that is supported, see your on-site SAS support personnel. For information about SAS data sets and SAS/ACCESS, see *SAS Intelligence Platform: Data Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Limit the Size of PostgreSQL Columns

There is no limit on the length of columns in PostgreSQL tables that are created from SAS processes. If you want to limit the column length, use the following guidelines:

- If the session encoding is UTF-8 and single- and multiple-byte characters are used, specify `DBCLIENT_MAX_BYTES=3`.

- If the session encoding is UTF-8 and only single-byte characters are used, specify `DBCLIENT_MAX_BYTES=1`.
- If the session encoding is LATIN1, specify `DBCLIENT_MAX_BYTES=1` or do not specify the setting.

For more information, see *SAS/ACCESS for Relational Databases* at http://support.sas.com/documentation/onlinedoc/access/index.html.

Registering Data for SAS Customer Intelligence

What Data Sources Do I Need to Register?

The data sources that must be registered to the SAS Metadata Repository for access by SAS Customer Intelligence include the following tables:

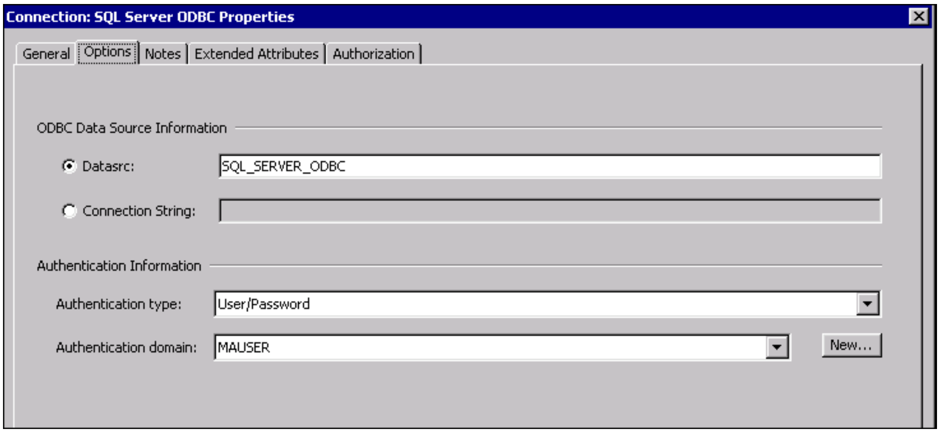
- any database source for SAS Customer Intelligence data (such as Oracle or Teradata) that is referenced by a SAS Information Map
- the common data model tables
- holding tables for SAS Marketing Automation or SAS Real-Time Decision Manager
- (optional) libraries to contain exported data, lists, and custom details. These libraries are specified in the business context

Mapping SQL Server to Multiple Databases

If a user ID in SQL Server is mapped to more than one database, specify the name of the initial database as the default database in the data source definition. To specify the name of the initial database:

1. In SAS Management Console, select the SQL server name from the Server Manager plug-in.
2. Right-click the connection name and select **Properties**.
3. On the **Options** tab, specify the name of the data source.

   Figure 1.1 *Specify Data Source Name*



Connection: SQL Server ODBC Properties

ODBC Data Source Information

● Datasrc: SQL_SERVER_ODBC

○ Connection String:

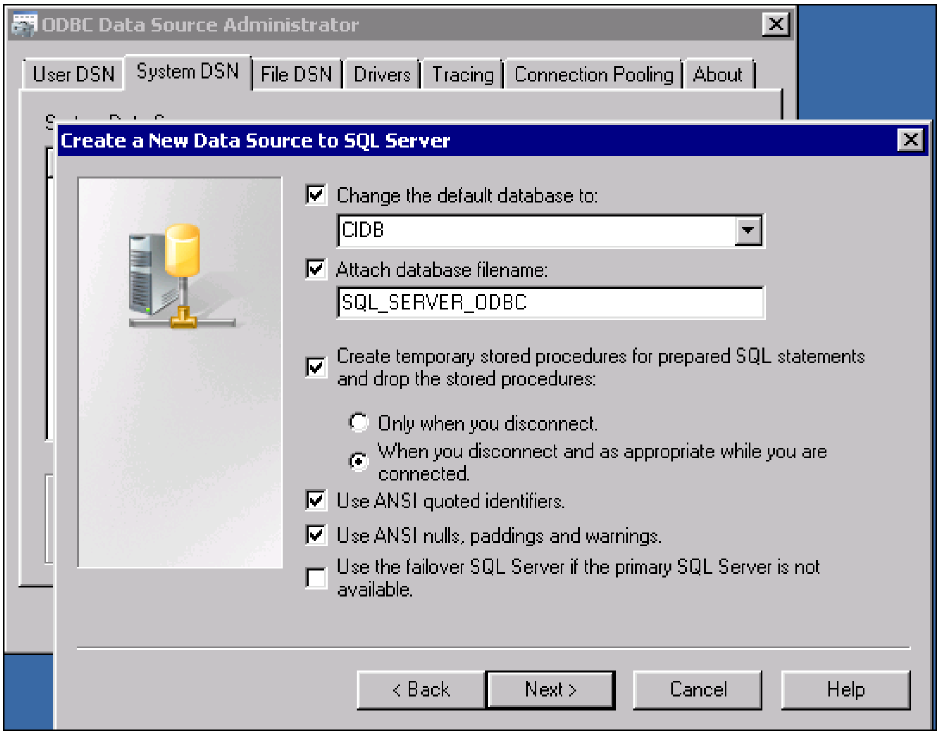
Authentication Information

Authentication type: User/Password

Authentication domain: MAUSER

4. In the ODBC Data Source Administrator, specify the name of the default database.

Figure 1.2 *Specify Default Database Name*



☑ Change the default database to:

CIDB

☑ Attach database filename:

SQL_SERVER_ODBC

☑ Create temporary stored procedures for prepared SQL statements and drop the stored procedures:

○ Only when you disconnect.

● When you disconnect and as appropriate while you are connected.

☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

□ Use the failover SQL Server if the primary SQL Server is not available.

SAS Library Resources

Overview of SAS Library Resources

SAS Customer Intelligence requires several SAS libraries that contain SAS metadata, holding tables, and operational data. Most of the required libraries are defined when the product is installed. The following library locations are specified on the **Settings** tab of the business context.

- Selection MATables library
- Selection report exports library
- Reporting libref

The metadata library is specified on the **Metadata** tab of the business context.

Verify that all SAS Customer Intelligence users, including saswbadm, have both Read and Write access to the physical location of the library.

You can specify the location of a libref in any preferred physical path. The physical path is represented in the following sections as <*plan-area*>.

MATables Library

The MATables libref specifies the location of the holding tables that are created by various stored processes. The MATables libref also specifies the location of the holding tables for metadata that is generated by SAS Customer Intelligence when a campaign is created. These holding tables are automatically deleted when they are no longer needed.

Here is an example of the LIBNAME statement that defines the MATables library:

LIBNAME MATables <*plan-area*> ;

This is the default location of the MATables library:<*config*>`Lev1/Applications/SASCustomerIntelligence/CampaignManagement/data/MATables`.

The libref for these holding tables must be MATables.

MAMisc Library

Here is an example of the LIBNAME statement that defines the MAMisc library:

LIBNAME MAMisc <*plan-area*>;

This is the location of the MAMisc library: `<config>/Lev1/Applications/SASCustomerIntelligence/CustomerIntelligenceCommon/Data/MAMisc`.

The MAMisc libref specifies the location of various tables that contain information such as sequence numbers and task lists. It also specifies the location of the defined SAS formats that are used by SAS data sets.

MAMOLib Libref

The MAMOLib libref specifies an area for temporary data storage, including tables that are generated during priority-only optimization. The library is listed as SASApp - MAMORoot in the Libraries folder of the Data Library Manager plug-in.

This is the location of the library: `<config>/Lev1/Applications/SASCustomerIntelligence/CampaignManagement/Data/MAMORoot`.

Defining Library References

Overview

Define the required libraries for accessing SAS Customer Intelligence data by using the Data Library Manager in SAS Management Console. For more information, see *SAS Intelligence Platform: Data Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Note: Although you can specify some library references (librefs) by editing the autoexec_usermods.sas file, these libref values are used only if no values for librefs have been specified in SAS Management Console. For more information, see “The autoexec_usermods.sas File” on page 31.

Making Table and SAS Export Librefs Available to Export Definitions

The mausrexp.sas Macro

You can use the mausrexp.sas macro to define your export libraries. You do not need to restart the Object Spawner in order to make these libraries available after you make changes to the macro. The changes are also not overwritten when you apply hot fixes. This method has the advantage of not disrupting every spawned SAS session.

On Windows, the mausrexp.sas file is normally found under **`C:\Program Files\SASHome\SASFoundation\9.4\ma\sasmacro`**.

On UNIX, the mausrexp.sas file is normally found under **`SASHome/SASFoundation/9.4/sasautos`**.

Use the following section within mausrexp.sas to list your export libraries.

```
/* BEGIN INTENDED USER MODIFICATION SECTION */
   /* Create Macros defined export libraries here */
   /* Example: %let expsas=%str(libname expsas 'some_path'); */

   /* END INTENDED USER MODIFICATION SECTION */
```

Here is an example entry for an Oracle export library.

```
/* BEGIN INTENDED USER MODIFICATION SECTION */
   /* Create Macros defined export libraries here */
   /* Example: %let expsas=%str(libname expsas 'some_path';); */
   %let expora=%str(LIBNAME EXPORA ORACLE  PATH=ORACLE.WORLD SCHEMA=MASCHEMA
USER=userid PASSWORD="{SAS002}xxxxxxxxxxxxxx");

   /* END INTENDED USER MODIFICATION SECTION */
```

Here is an example for a SAS data set export.

```
%let SASEXP=%str(LIBNAME SASEXP '<path to writeable location on SAS server tier>');
```

To ensure that your customizations to the macro are not overwritten in the future, place a copy of the updated mausrexp.sas macro into an automacro location. This ensures that the updated version is always referenced, rather than the default version.

The autoexec_usermods.sas File

The autoexec_usermods.sas file specifies some options in the SAS environment for all SAS sessions that are created by the object spawner and used by SAS Customer Intelligence. SAS sessions include workspace server sessions and calls to the SAS Stored Process Server. This file can be edited to specify the following options:

- SAS Marketing Automation libref for campaign export files. Although you normally use the SAS Management Console to register libraries for SAS Marketing Automation, you should specify the libref for campaign export files in autoexec_usermods.sas. See also Chapter 10, “Scheduling SAS Marketing Automation Campaigns,” on page 169 for more information about the export files that are used in SAS Marketing Automation.
- SAS options that specify the search order for locating library resources.

The library reference values that are specified in autoexec_usermods.sas are used only if the libraries have not been defined in SAS Management Console.

By default, autoexec_usermods.sas is installed in the configuration directory for the main SAS server. The configuration directory is typically located at **`.../Lev1/<`*`SASApp`*`>/WorkspaceServer`** , where *SASApp* is the default context server.

Note:

Several SAS system options designate the search order that is used to locate SAS library resources. You can specify the statements anywhere within autoexec_usermods.sas.

```
FMTSEARCH=(MAMisc SecondLib ThirdLib) NOFMTERR;
```

FMTSEARCH specifies the search order for the format catalog or catalogs. The previous example specifies the search to begin in the MAMisc library, then to continue in a library called SecondLib, and finally to end in a library called ThirdLib.

NOFMTERR specifies to issue a note and continue processing if the specified variable format cannot be located.

For details about these SAS system options, see *SAS System Options: Reference* at http://support.sas.com/documentation/onlinedoc/base/index.html.

2

Administration

Administrative Objectives

Administrators set up and maintain the SAS Customer Intelligence Studio environment and the supporting SAS Intelligence Platform environment.

For information about integrating SAS products with other applications in your enterprise, see *SAS Integration Technologies* at http://support.sas.com/documentation/onlinedoc/inttech.

Tasks

After you complete the setup tasks by following the instructions that are provided in the SAS Deployment Wizard User's Guide, you might be required to perform additional tasks. These tasks might include setting options for environmental components that include third-party application servers, SAS servers, and scripts to support campaign activities by users.

1 Control user access to business contexts, campaigns, and reports. For more information, see Chapter 3, "Security," on page 47.

2 Create information maps for campaign activities. For more information, see Chapter 5, "SAS Information Maps for Campaigns," on page 73.You can also modify the provided information map for SAS Customer Intelligence.

3 Configure the Reports workspace to display data from the common data model. For more information, see Chapter 8, "Displaying Reports in the Reports Workspace," on page 143.

4 Manage the scheduling tasks for SAS Marketing Automation. See *Scheduling in SAS* at http://support.sas.com/documentation/onlinedoc/sasmc/index.html for details about how to administer scheduling tools. These tools enable you to automate the scheduling and execution of SAS jobs in general, including jobs from SAS Marketing Automation. See Chapter 10, "Scheduling SAS Marketing Automation Campaigns," on page 169 for details about configuring the components that are specifically used for SAS Marketing Automation.

5 Extract and import campaigns, and other objects as needed, by using either Integration Utilities for SAS Customer Intelligence, or SAS promotion tools. For more information about SAS promotion tools, see "Using the Promotion Tools" in SAS Intelligence Platform: System Administration Guide. See also the *SAS Customer Intelligence Integration Utilities: User's Guide* at http://support.sas.com/documentation/solutions/ci/index.html. To access this page, log on as user ID *sas*, and use password *CIadmin123*.

6 Update the counts metadata as needed. For more information, see "Understanding the Counts Metadata" on page 102.

7 Optimize the SAS environment by customizing the autoexec_usermods.sas file. For more information, see "The autoexec_usermods.sas File" on page 31.

8 Define a strategy for backing up and restoring information maps and campaigns. For more information, see Chapter 12, "Backing Up and Restoring Data," on page 205.

9 Troubleshoot problems by setting logging levels. For more information, see Chapter 14, "Troubleshooting," on page 221.

Setting Up

Introduction

See your on-site SAS support personnel for installation instructions.

To view the setup instructions for the common SAS Intelligence Platform, see the *SAS Intelligence Platform: Installation and Configuration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Use a Single Install Account

Problems often occur when different accounts are used for various installations and subsequent installations. To eliminate potential problems, always use the same account (such as the SAS install account that was specified on the pre-installation checklist) for all installations. This practice is particularly important when applying service packs. If the user is running Windows Terminal Services, then you should operate in install mode.

Use Fully Qualified Server Names

Specify fully qualified server machine names in the SAS Deployment Wizard of SAS Management Console in order to avoid problems that can result in connecting to the workspace and stored process servers.

Table 2.1 *Example of a Fully Qualified Server Name*

Incorrect	joesmachine
Correct (Fully Qualified)	joesmachine.unx.mybusiness.com

Edit and correct any server machine names that are not fully qualified by modifying the server's properties in the `Server Manager` folder in SAS Management Console.

Specify Date Format for Exported Files

Instead of using the default date format for exported files, you can set a configuration option to make the date format consistent with the current locale.

To set the date format:

1 Open SAS Management Console.

2 Select the **Plug-ins** tab.

3 Expand the `Application Management` folder.

4 Expand the **Configuration Manager** plug-in.

5 Right-click the **Customer Intelligence Core 6.5** plug-in and select **Properties**.

6 In the Customer Intelligence Core 6.5Properties dialog box, select the **Advanced** tab.

7 For the ExportDateFormat property, specify `nldate.`. Include the period (.) as part of the date format.

Enable Email Notifications

To enable email notifications:

1 Open SAS Management Console.

2 On the **Plug-ins** tab, expand the `Application Management` folder.

3 Expand the Configuration Manager plug-in.

4 Right-click the SAS Application Infrastructure plug-in and select **Properties**.

5 On the **Settings** tab of the SAS Application Infrastructure Properties dialog box, select **General Configuration**.

6 In the **Alert notifications type** field, select **Email** and **Portal** and add them to the **Selected** list.

7 On the **Advanced** tab of the SAS Application Infrastructure Properties dialog box, enter values for **Email.Host** and **Email.Port**.

In a standard installation of SAS middle-tier components, the configuration of the Java Mail Session is typically automated using prompted values that are provided by the installer.

Support Double-Byte Character Sets

Text in many Asian languages requires the support of double-byte character sets. In order to display this text properly in SAS Customer Intelligence tables, an option must be set in the SAS configuration file.

Specify the following option in the sasv9.cfg file:

```
-VALIDVARNAME ANY
```

By default, the location for sasv9.cfg is `C:\Program Files\SASHome\SASFoundation\9.4\sasv9.cfg`.

Set Session Time-out Value

By default, the HTTP session time-out value for SAS Customer Intelligence is seven days.

To change the session time-out value, edit these files. The path on your system is determined by the SAS Web Application Server.

1 In `<`*`SAS-configuration-directory`*`>/Lev`*`n`*`/Web/WebAppServer/SASServer6_1/sas_webapps/sas.customerintelligencestudio.war/config.xml`, edit the session time-out value in the following code:

```
<timeout>
   <enabled>true</enabled>
    <!-- Default to 7 days. Must match timeout in web.xml -->
   <maxInactiveInterval>604800</maxInactiveInterval>
 </timeout>
```

2. In **<SASConfigDir>/Lev*n*/Web/WebAppServer/SASServer6_1/sas_webapps/sas.customerintelligencestudio.war/WEB-INF/web.xml**, edit the following code:

```
<session-config>
   <!-- default is 7 days -->
  <session-timeout>10080</session-timeout>
   <!-- this deploys fine into TCServer -->
  <cookie-config>
   <path>/</path>
   </cookie-config>
 </session-config>
```

The time-out values in both files must be equivalent.

Note: The settings for time-out values are overwritten when you apply a SAS Customer Intelligence hotfix and redeploy the SAS Customer Intelligence Studio web application.

Prevent Users from Adding Attachments

You can prevent users from adding images and attachments to campaigns. To disable attachments, set the following property in the config.xml file:

```
<turnAttachmentsOff>true</turnAttachmentsOff>
```

Attachments and images that were added previously to diagrams, the Attachments page, and the Properties page are still displayed. The Comments page does not display attachments that were added previously. Users are not able to add any new attachments or images.

Create Report Exports in SAS Web Report Studio

You use the Report Link node in SAS Customer Intelligence Studio to select an export from a SAS Web Report Studio report. Before you can select an export from the Report Link node, you must create the export in SAS Web Report Studio.

Set Up SAS Information Map Studio

1. Open SAS Information Map Studio and select the information map that is used in the SAS Web Report Studio report.

2. Enable custom properties in SAS Information Map Studio by selecting **Tools** ► **Options**. On the **Advanced** tab of the Options window, select **Custom properties at start-up**.

3. Add the `CIBusinessContext` extended attribute to the Custom Properties section of the information map properties. The attribute value is the name of the business context that is associated with the campaign that contains the Report Link node.

4. Make sure that the subjects in this information map are compatible with the subjects in the information map that is used by the business context for the campaign that contains the Report Link node.

For more information, see "All Custom Properties for Information Maps" on page 91, Table 11.1 on page 181, *SAS Marketing Automation: User's Guide*, and the online Help for SAS Information Map Studio.

Set Up SAS Management Console

1. In the Data Library Manager in SAS Management Console, verify that the SASApp -MAStatic library exists, and that it has these properties:

   Selected Server
   SASApp.

   Libref
   MAStatic. This is the required value.

   Engine
   BASE.

   Path Specification
   a valid pathname.

2. In the User Manager plug-in in SAS Management Console, verify that the user of the information map in SAS Web Report Studio is a member of the Advanced role in SAS Web Report Studio.

Set Up SAS Web Report Studio

1. In SAS Web Report Studio, create a list report that is based on the information map that you designated in "Set Up SAS Information Map Studio". When you select a data source for the report, include at least one data item that is mapped as subject ID in the information map.

2. Do not assign subject IDs to any report variables. Subject ID data items should be hidden. The report should not have subject ID columns as data items. To hide data items, right-click the report in Edit mode and select **Assign Data**. Select all of the columns that are defined as subject IDs in the information map. Click **Move Items** and select **Move to Hidden.**

3. In Edit mode, right-click the report and select **Report Linking**. Select **Include report links**. Select **Another SAS application** as the link destination. Select **SAS Customer Intelligence/QueryTransferService** as SAS applications.

4. In View mode, right-click a link in the report to display the SAS Customer Intelligence Integration window. Select a folder name for the report export.

   To create a static report export that is based on selected data, select **Export the data that represents this selection**.

   To create a dynamic report export that is based on criteria for selected data, select **Export the criteria that is produced by this selection**.

For more information, see *SAS Web Report Studio: User's Guide* at http://support.sas.com/documentation/onlinedoc/wrs/.

Set Up SAS Customer Intelligence Studio

1 In SAS Customer Intelligence Studio, open the business context that is defined by the CIBusinessContext extended attribute in the information map. The extended attribute was defined in "Set Up SAS Information Map Studio" on page 37. In the Business Context Properties window, select the **Options** tab. Set the value of **Selection report exports library** to the folder that is defined in SAS Management Console in "Set Up SAS Management Console" on page 38.

2 Log out and then log in to SAS Customer Intelligence Studio.

3 Add a Report Link node to the diagram for a selection campaign.

4 Double-click the Report Link node and select the report export that you created in "Set Up SAS Web Report Studio" on page 38.

5 After you close the node and save your changes, the Report Link node is linked to a SAS Web Report Studio report.

Middle-Tier Clustering

A cluster of middle-tier application servers provides hardware and software fault tolerance for SAS Marketing Automation. The hardware fault tolerance is the result of clustering multiple physical middle-tier servers. A single hardware fault does not cause SAS Customer Intelligence Studio applications to become unavailable. The software fault tolerance is the result of the ability to run one or more application server instances on each physical server. A single software fault does not cause SAS Customer Intelligence Studio applications to become unavailable. Hardware and software fault tolerance can be combined in a single installation.

A load balancer distributes user sessions across the cluster of servers, or nodes. When a server is unavailable, the load balancer routes new user sessions to other nodes in the cluster. Each user session is associated with a single node in the cluster. When that node fails, the user session is no longer functional. When the user closes the browser and logs back on to SAS Customer Intelligence Studio, the load balancer routes the new session to another node in the cluster.

CAUTION! A new browser session must be opened when a node is in Drain mode. If you select the **Drain** option for a cluster node, notify the users of that node. In addition to logging off from SAS Customer Intelligence Studio, the users must start a new HTTP session. A new HTTP session be started either by selecting a new session from the browser, or by closing and opening the browser. If the user logs off and back on without starting new HTTP session, their SAS Customer Intelligence Studio session continues on the node that is in Drain mode.

If all of the nodes in the cluster fail, SAS Customer Intelligence Studio applications are no longer available.

If a user is working in SAS Customer Intelligence Studio, and operations start to fail, the node that is hosting the session might be unavailable. The user should notify the administrator, and then log off and log back on to create a new session. Unsaved changes are lost.

For information about middle-tier application server clustering, see *SAS Intelligence Platform: Middle-Tier Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

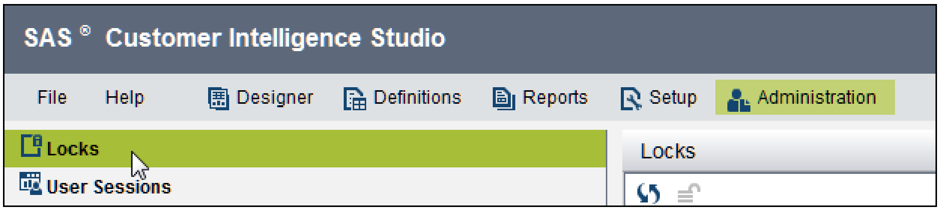
Metadata server clustering is also supported. For information about metadata server clustering, see *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

The Locks page and the User Sessions page in the Administration workspace in SAS Customer Intelligence Studio list the cluster node that hosts each object or user session.

Release Locked Objects

In some circumstances, you might not be able to edit an object such as a campaign even if you have Edit permission. To unlock an object so that you can edit it, select the Locks category in the Administration workspace.

Figure 2.1 *Locks*



Only those objects that you have permission to view or edit are displayed. You must have Edit permission in order to release a locked object.

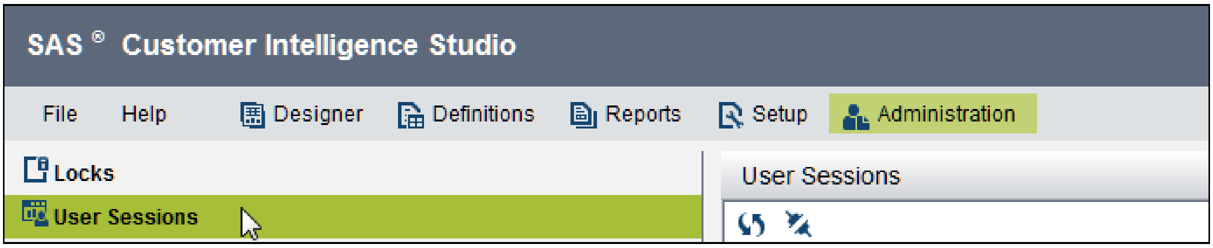
Select the object that you want to release and click 🔓 to be able to edit the object. After an object is released, any changes to the object can be saved only by selecting **Save as**.

Manage User Sessions

Managing User Sessions in the Administration Workspace

As an administrator, you might want to manage active sessions in SAS Customer Intelligence Studio. For example, you might need to log off from all user sessions in order to provide system maintenance.

To manage user sessions, select the User Sessions category in the Administration workspace.



To log off from a user session, select the session and click ✖.

If a campaign is closed while it is executing, the execution continues to run. A best practice is to set up notifications so that users receive an email message if the execution fails. For more information, see *SAS Marketing Automation: User's Guide*.

Managing User Sessions If Secure Sockets Layer (SSL) Is Configured

If SAS Customer Intelligence is configured to use Secure Sockets Layer (SSL) at your site, you cannot log off users through the Administration workspace unless you modify the SAS Web Application Server parameters. To modify the parameters:

1. Open the file that is used to start the SAS Web Application Server.

   On UNIX or Linux, the file is `<`***`SAS-configuration-directory`***`>/Lev1/Web/WebAppServer/SASServer6_1/conf/setenv.sh`.

   On Windows, the file is `<`***`SAS-configuration-directory`***`>\Lev1\Web\WebAppServer\SASServer6_1\conf\wrapper.conf`

2. Add the following parameters to the file:

   ```
   -Dsas.ci.cluster.protocol=https
   -Dsas.ci.cluster.hostname=load-balancer-hostname
   -Dsas.auto.publish.port=443
   ```

   If the environment is not configured to use the default port, use the HTTPS port number.

3. Restart the SAS Web Application Server.

Update Dynamic Custom Detail Values

You can choose whether to update the values for dynamic custom details in a campaign when the values are changed in treatments.

Add the option to the following file on every SASServer6 node where SAS Marketing Automation is deployed.

On Windows, the file is **`LevConfig\Web\WebAppServer\SASServer6_cluster_number\conf\wrapper.conf`**.

On UNIX, the file is **`LevConfig/Web/WebAppServer/SASServer6_cluster_number/bin/setenv.sh`**.

By default, dynamic custom details are updated only in a campaign when the custom detail names are not the same in the old and new version of the campaign. To specify that dynamic custom detail values are updated when the values change in a treatment, enter this option:

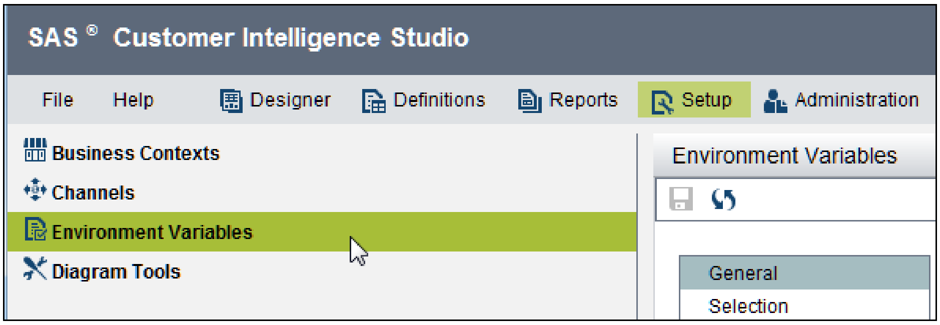
```
-Dcom.sas.crm.overwrite_all_custom_details_on_treatment_update.enabled=true
```

Environment Variables

Overview of Environment Variables

The Environment Variables category in the Setup workspace enables you to modify environment settings such as log size.

Figure 2.2 Environment Variables



Note: If more than one administrator is editing environment variables at the same time, the last saved edits override previous edits.

Set Row and Item Recovery Options

Row options are set on the General page.

Figure 2.3 General Page

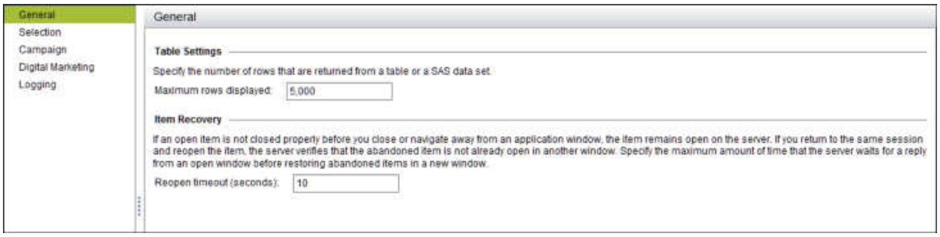


Table Settings

Specify the number of rows that are returned from a table or a SAS data set.

Maximum rows displayed: 5,000

Item Recovery

If an open item is not closed properly before you close or navigate away from an application window, the item remains open on the server. If you return to the same session and reopen the item, the server verifies that the abandoned item is not already open in another window. Specify the maximum amount of time that the server waits for a reply from an open window before restoring abandoned items in a new window.

Reopen timeout (seconds): 10

Enter a number in the **Maximum rows displayed** field to specify the maximum number of rows that are returned from a table or a SAS data set. This setting affects external tables that contain the following items.

- External treatments

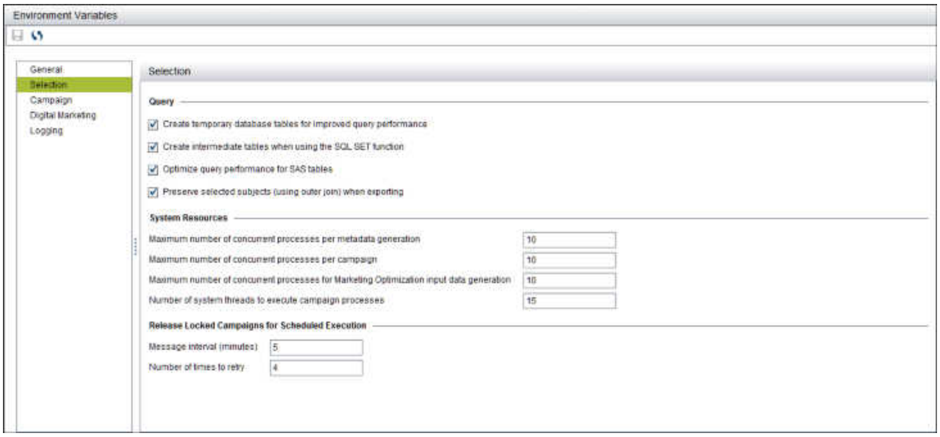
If an open item is not closed properly before you close or navigate away from an application window, the item remains open on the server. If you return to the same session and attempt to reopen the item, the server verifies that the abandoned item is not already open in another window. Enter the number of seconds in the **Reopen timeout** field to specify the maximum amount of time that the server waits for a reply from an open window before restoring abandoned items in a new window.

Selection Campaign Settings

Overview of Selection Campaign Settings

Selection campaign options are set on the Selection page.

Display 2.1 *Selection Page*



Selection

Query

☑ Create temporary database tables for improved query performance

☑ Create intermediate tables when using the SQL SET function

☑ Optimize query performance for SAS tables

☑ Preserve selected subjects (using outer join) when exporting

System Resources

Maximum number of concurrent processes per metadata generation	10
Maximum number of concurrent processes per campaign	10
Maximum number of concurrent processes for Marketing Optimization input data generation	10
Number of system threads to execute campaign processes	15

Release Locked Campaigns for Scheduled Execution

Message interval (minutes)	5
Number of times to retry	4

Create Temporary Database Tables

You can send the contents of cells to the database that is being used by SAS Customer Intelligence Studio so that the query can be processed by the database. Select **Create temporary database tables for improved query performance**. When the query has finished running, the query results are passed back to SAS Customer Intelligence Studio.

Create Intermediate SQL Tables

To improve application performance by creating temporary tables when the SQL SET function is used, select **Create intermediate tables when using the SQL SET function**.

Optimize Query Performance

To optimize query performance in a campaign that contains SAS tables, select **Optimize query performance for SAS tables**.

During query performance optimization, you might update the count in a node that is preceded by a node that creates SAS tables. The section of the process flow between that node and the node that is updating counts is optimized. Examples of nodes that create SAS tables include the Cell and Link nodes, as well as any stored process node (Cluster, Split, or Limit).

Query performance optimization will not take place if you update counts from an Export node or a Report node. These nodes define output variables that might directly reference database tables.

Preserve Rows By Using Outer Joins

To preserve rows so that records are not lost because of an inner join in the information map, select **Preserve selected subjects (using outer join) when exporting**.

For example, a CUSTOMER table of customer names and a CHECKING table of checking accounts might have an inner join in the information map. The inner join results contain only those customers who have checking accounts. A query that requested all the customers who do not have a closed checking account would result in a table of all customers who have an open checking account. Customers who do not have checking accounts would be excluded. If you select **Preserve selected subjects (using outer join)**, customers who do not have checking accounts would be included in your query results.

Set Process Limit

Specify the number of concurrent processes per metadata generation by entering or selecting a number for **Maximum number of concurrent processes per metadata generation**.

Specify the Number of Tasks Assigned to Query Processing

To specify the number of tasks that are assigned to query processing when counts are updated for a Cell node, enter or select a number for **Specify maximum number of concurrent processes per campaign**.

Set Number of System Threads

To control the maximum number of executing stored processes and queries in the system across all executing campaigns, enter a number for **Number of system threads to execute campaign processes**. Changing the setting immediately changes system behavior. An increase in the number means that system will start executing any ready tasks. A decrease in the number will shut down threads. If the system is currently executing at capacity, then threads will shut down as their tasks complete.

Release Locked Campaigns for Scheduled Execution

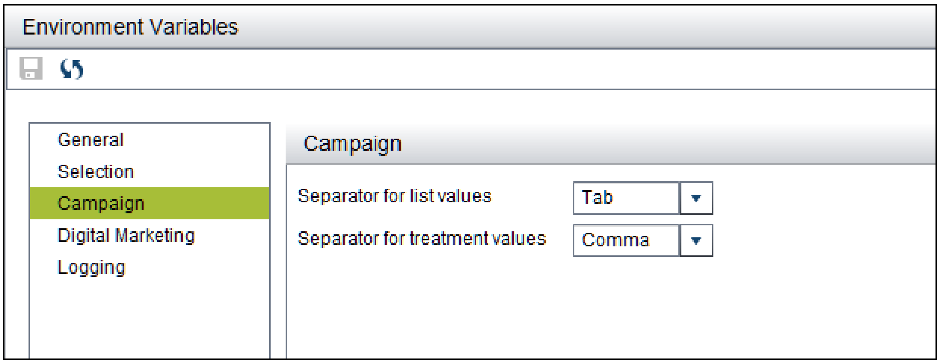
The Launcher might attempt to open a campaign for edit while the campaign is already open in a SAS Customer Intelligence Studio session. Warning messages are sent to the SAS Customer Intelligence Studio session advising the user that the lock on the campaign will be released. The frequency of the messages is controlled by the **Message interval (minutes)** setting. For example, the message could be sent every five minutes.

To set the number of times the message should be sent before the lock is released, select a value from **Number of times to retry**.

Set Separators for Lists and Treatment Values

The separators for list and treatment values are displayed on the Campaign page.

Figure 2.4 Campaign Page



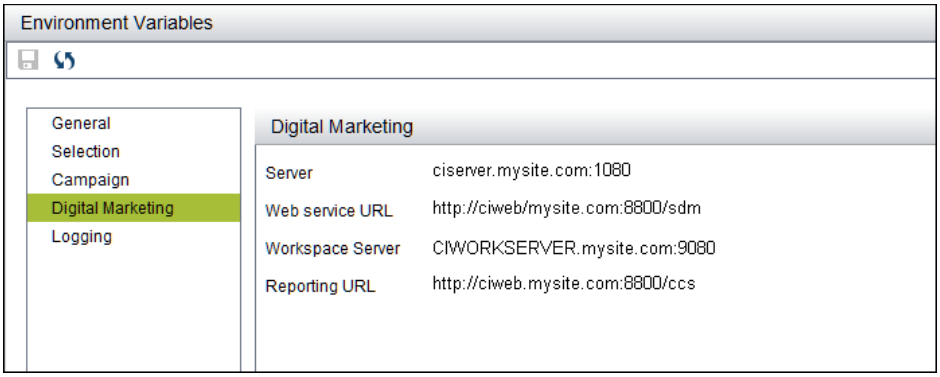
Separator for list values	Tab
Separator for treatment values	Comma

To set the delimiter for lists and treatment values that are exported to external files, enter or select a separator. When you export more than one treatment for a customer, you can set the delimiter for a list of treatments.

View Digital Marketing Information

If SAS Digital Marketing is installed at your site, the Digital Marketing page displays the server name and other information about the installation.

Display 2.2 Digital Marketing Page

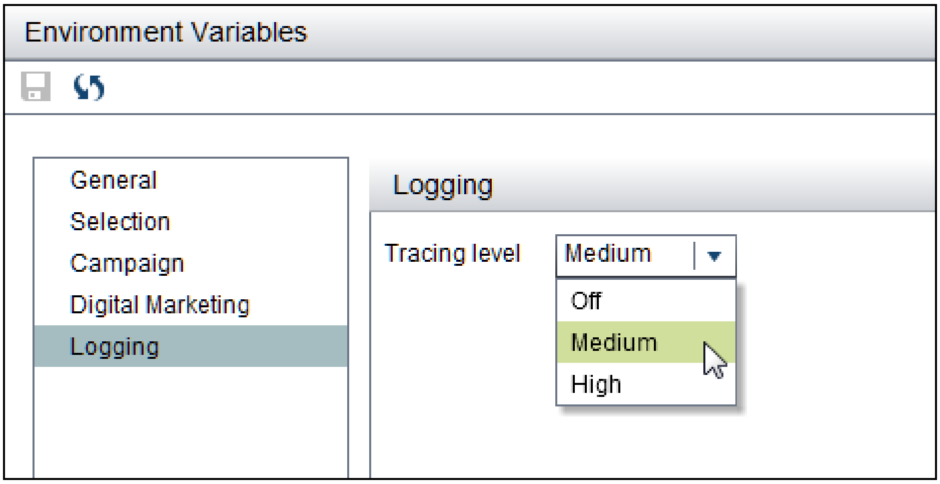


Server	ciserver.mysite.com:1080
Web service URL	http://ciweb/mysite.com:8800/sdm
Workspace Server	CIWORKSERVER.mysite.com:9080
Reporting URL	http://ciweb.mysite.com:8800/ccs

Set Logging Level

The trace level for logging is set on the Logging page.

Figure 2.5 Logging Page


Environment Variables

Logging

Tracing level: Medium

Select a tracing level to set the amount of detail in log messages.

Off
suppresses all logs except those logs that have a setting of DEBUGLEVEL_ALWAYS.

Medium
produces logs with a typical level of detail.

High
produces logs with a very high level of detail.

This option controls the size of the log that is generated for any SAS code that is run during a SAS Customer Intelligence Studio session.

3

Security

Overview of Security Administration

The security model for SAS Customer Intelligence and the SAS Intelligence Platform provides the following features:

- secure access to data and metadata

- role-based access to application features
- confidential transmission and storage of data
- logging and auditing of security events
- access control reporting

Security administration consists of the following tasks:

- administering SAS identities for your users by adding account information to the SAS Metadata Server
- administering groups of users in order to simplify the management of roles and permissions
- administering roles, which provide users with access to specific application features
- administering users' permissions to access metadata repositories, folders, and objects

Note: Users, groups, and roles must all have unique names.

For more information about security administration, see *SAS Intelligence Platform: Security Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

How the LOCKDOWN Statement Affects Campaigns

Overview of the LOCKDOWN Statement

The LOCKDOWN statement secures a SAS Foundation server by restricting access from within a server process to the host operating environment. A SAS server in the locked-down state validates all access to the host file system through the lockdown path list. This list is often referred to as a whitelist.

For more information, see *SAS Intelligence Platform: Application Server Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

LOCKDOWN and External Treatment Lists

On the **Settings** tab of the business context Properties window, you can select tables that are used in an external treatment list. If the tables were registered before the LOCKDOWN statement was issued, you can select the tables even if they are not on the whitelist. However, campaign treatments cannot reference the tables.

LOCKDOWN, Stored Processes, and Export Files

By default, the current working directory for the server is included in the whitelist. Campaigns and stored processes can create export files and tables in subdirectories of the working directory.

For example, if `/tmp` is the working directory, export files can be created in the `/tmp/exportfiles` subdirectory.

Administering SAS Identities for Users

Overview of SAS Identities

For each user, you must create an individual SAS identity on the SAS Metadata Server. The SAS identity is a copy of the ID with which the user logs on to SAS applications. Based on this identity, the system can determine who can access which application, and can audit individual actions in the metadata layer. The SAS identity consists of a name and the user ID for the user's external account. This ID can be any type of account that is known to the metadata server's host such as an LDAP, Active Directory, host, or other type of account. When entering user IDs for Windows accounts, be sure to qualify the ID (for example, `WIN\myID` or `myID@mycompany.com`).

Create SAS Identities

To create SAS identities for your users, manually enter the information for each user through the User Manager plug-in in SAS Management Console. If you have a large number of users, then you can extract user and group information from one or more enterprise identity sources. You can then use SAS bulk-load macros to create the identity metadata from the extracted information.

For more information, see *SAS Intelligence Platform: Security Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

The SAS Customer Intelligence Services User ID

A SAS Customer Intelligence Services User ID is created during the installation and configuration of SAS Customer Intelligence. The user ID has access to several system-level operations. The user ID is automatically assigned to the correct roles and capabilities. Make sure that this user ID has access permissions for the databases that are going to be used. This user ID must also have access to all business contexts. This user ID is added by default to the Customer Intelligence access control template during installation. For more information, see "Using the Access Control Template (ACT)" on page 61.

Administering Groups and Roles

About Roles

In SAS Customer Intelligence applications, certain actions are available only to users or groups that have a particular role. A role is a set of capabilities that correspond to particular application features such as menu items and plug-ins. Any user or group who is a member of a role has all of that role's capabilities.

Roles can contribute to one another. A role automatically includes all of the capabilities of a role that contributes to it.

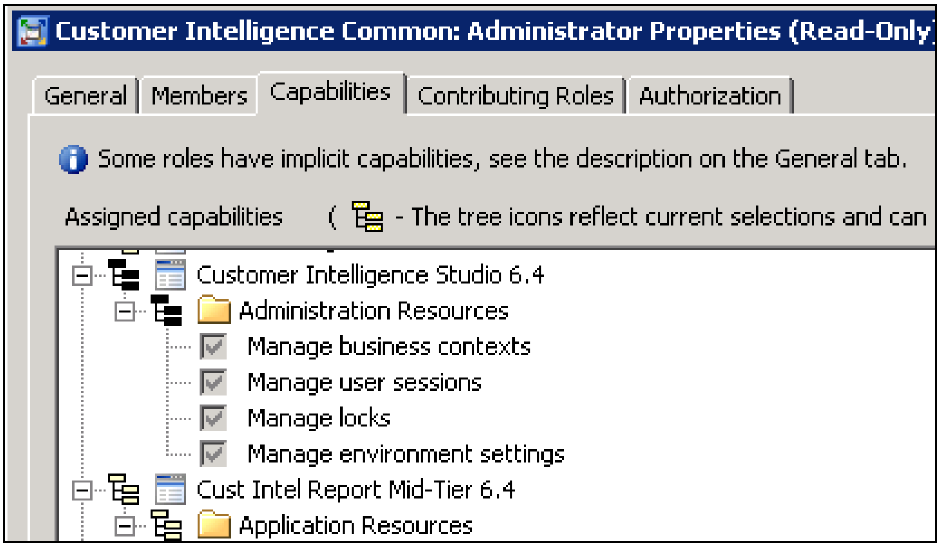
Roles differ from permissions. In general, roles do not affect access to metadata or data.

Viewing Roles in SAS Management Console

To view details about roles, open the User Manager plug-in in SAS Management Console, right-click the role, and select **Properties**. You can then view tabs that display the role's members, capabilities, and contributing roles.

For example, here is the **Capabilities** tab for the initial configuration of the Customer Intelligence Common: Administrator role:

Display 3.1 Capabilities Tab for the Customer Intelligence Common: Administrator Role



Customer Intelligence Common: Administrator Properties (Read-Only)
General
Members
Capabilities
Contributing Roles
Authorization

Some roles have implicit capabilities, see the description on the General tab.

Assigned capabilities (- The tree icons reflect current selections and can

- Customer Intelligence Studio 6.4
  - Administration Resources
    - ☑ Manage business contexts
    - ☑ Manage user sessions
    - ☑ Manage locks
    - ☑ Manage environment settings
- Cust Intel Report Mid-Tier 6.4
  - Application Resources

The following icons provide information about the capabilities:

Means that none of the capabilities in this category have been specified for this role.

Means that some of the capabilities in this category have been specified for this role, either explicitly or through a contributing role.

Means that all of the capabilities in this category have been specified for this role, either explicitly or through a contributing role. To see details, click the plus sign (+).

- Shaded check boxes indicate capabilities that come from contributing roles.
- Some roles have implicit capabilities that are not specified on the **Capabilities** tab.

About Predefined Roles for SAS Customer Intelligence

Your installation includes several predefined roles for administrators and users of SAS Customer Intelligence. Depending on what software you have installed, you might have other predefined roles.

The following roles are provided for users and administrators:

Customer Intelligence: Usage
: This role provides the capability to log on to SAS Customer Intelligence applications. This role contributes to the other Customer Intelligence roles. The capability is implicit. It cannot be selected from the SAS Management Console.

Customer Intelligence: Basic Campaign Design
: In addition to the Customer Intelligence: Usage capabilities, users in this role have the implicit capability to access nodes with **All users** permission to design new campaigns. Node permissions are set in the Diagram Tools category of the Setup workspace in SAS Customer Intelligence Studio. For more information, see *SAS Marketing Automation: User's Guide*.

Customer Intelligence: Advanced Campaign Design
: In addition to the Customer Intelligence: Usage and Customer Intelligence: Basic Campaign Design capabilities, users in this role can access nodes with **Advanced users** permission to design new campaigns, write code in Process nodes, and access operations and application resources features. Node permissions are set in the Diagram Tools category of the Setup workspace in SAS Customer Intelligence Studio. For more information, see *SAS Marketing Automation: User's Guide*.

Customer Intelligence: Administration
: In addition to the Customer Intelligence: Usage capability, users in this role can manage diagram tools, optimization, channels, information map metadata, staged treatments, custom processes, and SAS Digital Marketing integration.

Customer Intelligence Common: Administrator
: Users in this role can access SAS Customer Intelligence Studio from the SAS Visual Analytics Home page and manage the categories in the Administration workspace: business contexts, user sessions, locks, and environment settings.

Note: The ability to access and update campaign metadata is subject to permissions that are placed on that metadata. The SAS Customer Intelligence roles do not affect permissions.

The following tables provide details about the capabilities in each of the predefined roles.

Table 3.1 Description of Capabilities

Capability	Description
Log on to Customer Intelligence applications	enables the user to sign in to SAS Customer Intelligence Studio. This capability cannot be selected.

Manage selection campaigns	enables the user to manage SAS Marketing Automation campaigns.
Manage selection campaign groups	enables the user to manage SAS Marketing Automation campaign groups.
Edit comments	enables the user to edit existing comments.
Manage treatments	enables the user to view and use treatments.
Delete comments	enables the user to delete existing comments.
View reports	enables the user to view reports in the Reports workspace.
Manage reports	enables the user to create, save, rename, and delete reports in the Reports workspace.
Allow use of advanced nodes	enables the user to create and use nodes with **Advanced user** permission.
Allow writing code in Process node	enables the user to enter code in the Process node.
Allow send to administrator for campaign or group schedule	enables the user to send details of a campaign or campaign group schedule to an administrator.
Allow send to scheduler for campaign or group schedule	enables the user to send details of a campaign or campaign group schedule directly to the scheduling software.
Execute campaign or group	enables the user to execute a campaign or a campaign group manually or through the scheduling software.
Manage calculated items	enables the user to create, edit, and delete calculated items.
Manage selection campaign definitions	enables the user to view and use SAS Marketing Automation campaign definitions.
Manage campaign group definitions	enables the user to view and use SAS Marketing Automation campaign group definitions.
Manage export definitions	enables the user to view and use export definitions.
Manage seeds	enables the user to view and use seeds.
Manage communication definitions	enables the user to view and use communication definitions.

Manage selection custom diagram tools	enables the user to view and use SAS Marketing Automation custom diagram tools.
Manage global variables	enables the user to view and use global variables.
Manage custom detail groups	enables the user to view and use custom detail groups.
Manage response definitions	enables the user to view and use response definitions.
Manage built-in diagram tools	enables the user to manage diagram tools that are supplied with the application.
Manage optimization	enables the user to optimize campaigns and campaign groups.
Manage channels	enables the user to manage channels.
Manage information map metadata	enables the user to manage information map metadata.
Manage staged treatments	enables the user to manage staged treatments.
Manage custom processes	enables the user to manage custom processes.
Manage SAS Digital Marketing integration	enables the user to use SAS Digital Marketing to create email broadcasts from SAS Marketing Automation.
Manage business contexts	enables the user to view and use business contexts.
Manage user sessions	enables the user to manage user sessions.
Manage locks	enables the user to view and release locked objects such as campaigns.
Manage environment settings	enables the user to manage environment variables.
Allow preview of export tables	enables the user to preview export tables in the Communication node or the Export node.

In the following table, an asterisk (*) indicates that the capability is from a contributing role.

Table 3.2 Predefined Roles and Assigned Capabilities

Capability	Customer Intelligence: Usage Role	Customer Intelligence: Basic Campaign Design Role	Customer Intelligence: Advanced Campaign Design Role	Customer Intelligence: Administration Role	Customer Intelligence Common: Administrator Role
Log on to Customer Intelligence applications	X	X*	X*	X*	X*
Manage selection campaigns		X	X		
Manage selection campaign groups		X	X		
Edit comments	X	X*	X*	X*	X*
Manage treatments		X	X		
Delete comments	X	X*	X*	X*	X*
View reports		X	X		
Manage reports			X		
Allow use of advanced nodes			X		
Allow writing code in Process node			X		
Allow send to administrator for campaign or group schedule			X		
Allow send to scheduler for campaign or group schedule			X		
Execute campaign or group			X		

Manage calculated items			X		
Manage selection campaign definitions			X		
Manage campaign group definitions			X		
Manage export definitions			X		
Manage seeds			X		
Manage communication definitions			X		
Manage selection custom diagram tools			X		
Manage global variables			X		
Manage custom detail groups			X		
Manage response definitions			X		
Manage built-in diagram tools			X	X	
Manage optimization			X	X	
Manage channels				X	
Manage information map metadata				X	
Manage staged treatments				X	
Manage custom processes				X	

Manage SAS Digital Marketing integration				X	
Manage business contexts					X
Manage user sessions					X
Manage locks					X
Manage environment settings					X
Allow preview of export tables					

Creating New Roles

The predefined roles might be sufficient for many sites. Other sites might need to make application features available to users on either a broader or more granular basis than the predefined roles allow. For example, you might want to enable a user to manage treatments, but not perform other application management tasks. In this case, create a role that specifies only the **Manage treatments** capability.

Other combinations of capabilities can be used to create a new role. You can use only the capabilities that already appear in User Manager. You cannot create new capabilities.

For detailed information about roles and how to create them, see *SAS Intelligence Platform: Security Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Modifying Roles

The User Manager plug-in in SAS Management Console enables you to modify roles by selecting or deselecting different capabilities.

CAUTION! No automated method can revert a role to its original set of capabilities. Instead of adjusting the capabilities of a predefined role, consider creating a new role. This advice is especially important if major changes are needed.

If you modify a role, then follow these best practices:

- Do not rename the predefined roles. Renaming the predefined roles makes it difficult for SAS Technical Support to help you resolve problems.
- Back up the metadata server before modifying roles, and keep a record of the changes that you make.

When modifying a role, you can use only the capabilities that already appear in User Manager. You cannot create new capabilities.

For more information about roles and how to modify them, see *SAS Intelligence Platform: Security Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

About Groups

Groups enable you to give multiple users membership in a role or permissions to metadata, thus simplifying security administration. You can create as many groups as are needed in order to manage your installation. In order to manage administration resources such business contexts, user sessions, and environment settings, a user must be a member of the Customer Intelligence Common Administrator group.

Note: The user interface displays only groups that are created with the Customer Intelligence Usage: Role.

The following groups are provided in your initial installation:

SAS Users
: This group includes everyone who can access the metadata server, either directly or through a trust relationship. If a user is able to log on to a client application and has an individual SAS identity, the user is assumed to be in this group. Because this group has implicit membership, you cannot explicitly add or remove users from this group.

Public
: This group includes everyone who can access the metadata server, either directly or through a trust relationship. If a user is able to log on to a client application but does not have an individual SAS identity, the user is assumed to be in the public group. Because this group has implicit membership, you cannot explicitly add or remove users from this group.

SAS Administrators
: This is a standard group for metadata administrators. In a standard configuration, members are granted broad access and administrative capabilities, but are not unrestricted.

Customer Intelligence Basic Campaign Designer
: In your initial installation, this group is a member of the Customer Intelligence: Basic Campaign Design role. You can add users to this group to give them access to basic campaign design functionality.

  If SAS Marketing Operations Management is installed at your site, the following groups are added to the Customer Intelligence Basic Campaign Designer group. These groups have the same capabilities as the Customer Intelligence Basic Campaign Designer group.

  - Marketing Operations Integration Analysts
  - Marketing Operations Integration Services

Customer Intelligence Advanced Campaign Designer
: In your initial installation, this group is a member of the Customer Intelligence: Advanced Campaign Design role. You can add users to this group in order to give them access to advanced campaign design functionality.

  If SAS Marketing Operations Management is installed at your site, the following groups are added to the Customer Intelligence Advanced Campaign Designer group. These groups have the same capabilities as the Customer Intelligence Advanced Campaign Designer group.

- Marketing Operations Integration
- Marketing Operations Integration Campaign Designer

Customer Intelligence Administrator
: In your initial installation, this group is a member of the Customer Intelligence: Administration role. You can add users to this group in order to enable them to administer Customer Intelligence applications.

Customer Intelligence Common Administrator
: In your initial installation, this group is a member of the Customer Intelligence Common: Administrator role. You can add users to this group in order to enable them to manage administration resources such as business contexts, user sessions, and environment settings.

SAS System Services
: This group enables members to export files on the **Folders** tab of SAS Management Console.

Administering Group and Role Membership

To administer group and role membership, use the User Manager plug-in in SAS Management Console. In most cases, the best way to place a user in a role is to add the user to a group that belongs to the role. You can also add users directly to roles.

To place a user in one of the predefined roles, you can add the user to one of the predefined groups:

- To add a user to the Customer Intelligence: Basic Campaign Design role, add the user to the Customer Intelligence Basic Campaign Designer group. The user also receives the capabilities of the contributing role, Customer Intelligence: Usage.
- To add a user to the Customer Intelligence: Advanced Campaign Design role, add the user to the Customer Intelligence Advanced Campaign Designer group. The user also gets the capabilities of the contributing roles, Customer Intelligence: Basic Campaign Design and Customer Intelligence: Usage.
- To add a user to the Customer Intelligence: Administration role, add the user to the Customer Intelligence Administrator group. The user also gets the capabilities of the contributing role, Customer Intelligence: Usage.

Note: There is no reason to add a user directly to the Customer Intelligence: Usage role. This role enables a user to log on, and can be used to filter lists of groups in other areas of the software.

A user can be added to more than one group, and a user or group can be added to more than one role. For example, suppose a user needs to perform both administration tasks and advanced application tasks. You could take one of the following actions:

- Add the user to both the Customer Intelligence Administrator group and the Customer Intelligence Advanced Campaign Designer group. This method might be appropriate if only one user needs this combination of capabilities.
- Create a new group called Customer Intelligence: Administrator and Advanced. You could then add the new group to both the Customer Intelligence: Administration role and the Customer Intelligence: Advanced

Campaign Design role. This method might be appropriate if multiple users need this combination of capabilities.

Capabilities and Campaign Execution

Users who do not have the Execute Campaign capability cannot execute Communication, Export, Code, Process, or Custom nodes. These users can update counts on these nodes.

Capabilities and the Reports Workspace

The ability to view and edit reports in the Reports workspace is controlled by two capabilities.

Manage reports
enables the user to create, save, rename, and delete reports.

View reports
enables the user to view reports.

These capabilities are in the `Applications` folder under `Cust Intel Report Mid-Tier`.

The ability to view reports must also be enabled for each business context. For more information, see *SAS Marketing Automation: User's Guide*.

Capabilities and Previewing Export Tables

In order to be able to preview export tables in the Communication node or the Export node, the user must have the **Allow preview of export tables** capability. This capability is not turned on by default for any role.

Capabilities and Scheduling

The ability to send a campaign or campaign group schedule to the scheduling software is controlled by two capabilities:

- Allow send to scheduler for campaign or group schedule
- Execute campaign or group

To be able to send a schedule to the scheduling software, the user must have both capabilities. If a user does not have these capabilities, the following capability enables the user to send a schedule to an administrator:

- Allow send to administrator for campaign or group schedule

In this case, the administrator must be able to send the schedule to the scheduling software and to execute the campaign or group.

Capabilities and Optimization

If SAS Marketing Optimization is installed at your site, users must have certain capabilities before they can perform some optimization tasks in SAS Marketing Automation. The following table lists the required SAS Marketing Optimization capabilities.

Table 3.3 SAS Marketing Optimization Capabilities

Task	Capability
Create new scenario	Perform scenario analysis
Edit scenario	Perform scenario analysis
Generate optimization data	Manage optimization data
Optimize now	Perform scenario analysis

Administering Permissions

About Permissions

SAS provides a metadata-based authorization layer that supplements the protections from the host environment and other systems. Protections are cumulative across authorization layers. In order to perform a task, a user must have sufficient access in all of the applicable layers.

Although permissions can be assigned to individual users, it is recommended that you assign permissions for groups and then place users in those groups. Placing users in groups with previously assigned permissions decreases the work of maintaining your permissions structure and helps you avoid orphaned objects for which no users have permissions

You can set permissions at several levels:

- Repository-level controls provide the default access controls for objects that do not have other access controls.
- Resource-level controls manage access to a specific item such as an information map, a campaign, a node, a business context, or a folder. The controls can be defined individually by using explicit settings or in patterns by using access control templates.
- Fine-grained controls affect access to subsets of data within a resource. You can use these controls to specify who can access either particular rows within a table or members within a cube dimension.

The effect of a selected permission setting is influenced by any related settings that have higher precedence. For example, if a campaign inherits a permission from its parent folder but also has an explicit denial, then the explicit setting determines the outcome. Similarly, if a group has been granted a permission, and a user who is a member of the group has an explicit denial, then the explicit setting determines the outcome.

Permissions are set by using the following methods:

- The access control template (ACT), which provides a set of default permissions.

- Business contexts, which are groupings of campaigns. When you create a business context, you specify which users can log in, and which users can view or edit the business context properties.
- Custom repositories, which you can use to physically separate metadata for storage or security purposes.

Each of these methods is described in more detail below.

Using the Access Control Template (ACT)

The Customer Intelligence access control template (ACT) provides a set of default permissions for SAS Marketing Automation resources. This template is automatically applied to all Customer Intelligence objects, including business contexts, campaigns, and nodes. In its initial configuration, this template denies ReadMetadata (RM) and WriteMetadata (WM) permission to the public group.

You might want to update the ACT in these situations:

- Give one or more users broad access to campaign data for the purpose of application troubleshooting or administration. To do this, you can add the user to the ACT, either permanently or temporarily, and specify the appropriate permissions.
- Access campaign metadata that was created by a user who is no longer in your organization. To do this, temporarily add an administrator to the ACT so that the administrator can transfer the campaign permissions to a different user.

To update the ACT, open the Access Control Templates folder in the Authorization Manager plug-in in SAS Management Console.

Assigning User Permissions for Business Contexts

Campaigns are grouped into user-defined business contexts. Business contexts enable you to separate campaign depending on which users should have access. When you define a new business context, you specify which users can log in, and which users can view or edit the business context properties. Then, as campaigns are designed and created within the business context, the software applies access control entries (ACEs). ACEs give those users and groups the appropriate permission to access the campaign data and metadata.

You must add users and groups to the appropriate roles before you can give them View or Edit permission for a business context.

You can limit business context access to only groups by first using an administrative user to create all of the business contexts for your site. When the administrative user first creates a business context, that user is added by default to the **Permissions** tab and to the **Users** tab of the business context Properties window. The administrative user can add groups to the **Permissions** and **Users** tabs, and then remove the administrative user ID from both tabs.

CAUTION! Users who are listed in the SAS Customer Intelligence ACT should not create business contexts These users are excluded from the permissions list for business contexts. If they create business contexts, no other users have permission, and no users are listed on the permissions list.

New users and groups might not have immediate access to a business context. By default, the cache of user permissions for a business context is refreshed every four hours. To modify the refresh interval, enter the following code in the **`Config/Lev1/Web/WebAppServer/SASServer6_`*`node_number`*`/bin/setenv.sh`** file of the web application server.

```
-Dcom.sas.analytics.crm.bctx.cache.recache.minutes=number of minutes
```

Assigning User Permissions for Sites That Use SAS Digital Marketing

For sites that have SAS Digital Marketing, in order to prevent problems with the update of the common data model with SAS Digital Marketing response information, add the sastrust account to the users group for the database that is hosting the common data model tables. For example, if the common data model tables are in an Oracle database, enter the sastrust user to the DBUSERS(ORACLE) group in the SAS Management Console.

Assigning User Permissions for Custom Repositories

If you create a custom repository in order to physically separate metadata for storage or security purposes, then you can apply permissions to specify which users do and do not have access to the repository. In the **SAS Folders** tree in SAS Management Console, select the folder that represents the repository. Then open the Properties dialog box and update the **Authorization** tab.

For more information about permissions, see *SAS Intelligence Platform: Security Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html and the user’s guide for your product.

Enable Integrated Windows Authentication in Firefox

If you are using the Firefox browser in an environment that requires Integrated Windows Authentication, you must add your network to the Firefox configuration. This enables you to log on to SAS Customer Intelligence Studio. To add your network to the Firefox configuration:

1. In the Firefox browser address bar, type `about:config`.
2. A warning message is displayed. Click the button to indicate agreement and display the configuration page.
3. On the configuration page, type `network.automatic` in the **Search** field.
4. Double-click **network.automatic-ntlm-auth.trusted-uris**.
5. Enter your site in the dialog box. Separate multiple sites with a comma. For example, you could enter multiple sites as `http://sas.com, http://myintranetsite.com`
6. Type `network.negotiate` in the **Search** field.
7. Double-click **network.negotiate-auth.delegation-uris** and enter your network name in the dialog box. For example, you could enter a network name as `cinetwork.com`.

8 Double-click **network.negotiate-auth.trusted-uris** and enter your network name in the dialog box.

For more information, see https://support.mozilla.org/en-US/products/firefox.

Password Updates

During SAS Customer Intelligence installation, passwords for the following accounts are updated when **Update Passwords** is selected from SAS Deployment Manager.

- SAS Marketing Optimization Data Store Account
- Customer Intelligence Scheduling User ID
- Customer Intelligence Services User
- SAS Digital Marketing Server Administrator account
- SAS Digital Marketing Mail Transport Agent

Note: Passwords that are controlled by an external provider (such as in LDAP, Active Directory, or the host operating system) are not synchronized. Make sure that the passwords that you provide as input match the actual passwords in your external provider.

4

Folders in SAS Management Console

Overview of Folders

The **Folders** tab of SAS Management Console displays the folders that contain metadata for SAS Customer Intelligence objects. You can import and export SAS Packages, and copy and paste objects between folders. SAS Customer Intelligence does not use the `My Folder` or `Shared Data` folders.

Folder permissions are applied to the contents of the folders. If a user has permission to edit a folder, then the user can edit the contents of the folder and delete an empty folder. If a user has permission to view the folder, then the user can view the contents of the folder. The user cannot add content to the folder, and cannot delete the folder. If a user has no access permissions to the folder, then the user cannot view or add contents to the folder, and cannot delete the folder. For more information about administrative roles and setting permissions, see the online Help for SAS Management Console.

You can create folders, but you cannot rename folders that are used by SAS Customer Intelligence.

When you create a folder, it inherits the permissions of the parent folder. You can change the inherited permissions.

Assets and Permissions

The following types of assets are viewable on the **Folders** tab of SAS Management Console:

- System assets such as SAS Customer Intelligence applications are stored in the `/System/Applications/SAS Customer Intelligence` folder.
- Campaign assets, such as campaign definitions and campaign processes, are stored in the folder that is assigned to the business context for the campaign .

Permissions for these two types of assets are configured differently.

Any user who has permission to write to system assets can also copy, paste, export, import, and delete the assets. No additional configuration is required.

To perform operations on campaign assets, the user must be a member of the Customer Intelligence: Usage role. Even with full Write permission, the user cannot perform any of these operations without membership in the Customer Intelligence: Usage role. For information about roles, see "Administering Groups and Roles" on page 49.

If you have Write permission to campaign assets, you can set permissions for an additional user or modify the permissions for an existing user. Within a business context folder, right-click an asset such as a campaign definition, and select **Properties**. Modify access permissions on the **Authorization** tab.

Location of SAS Customer Intelligence Objects

Most SAS Customer Intelligence objects, such as campaigns and campaign definitions, are stored in folders that are specified by the user. Response definitions are stored in a user folder in `Products\SAS Customer Intelligence`. Identifiers, calculated items, and custom detail tags do not appear in the **Folders** tab in SAS Management Console.

There are two types of business context objects in SAS Management Console. Note the difference in the capitalization of the word "intelligence."

Customer intelligence business context
: is a business context that was created in a release prior to SAS Customer Intelligence 6.1.

Customer Intelligence business context
: is a business context that was created in SAS Customer Intelligence 6.1 or later releases.

If you search for business contexts, make sure that you select the correct type on the **Search** tab of SAS Management Console.

Copying and Pasting

You can copy and paste folders and their contents. Before you copy and paste objects that depend on information maps, SAS Marketing Automation must be installed and configured, and information maps, business contexts, users, and groups must be defined. For objects that do not depend on information maps, SAS Marketing Automation must be installed and configured. Use copy and paste, rather than export and import, to move an item to a new business context on the same machine. The codes that are retained depend on the settings for the business context.

To paste an item into a folder, you must have the appropriate access permissions for the folder. An administrator can temporarily add a user to the access control template. The user can be removed after the required operation has been completed.

To copy an item, right-click the item and select **Copy**. To paste an item, right-click the destination and select **Paste** or **Paste Special**.

Note: You cannot rename objects.

When you select **Paste Special**, a copy of the item is created. If an item by the same name already exists in the target destination, the item is named `Copy of` *item name*.

To use **Paste Special** to include all of the dependencies of a copied object, select the object on the **Select Objects to Copy** page of the Paste Special wizard. Select **Select All** to include all of the objects that are listed on the **Dependencies** tab. If you select individual objects on the **Dependencies** tab, select the objects under the **Copied Objects** folder, and then select their dependencies on the **Dependencies** tab.

If you are copying assets such as campaign definitions, use **Paste Special** to retain all of the dependencies. For example, copies of campaigns and the campaigns that they link to are installed with the same folder structure as the original items. However, assets that are associated with inbound campaigns do not retain their dependencies. Tags that are no longer in the same environment as the copied treatment are not retained. If you use **Paste Special** to paste copied items directly into the **SAS Folders** folder, the source paths are not preserved. Instead, use **Export SAS Package** and **Import SAS Package** to preserve source paths in the **SAS Folders** folder.

When you export a communication definition, the linked response definitions are displayed on the **Dependencies** tab of the Export SAS Package window. When you export a response definition, the linked communication definitions are displayed on the **Used By** tab of the Export SAS Package window. The link between communication and response definitions is retained when the linked objects exist in the imported package.

Copied items retain the permissions of the original items. You can change the permissions of individual objects, such as campaign definitions. Do not change permissions on processes; there might be several campaigns that rely on access to a process.

When you copy and paste campaigns and treatments, codes and control group names are retained according to the setting for the business context, and new surrogate keys are generated in the Common Data Model. Codes and control group names are retained only when the copied object is pasted within the same business context.

Importing and Exporting SAS Packages

The SAS Package Wizard

You can use the SAS Package Wizard to export and import collections of objects and their dependencies into subfolders in the SAS Folders tree. To export the contents of a folder, right-click the folder and select **Export SAS Package**. You can then select the objects for inclusion in the export and save a SAS package file. To import a collection of objects and their dependencies, right-click the destination, select **Import SAS Package**, and select a SAS package file. The best practice is to export and import all objects simultaneously, to avoid losing the dependencies between objects. The system that you are exporting from and the system that you are importing to must have the same encoding. Imported items retain their original names and overwrite items of the same name that are in the target location. You can import all objects or only new objects.

Make sure that users are logged off from SAS Customer Intelligence Studio before you import or export SAS packages. For more information, see “Manage User Sessions” on page 40.

For more information about the SAS Package Wizard, see the Help for SAS Management Console.

User Permissions

A user who has View permission can export a SAS package. The import process automatically grants the importing user WriteMetadata permission on newly imported objects if the user does not already have this permission.

To import an item into a folder, you must have the appropriate access permissions for the folder. An administrator can temporarily add a user to the access control template. The user can be removed after the required operation has been completed.

If you select **Include access controls** in the Import SAS Package wizard, you must have Write permission to the objects that are imported.

Business Contexts

Source and target business contexts can be in different folder locations relative to their root directories.

If you export business context assets separately, and you want to keep the relationship between assets, the assets must be imported to the same folder locations relative to each business context root folder. For example, if you want to import a campaign separately from the campaign group that references it, and you want the linkage to be automatically restored, then the campaign should be imported to the same relative folder. If the assets are exported and imported together, they do not have to be imported to the same folder locations relative to each business context. Assets that are not referenced by other assets can be imported into any folder.

Calculated Items

Calculated items that are used in an object are exported with the object. When the object is imported, the calculated items are mapped to existing calculated items, or they are added to the Calculated Items category in SAS Customer Intelligence Studio.

Comments

Comments are removed from the imported object.

Common Data Model

The publishing information for imported campaigns is reset, except for the following codes in the common data model:

- CAMPAIGN_CD
- COMMUNICATION_CD
- MARKETING_CELL_CD
- PACKAGE_CD

Custom Diagram Tools and Custom Nodes

Custom diagram tools cannot be imported into a new environment. They must be re-created in the campaign in the destination environment.

Custom nodes that are part of a diagram are exported and imported with the campaign. For information about exporting and importing a stored process that is associated with a Custom node, see “Stored Processes” on page 70.

Definitions

Definitions should be imported into the corresponding folder in the new environment. Because definitions are shared across business contexts, a change to a shared definition results in changes to the definition in other business contexts. If you include dependent objects when you export a campaign, the campaign definition is not exported.

If importing a response definition would result in a duplicate channel response code for a specific channel, the channel response code field is empty in the imported definition.

When you import a definition, the only business contexts that remain associated with that definition are those business contexts for which you have Write permission.

Communication Nodes and Export Nodes

To view complete details of the Communication and Export nodes that are associated with specific paths, open the imported package. View the SubstitutionProperties.subprop file, which contains details of each Communication node, export definition, and output name combination.

Link Nodes

If a Link node is used in the campaign, the campaign that includes the cell that is linked to is included in the export.

Stored Processes

When a Process node or a Custom node includes a stored process, the stored process is associated with the node by pathname and by folder. You can create different stored processes of the same name in different folders.

When you export and import campaigns, be sure to retain the paths between nodes and their stored processes. You first export and import the stored process and then you export and import the campaign. To export and import a campaign and the associated stored processes:

1. On the **Folders** tab of SAS Management Console, right-click `SAS Folders` and select **Export SAS Packages**.
2. Select the stored processes that you want to export and create the package file.
3. Right-click `SAS Folders` and select **Import SAS Package**.
4. In the Import SAS Package wizard, select the package that contains the stored processes.
5. Select **Preserve source path information for objects in the package** and finish importing the package.
6. Export and import the campaign into the destination environment.

Treatments

When you import treatments along with a campaign, an information icon (ⓘ) indicates that you should verify that the treatments are correct when you open the imported campaign. Imported treatments with attached images must map to treatments in the destination folder. Otherwise, the images are not available to campaigns.

Memory Size

If you import a large package, you might need to increase memory size. In the sasmc.ini file, modify the following code.

```
JavaArgs_1=-Xmx1024m
```

If you import a large package, you might need to increase memory size. In the ImportPackage.ini file, modify the following code.

```
JavaArgs_1=-Xmx1024m
```

Increasing the memory size to the following value allows the import process to accommodate a package of more than 5 megabytes.

```
JavaArgs_1=-Xmx2048m
```

Best Practices for Exporting and Importing Objects

To configure the destination environment, do the following:

1. Create a folder structure in the destination environment that is identical to the folder structure in the source environment.
2. Create an information map in the destination environment that is identical to the information map in the source environment.
3. Create user names in the destination environment that are identical to the user names in the source environment. Assign the same roles and capabilities to the destination user names. User names are case-sensitive.
4. Create a repository in the destination environment that has the same name as the repository in the source environment.
5. Create a business context in the destination environment that is identical to the business context in the source environment. The destination business context must have the same name as the source business context.
6. If the objects in the source environment contain dynamic lists, create data libraries in the destination environment that contain the list values. Make sure that the data libraries in the destination environment and the data libraries in the source environment have the same names.
7. You must have Write permission in order to export and import objects. An import that fails because of permission issues results in a corrupted object. The visual indicator of a corrupted object is a red exclamation point (❗). This object can be deleted only by a user with Write permission.

To export objects from the source environment, do the following:

1. Select the SAS Management Console objects, such as campaign definitions, that you want to export. Right-click the objects and select **Export SAS Package**. Select **Include dependent objects when retrieving initial collection of objects** in the Export SAS Package Wizard. Select a folder to export the contents of the folder.
2. Select the SAS Customer Intelligence objects, such as campaigns, in the folder where they are stored. Right-click the objects and select **Export SAS Package**. Select **Include dependent objects when retrieving initial collection of objects** in the Export SAS Package Wizard. Select a folder to export the contents of the folder.

3. Right-click the destination folder and select **Import SAS Package**. Select **Include access controls** in the Import SAS Package Wizard.

The exporting and importing process retains all codes, but regenerates surrogate keys.

For information about using command-line utilities to export and import objects, see “Batch Export and Import Tools” on page 180 and “Update Export Paths” on page 178.

Importing and Pasting Calculated Items

When you import or use **Paste Special** to paste calculated items in the **Folders** tab, keep in mind the following results:

- If an item with the same name already exists in the target folder, the item is named *item-name* (*n*), where *n* is an incremented number.
- If there are two items that are identical in all but name, and an item matches these items in the target folder, both pasted or imported items are mapped to the item that exists in the target folder.
- If there are two items that are identical in all but name, and two items with different names match these items in the target folder, both pasted or imported items are mapped to the first item that is first in alphabetical order.
- If there are two items that are identical in all but name, and two items with the same names match these items in the target folder, both pasted or imported items are mapped to the items that existed in the target folder.
- If there are two items that are identical in all but name, and one target item has the same name and one target item has a different name, the imported or copied item is mapped to the target item with the same name. The second item is mapped to the item that is first in alphabetical order.

5

SAS Information Maps for Campaigns

Identifying Data for Campaign Activities

As an administrator, you find the data that is available to SAS Customer Intelligence users. You then identify all of the data by creating a single information map in SAS Information Map Studio. SAS Customer Intelligence users analyze this data, prioritize targets based on this data, and export this data.

For example, you might want to include demographic data about customers such as buying patterns or account balances. You might also want to include any information for SAS Customer Intelligence Studio to export directly, such as name and address data.

You can include as many physical tables as you want. Note, however, that performance is optimized when fewer tables are used.

Although some businesses need only one information map, some large enterprises create additional information maps for various business contexts. New business contexts can be created and associated with an existing information map in SAS Customer Intelligence.

Note: Data items that are used as variables in optimization scores must not exceed 32 bytes.

The Role of the Information Map in SAS Customer Intelligence

What Is an Information Map?

An information map is a collection of data items that describe and present a view of physical data tables in a form that is relevant and meaningful to a business user. An information map does not contain any data. It is a map to the physical data. A SAS Customer Intelligence information map describes the data sources that enable SAS Customer Intelligence users to easily create campaigns. For example, instead of viewing a large collection of tables and columns, the business user views a simple list of business terms.

All information maps are stored in the SAS Metadata Repository in a folder that is designated as `ReportStudio/Maps`. On the **Folders** tab of SAS Management Console, SAS Customer Intelligence information maps are typically stored in an `Information Maps` subfolder. You can save an information map in any folder on the SAS server tier. For more information, see *SAS Marketing Automation: User's Guide*.

Note: SAS Customer Intelligence Studio does not support information map filters.

Creating a SAS Information Map for SAS Customer Intelligence

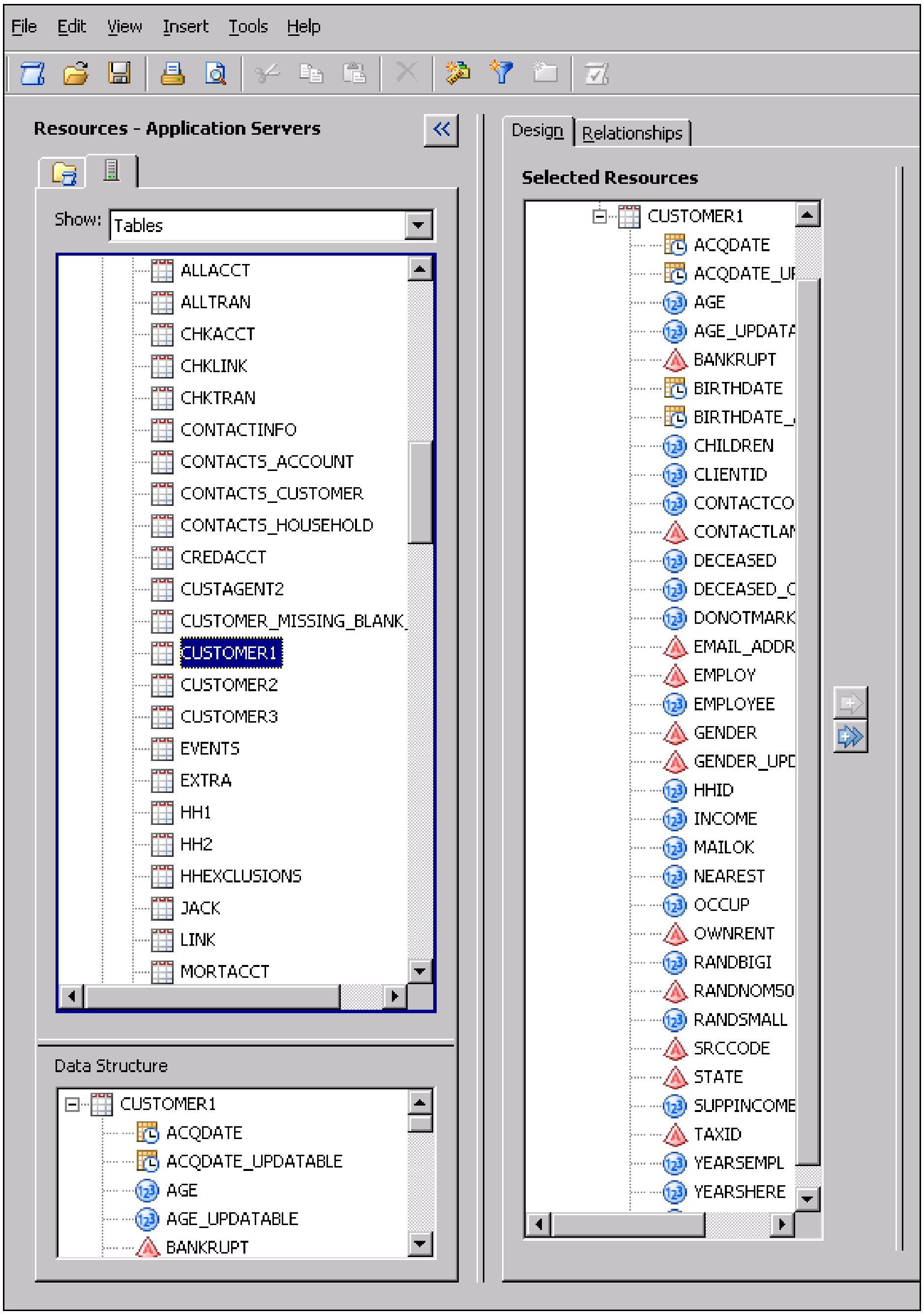
Overview of Steps

After you register the locations of the physical data, you create an information map in order to make data available to SAS Customer Intelligence.

To create an information map, follow these steps.

1 In SAS Information Map Studio, specify the data sources for the data items that are available to SAS Customer Intelligence users. For more information, see "Organizing Your Folders as Categories" on page 76.

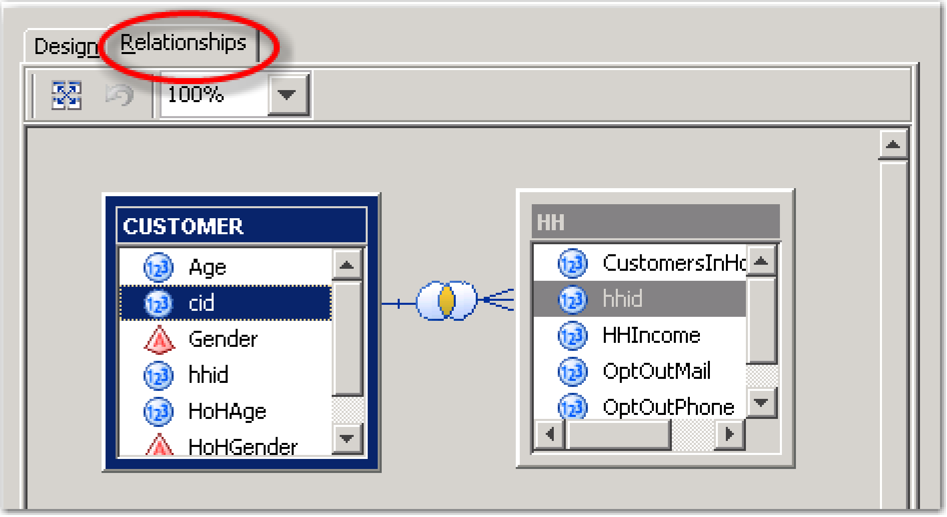
Display 5.1 *Selected Resources*



Show: Tables

2 Specify the table relationships on the **Relationships** tab to enable the retrieval of information from multiple tables.

Display 5.2 Relationships



For more information, see the online Help for SAS Information Map Studio and "Tips: Building Your Information Map" on page 76.

Note: Table relationships are established by specifying join keys and join types.

3 Specify any custom properties (extended attributes) in the information map. For more information, see "About Custom Properties" on page 79.

4 Save the information map. You can store it in any folder. For more information, see "Organizing Your Folders as Categories" on page 76.

Organizing Your Folders as Categories

In order to present the data in a business context, organize pertinent table columns as folders. Not all of the columns of a table are required to be added to the information map. For example, the attributes that are used to support the data loading processes can be hidden from users by not adding those attributes to the information map.

A folder can contain the data from one or more tables. Not all of the columns of a table have to be contained in a single folder; organize the data in order to help the end users perform their tasks.

Tips: Building Your Information Map

Adding a Relationship in a Cluttered Work Area

After many new data sources have been added, it becomes difficult to use the "select and drag" method to create a new relationship. When the **Relationship** tab becomes cluttered, use the following method to define a new relationship:

1 After you add a data source, right-click in a blank area of the entity box and select **Insert Relationship**.

2 Use the Insert Relationship dialog box to create the relationship.

Editing Entities That Are Joined with Multiple Fields

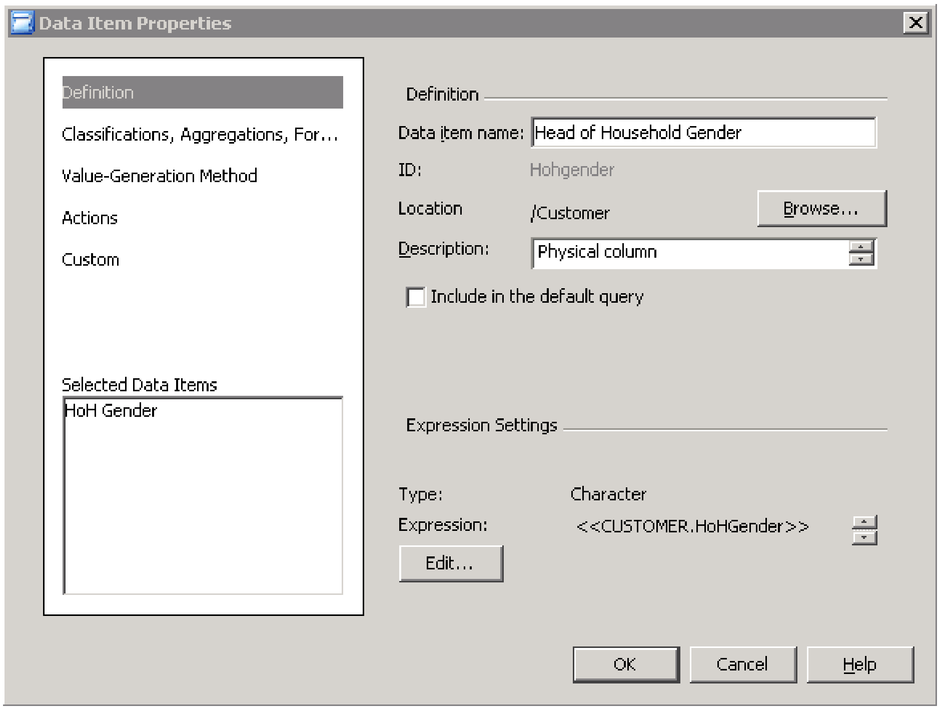
On the **Relationships** tab where entities are joined together using multiple fields, instead of dragging all of the fields at the same time, drag one field, and then edit the relationship in order to add the remaining fields.

Assign Meaningful Business Names

Replace the physical data names with more meaningful business names.

In the figure below, the column HoHGender is specified to be displayed as Head of Household Gender in SAS Customer Intelligence.

Figure 5.1 *Meaningful Business Names*



Data Item Properties

Definition
Classifications, Aggregations, For...
Value-Generation Method
Actions
Custom

Selected Data Items
HoH Gender

Definition
Data item name: Head of Household Gender
ID: Hohgender
Location /Customer
Description: Physical column
☐ Include in the default query

Expression Settings
Type: Character
Expression: <<CUSTOMER.HoHGender>>

Use the Cancel Button to Exit More Quickly

On the **Presentation** tab, if you make no changes, select **Cancel** to resume control more quickly than selecting **OK**.

Find Physical Data More Quickly

On the **Presentation** tab, when you are working with the physical data, close all of the open tables. Closing all of the open tables enables you to find the data to be mapped more quickly, especially when there are data sources that have many attributes.

Improve Performance When Generating the Counts Metadata

Eliminate the generation of the counts metadata for those data items where frequency counts are not applicable, thereby reducing the time it takes to generate metadata in general.

Best Practice: Use a Phased Approach to Create Your Information Map

Phase I: One Subject

A phased approach to creating an information map and generating its associated metadata can simplify troubleshooting.

Create a simple information map with one of each of the following data items:

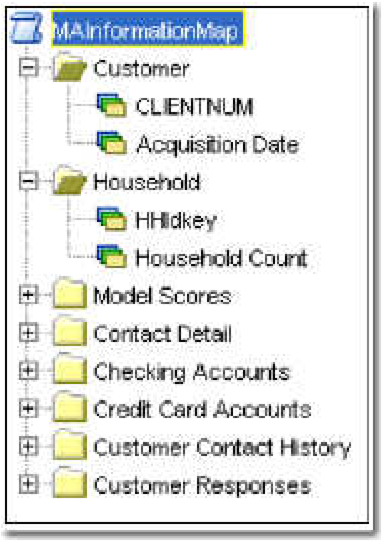
- table
- folder
- subject
- key
- other data item

Generate the counts metadata and validate it to ensure that the physical data and server infrastructure are working as expected.

Phase II: Multiple Subjects

Continue to build the simple information map by adding tables, subjects, subject keys, and folders. Add only one other data item to each folder. Generate metadata and validate it to ensure that table and subject relationships are properly specified.

Display 5.3 Multiple Subjects



Phase III: Fully Populated Map

Continue to build the information map incrementally by adding data items by folder, function, type, or any other logical progression until the information map is fully populated. Generate the counts metadata and validate it frequently to ensure that the physical data and server infrastructure are working as expected.

About Custom Properties

Overview

During the installation of SAS Information Map Studio, if you indicate that SAS Customer Intelligence is also installed, then a file named MAtemplate.txt is automatically loaded. MAtemplate.txt contains a prescribed list of custom properties that are needed to create information maps for SAS Customer Intelligence. For more information, see "Example: Adding Custom Subjects to MAtemplate.txt" on page 86.

Required Custom Properties

An information map that has two subjects requires the following custom properties:

- At the information map level:
  - Subject_ID_<*Name1*>
  - Subject_ID_<*Name2*>
  - From_Subject_ID_<*Name1*>_To_Subject_ID_<*Name2*>
  - From_Subject_ID_<*Name2*>_To_Subject_ID_<*Name1*>
  - MAMeta. For more information, see "Specify the MAMeta Library" on page 79.
  - MetadataTable_Prefix_Subject_ID_<*Name1*>
  - MetadataTable_Prefix_Subject_ID_<*Name2*>
  - Subject_Default
- At the data item level:
  - Level
  - UseInSubjectID

Specify the MAMeta Library

If you selected **Use information map** as the source of the metadata library on the **Metadata** tab of the business context, specify the MAMeta library as a custom property in the information map for the business context. For best performance, data should be stored on the SAS server rather than the data server.

You can also use a MAMeta libref that points to a SAS library as the source for the metadata library. For more information, see *SAS Marketing Automation: User's Guide*.

Note: Do not define the MAMeta library in autoexec_usermods.sas. Otherwise, whenever the first stored process that is started by SAS Customer Intelligence finishes running, it closes all libraries that are defined in the autoexec_usermods.sas file, including the MAMeta library. This action prevents SAS Customer Intelligence from accessing your marketing metadata.

To specify the MAMeta library as a custom property, follow these steps:

1 Using SAS Information Map Studio, locate and select the appropriate information map.

2 In the Information Map Contents pane, right-click the information map name, and select **Properties**.

3 Select the **Custom** tab and locate the MAMeta label in the table of custom properties.

   If the **Custom** tab is not displayed in SAS Information Map Studio, then see "Making Custom Properties Available" on page 82.

4 Specify the appropriate path for the MAMeta information map libref in the Value column.

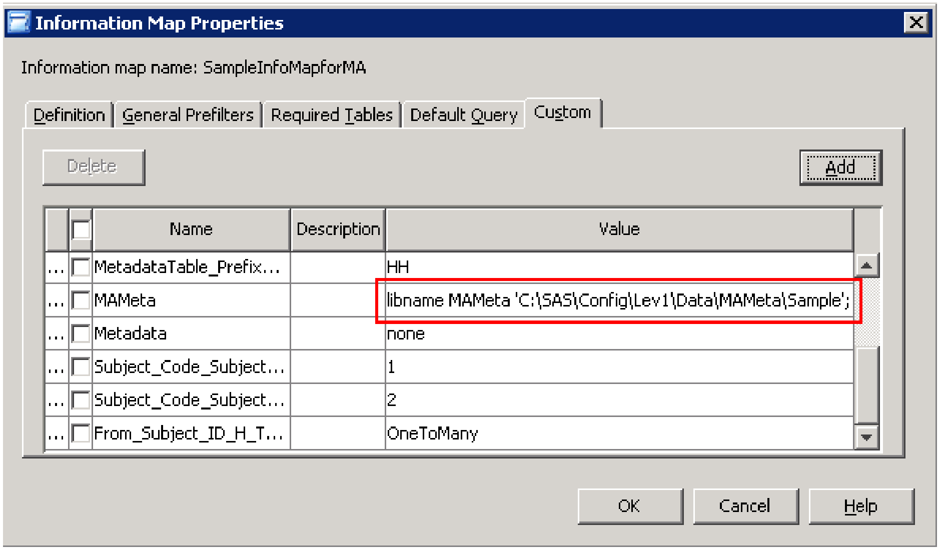
   If the libref points to a UNIX server, then follow these steps to ensure that the correct path is specified:

   a In a DOS window, navigate to the directory on the UNIX server where the MAMeta information map is located.

   b Type `pwd`. The full current path is returned.

   c Copy the path and paste it into the Value column.

Figure 5.2 MAMeta Pathname



Information map name: SampleInfoMapforMA

Name	Description	Value
MetadataTable_Prefix...		HH
MAMeta		libname MAMeta 'C:\SAS\Config\Lev1\Data\MAMeta\Sample';
Metadata		none
Subject_Code_Subject...		1
Subject_Code_Subject...		2
From_Subject_ID_H_T...		OneToMany

5 Click **OK** to save and close the **Custom** tab.

6 Verify that SAS Customer Intelligence users (including the users of the saswbadm user account) have both READ and WRITE access to the physical location of the MAMeta library.

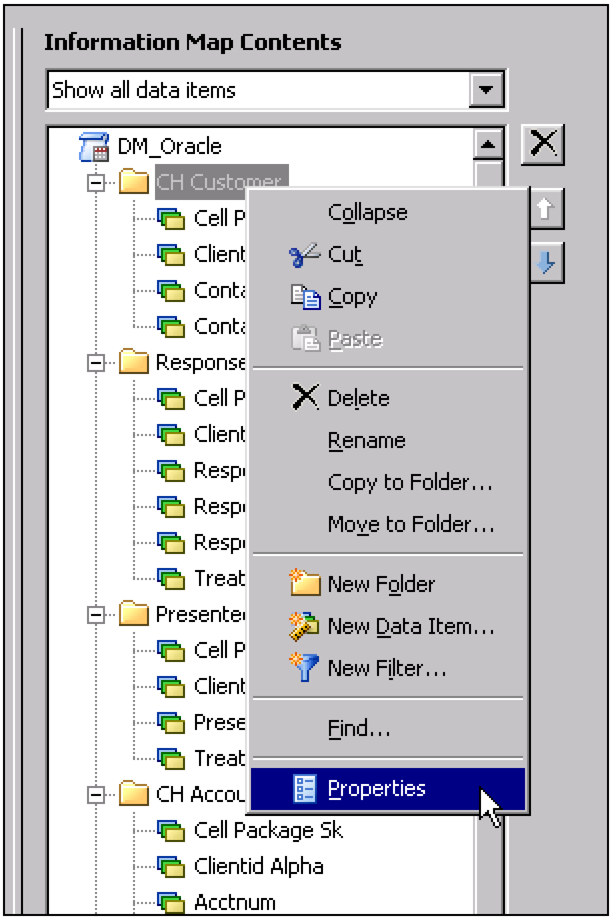
Add Custom Properties to an Information Map: Steps

To add custom properties to a SAS Information Map at the map level, folder level, or data item level, follow these steps:

1 Open the information map in SAS Information Map Studio.

2 In the Information Map Contents pane, right-click the information map, folder, or data item, and select **Properties**.

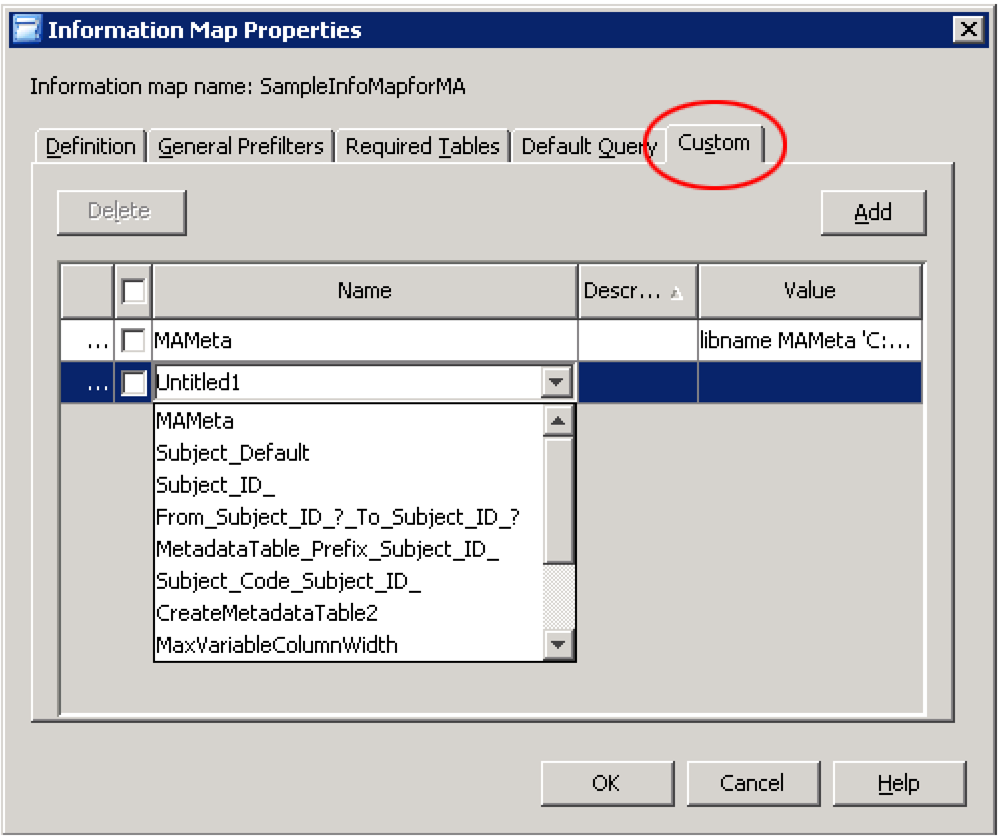
Figure 5.3 Folder Properties



3 Select the **Custom** tab.

4 If an Untitled1 row is not already displayed, then click **Add** to create a new row. Otherwise, click in the right side of the Untitled1 cell to view the drop-down list of custom properties that are available, and select a custom property.

Figure 5.4 Custom Properties



Information map name: SampleInfoMapforMA

	□	Name	Descr...	Value
...	□	MAMeta		libname MAMeta 'C:...
...	□	Untitled1		

MAMeta
Subject_Default
Subject_ID_
From_Subject_ID_?_To_Subject_ID_?
MetadataTable_Prefix_Subject_ID_
Subject_Code_Subject_ID_
CreateMetadataTable2
MaxVariableColumnWidth

5 Complete any partial data items that you select. To enter text in a cell, double-click in the cell. For more information, see "Custom Properties (Map Level)" on page 87.

6 After you enter the name, description, and value for a new custom property, click within a header cell to remove the highlighting from the new row. If you click **OK** while any part of a row is highlighted, then the changes in that row are not saved.

7 Add the custom properties that are required for the map level. For more information, see "Custom Properties (Map Level)" on page 87.

8 To exit and save the new custom properties, click a heading cell or an existing row to remove the highlighting from any attribute, and then click **OK**.

Making Custom Properties Available

Overview of the MAtemplate.txt File

The MAtemplate.txt file enables you to specify the valid values for properties for an information map, its folders, and its data items. These are all elements that might be used in an information map that is created for SAS Customer Intelligence users.

When you add or edit a custom property in SAS Information Map Studio, SAS Information Map Studio displays a drop-down list of items that is read from MAtemplate.txt.

MAtemplate.txt is typically activated during the installation process. If the customer properties in MAtemplate.txt have not been loaded into the information map properties, then you still need to activate the MAtemplate.txt file by making custom properties available.

Contents of the Default MAtemplate.txt File

These are the contents of the default MAtemplate.txt file.

```
<ExtendedAttributesTemplate>
  <InformationMap EditablePicklist="True">
    <Item Key="MAMeta" EditablePicklist="True">
      <Option Value="Libname MAMeta "/>
    </Item>
    <Item Key="Subject_Default" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </Item>
    <Item Key="Subject_ID_"/>
    <Item Key="From_Subject_ID_?_To_Subject_ID_?" EditablePicklist="False">
      <Option Value="OneToOne"/>
      <Option Value="OneToMany"/>
      <Option Value="ManyToOne"/>
      <Option Value="ManyToMany"/>
    </Item>
    <Item Key="MetadataTable_Prefix_Subject_ID_"/>
    <Item Key="Subject_Code_Subject_ID_"/>
    <OptionalItem Key="CreateMetadataTable2" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="MaxVariableColumnWidth" DefaultValue="40"/>
    <OptionalItem Key="MaxValueColumnWidth" DefaultValue="200"/>
    <OptionalItem Key="Verbose" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="DiscreteSql" DefaultValue="SP" EditablePicklist="False">
      <Option Value="IQ"/>
      <Option Value="SP"/>
    </OptionalItem>
    <OptionalItem Key="Metadata" DefaultValue="Counts" EditablePicklist="False">
      <Option Value="Counts"/>
      <Option Value="Values"/>
      <Option Value="None"/>
    </OptionalItem>
    <OptionalItem key="MaxItemsForInQuery" DefaultValue="1000"/>
    <OptionalItem key="FilteredTableExport" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem key="ForbidProblemUploads" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem key="SPRefinementExport" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem key="ExtraBlankQuery" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
  </InformationMap>
  <Folders EditablePicklist="True">
    <Item Key="Subject_ID_">
      <Option Value="Subject_ID_"/>
    </Item>
    <OptionalItem Key="Histogram_NumBins" DefaultValue="64"/>
    <OptionalItem Key="Histogram_DisplayNumBins" DefaultValue="16"/>
    <OptionalItem Key="Metadata" DefaultValue="Counts" EditablePicklist="False">
      <Option Value="Counts"/>
      <Option Value="Values"/>
      <Option Value="None"/>
    </OptionalItem>
  </Folders>
  <DataItems EditablePicklist="True">
    <Item Key="Level" EditablePicklist="False">
      <Option Value="Id"/>
      <Option Value="Nominal"/>
      <Option Value="Interval"/>
      <Option Value="Binary"/>
      <Option Value="Ordinal"/>
      <Option Value="Unary"/>
    </Item>
        <OptionalItem Key="UseInCluster" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="UseInSubjectId" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="UseInSubjectIdTop" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="IsFilterItem" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="Precision"/>
    <OptionalItem Key="VarUsedWithFilter"/>
    <OptionalItem Key="Histogram_NumBins" DefaultValue="16"/>
    <OptionalItem Key="Histogram_DisplayNumBins" DefaultValue="16"/>
    <OptionalItem Key="Histogram_Start"/>
    <OptionalItem Key="Histogram_Increment"/>
    <OptionalItem Key="Contact_History" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="Responses" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="Presented_Treatments" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="NoMetadata" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="AllowBlankValue" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
    <OptionalItem Key="OutputColumnName"/>
    <OptionalItem Key="Metadata" DefaultValue="Counts" EditablePicklist="False">
      <Option Value="Counts"/>
      <Option Value="Values"/>
      <Option Value="None"/>
    </OptionalItem>
    <OptionalItem Key="UseInOptimization" DefaultValue="Subject_ID_" EditablePicklist="True">
      <Option Value="Subject_ID_"/>
    </OptionalItem>
    <OptionalItem Key="Identifier"/>
    <OptionalItem Key="ExportIgnoresRefinement" DefaultValue="False" EditablePicklist="False">
      <Option Value="True"/>
      <Option Value="False"/>
    </OptionalItem>
  </DataItems>
  <FilterItems EditablePicklist="True">
  </FilterItems>
  <Relationships EditablePicklist="True">
  </Relationships>
</ExtendedAttributesTemplate>
```

Make Custom Properties Available: Steps

1 In SAS Information Map Studio, select **Tools** ▸ **Options**.

2 On the **Advanced** tab of the Options dialog box, select the **Custom properties at start-up** check box.

3 Specify the MAtemplate.txt file. The file is installed on the SAS server tier.

   On UNIX, the typical location is `/local/install/SASServer/SASHome/SASFoundation/9.4/misc/ma`.

   On Windows, the typical location is `\local\install\SASServer\SASHome\SASFoundation\9.4\ma\sasmisc`.

   The location might be different for your site.

4 Click **OK** to close the Options dialog box.

5 Restart SAS Information Map Studio to apply the new setting.

Verify That Custom Properties Are Available

To verify that the custom properties for editing an information map are available in SAS Information Map Studio, follow these steps:

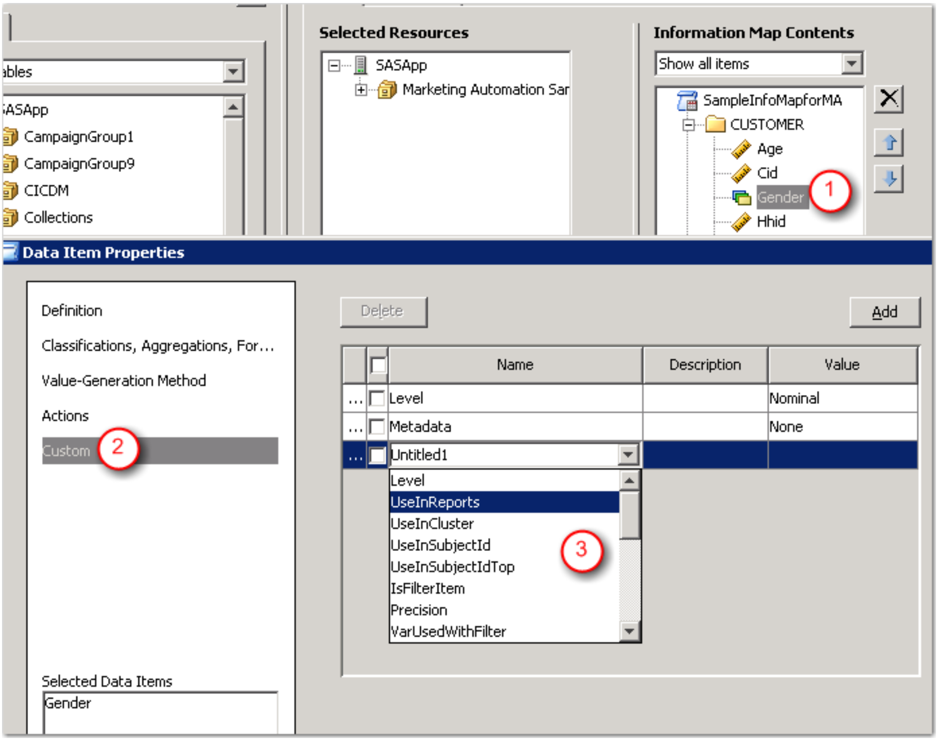
1 In SAS Information Map Studio, open an information map for SAS Customer Intelligence.

2 On the **Design** tab, in the Information Map Contents pane, double-click an information map, folder, or specific data item. **Gender** (label 1) is the selected data item.

3 In the Information Map Properties dialog box that appears, if the **Custom** option (label 2) is populated with custom properties, then MAtemplate.txt is successfully activated.

4 Click in a row beneath the Name column. The drop-down list (label 3) contains items that SAS Information Map Studio reads from MAtemplate.txt.

Figure 5.5 Verify Custom Properties



Customize Subject_ID in MAtemplate.txt

The following example demonstrates how you modify MAtemplate.txt in order to specify the list of subjects on the **Custom** tab in SAS Information Map Studio.

Any modification that you make to MAtemplate.txt are overwritten by an upgrade of SAS Customer Intelligence software. Therefore, if you want to retain your modified MAtemplate.txt file, then save a copy of it in a safe location in order to replace the new default template file.

Note: If you modify a custom property in MAtemplate.txt, the change does not affect any previously created information maps. The change affects only the new information maps that are created for SAS Customer Intelligence. If you specify a new value for an existing custom property in MAtemplate.txt, then the new value is displayed in the list of possible values for the custom property.

Example: Adding Custom Subjects to MAtemplate.txt

This example produces a list of values that the user can select.

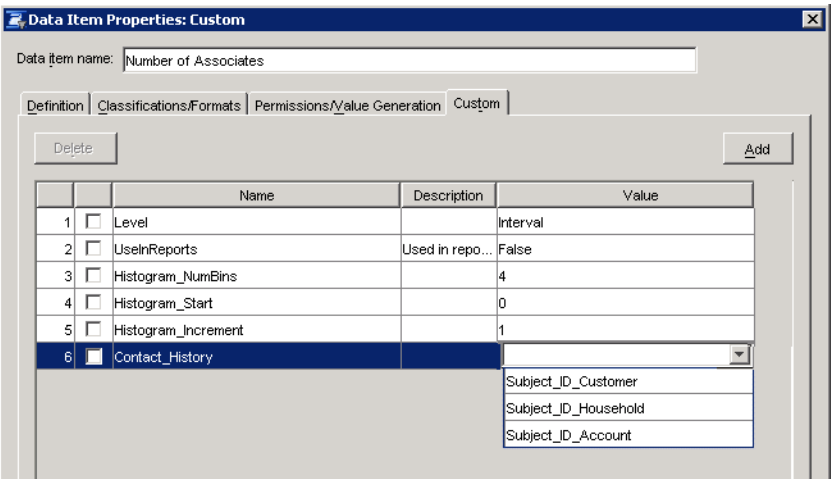
Search for instances of the following code in MAtemplate.txt.

```
<Option Value="Subject_ID_">
```

Replace each instance of the code in MAtemplate.txt with the following code:

```
<Option Value="Subject_ID_Customer"/>
<Option Value="Subject_ID_Household"/>
<Option Value="Subject_ID_Account"/>
```

Figure 5.6 *Custom Subjects*



Data item name: Number of Associates

		Name	Description	Value
1	□	Level		Interval
2	□	UseInReports	Used in repo...	False
3	□	Histogram_NumBins		4
4	□	Histogram_Start		0
5	□	Histogram_Increment		1
6	□	Contact_History		Subject_ID_Customer Subject_ID_Household Subject_ID_Account

Verify that the list of subjects is correctly displayed. For more information, see "Verify That Custom Properties Are Available" on page 85.

Custom Properties (Map Level)

In the example values in the following table, the two subjects are Customer and Household.

The extended attributes for a SAS Customer Intelligence information map that are related to subject are in the following table.

Table 5.1 *Custom Properties (Map Level) That Are Related to Subject*

Label	Description	Example Value	Required?	Default Value
FilteredTableExport	affects the refinement of export behavior. If FilteredTableExport is set to True, then when a subject is specified to be exported, at least one line for every subject is exported.	True	no	-
DataSetOptions_ *engine*	sets the data set options for the specified engine. This setting applies to all of the tables that access data through that engine.	DataSetOptions_ ORACLE=orhints='/* */'	no	-

From_Subject_ID_ *subjectname1_*
To_Subject_ID_ *subjectname2*
	is defined for each relationship between the valid subjects on the source tables. An entry is necessary for each direction of the relationship. These are valid relationships:
	■ OneToOne
	■ OneToMany
	■ ManyToOne
	■ ManyToMany*
	* not recommended because if the cells or lists of customers are created, then any duplicates are deleted.
		From_Subject_ID_Customer_To_Subject_ID_Household = ManyToOne
		From_Subject_ID_Household_To_Subject_ID_Customer = OneToMany
			yes	-

MetadataTable_Prefix_ Subject_ID_*subjectname* (for example, MetadataTable_Prefix_ Subject_ID_Customer)	is the prefix name for the SAS data sets that contain the generated data (the counts metadata). This is also the prefix name of data items for which NoMetadata extended attribute is set to False. Only one extended attribute per subject should exist.	Customer	yes	-
Subject_Code_Subject_ID__ *subjectname*	associates a two-digit code with a subject. Only one extended attribute per subject should exist. This code is encoded as part of the ResponseTrackingCode that is a mandatory column to be passed to SAS Customer Intelligence Common Services. This code enables SAS Customer Intelligence Common Services to determine which subject table to write the contact responses and the presented treatment rows to.	Subject_Code_Subject_ID__ Customer = 01	yes	-
Subject_Default	is the default subject extended attribute to be used throughout the information map for any data items where an associated subject extended attribute has not been explicitly set.	Subject_ID_Customer	yes	-

Subject_ID_*subjectname* (for example, Subject_ID_Customer)	is established for each subject as a unique extended attribute. Only one such extended attribute exists per subject. This *subjectname* is used in many places for other extended attribute values. The value appears in the **Subject** drop-down list in the Select Data Item dialog box in SAS Customer Intelligence. The actual physical data field name that makes up this subject is defined at the data item level.	Customer	yes	-

Custom Properties (Folder Level)

Table 5.2 Custom Properties (Folder Level) That Are Related to Subject

Label	Description	Example Value	Required?	Default Value
Subject_ID_*x* (for example, Subject_ID_1	is a shortcut that defines the subject attribute of this data item as this value for all of the data items in this folder, unless otherwise specified. Each Subject_ID_*x* label defines a valid subject for selection rules when a user selects data items in this folder. The value must correspond to a subject extended attribute name that was defined at the information map level. More than one of these extended attributes can be defined in a folder, by incrementing the number. In a diagram node, the user chooses a category (folder) and a subject. Then the user selects the data items in the corresponding folder that has the chosen subject as a value of one of these Subject_ID_*x* extended attributes at either the folder level or data item level. If Subject_ID_1 is not set for either a given data item or parent folder, then the subject default value that is set at the Information Map level is assumed. For example, if for the data item Account Balance), Subject_ID_1 = Subject_ID_Customer and Subject_ID_2 = Subject_ID_Account, then Account Balance can be selected only if the user's chosen subject is either Customer or Account.	Subject_ID _Customer	no	-

Custom Properties (Data Item Level)

Table 5.3 Custom Properties (Data Item Level) That Are Related to Subject

Label	Description	Example Value	Required?	Default Value
Contact_History	identifies the table in which contact history records are added for the specified subject. The contact history property is used by SAS Customer Intelligence in order to direct new records to the contact history table. The underlying physical column name or names of the data items that have UseInSubjectID equal to this value are the physical column name or names of the subject on Contact History. This attribute should be set once per subject.	Subject_ID_Customer	yes for one data item derived from each contact history table	-
ExportIgnoresRefinement	affects the refinement of export behavior. If ExportIgnoresRefinement is set to True for a data item, then the data item ignores any refinement that has been added. This extended attribute is useful for aggregated items.	True	No	-
Presented_Treatments	specifies that, for the specified subject, the presented treatment is stored in the table that contains this data item. This attribute should be set once per subject.	Subject_ID_Customer	yes for one data item	-
Responses	identifies the table in which the responses are stored for the specified subject. The Responses property is used by SAS Customer Intelligence in order to fetch the appropriate response records from the response table for the response node. Also, the underlying physical column names of the data items that have UseInSubjectID equal to this value are the physical column name or names of the subject on responses. This attribute should be set once per subject.	Subject_ID_Customer	yes for one data item derived from each response table	-

UseInSubjectID	is set only for the data items that are derived from primary keys of a primary subject table. The value that is specified here is the Subject_ID_*x* extended attribute value that is assigned for this subject table at the information map level. For example, if this extended attribute is on data item Customer ID whose underlying physical column name is CUSTTABLE.CUST_ID, then this extended attribute defines the physical column as CUSTTABLE.CUST_ID for subject Subject_ID_Customer. If two physical fields make up a single subject, then a UseInSubjectID extended attribute exists for both of the data items that are associated with those fields. Only one (or one set) of these attributes for each subject that is represented in the information map should exist. It does not matter which folder the data item with this attribute is in; it is the physical subject column or columns that are derived for all data items in the information map that is associated with that subject. The field name and type are used to generate the holding table when the selections are made at this subject level.	Subject_ID _Customer	yes for data items from the primary key of the subject tables	-

All Custom Properties for Information Maps

The extended attributes in an information map describe settings for the following levels:

Map

The custom properties (for information maps) typically describe the metadata and define the subjects in an information map. Subject properties specify either a default subject or valid subjects for selection rules, and specify the relationship between valid subjects of selection rules. Metadata properties specify the location for the metadata tables, the names for tables, and the metadata column widths.

Folder

The custom properties for folders specify valid subjects for selection rules, and specify the details about histograms.

Data Item

The custom properties for data items define the supported data types, the creation of primary keys and multi-keys, and the location of contact history and response tables for each subject.

Each of the following tables includes all of the extended attributes for information maps at the map level, folder level, and data item level, respectively.

For information about using extended attributes to configure BULKLOAD options in the information map, see “Using Multiple Database Engines and Database Servers” on page 215.

Table 5.4 Map Level: All Custom Properties

Label	Description	Example Value	Require d?	Default Value
CIBusinessContext	This attribute is needed for Visual Selection integration with Web Report Studio. It is added to the Information Map to be used in the Web Report Studio Report. The user of the information map must be assigned the Web Report Studio Advanced Users role. The value is the name of the business context where the report export is available. Report exports are the SAS data sets or criteria that are created by Web Report Studio. Only one of these extended attributes can be set per information map.	BC_Name	no	-
Count(*)	Use this attribute to specify whether to use a select count(*) to get a count for the query. The default value uses select count(*) to obtain a row count for the query instead of using a JDBC method to obtain a row count. **Caution**This attribute is designed to be used only for performance testing, and should not be used in the production environment.	True False	no	True
CreateMetadataTable2	If this attribute is set to True, then the suffix 2 is added to each generated table name. The tables are created in the library that is defined by the MAMeta library reference.	True False	no	False

DiscreteSql	is used to specify the SQL method to use in order to generate discrete (NUM and CHAR) metadata.	DiscreteSql = [SP]
		SP: use a stored process
		DiscreteSql = [IQ]
		IQ: use Iquery (JDBC)
			no	SP

FilteredTableExport	affects the refinement of export behavior. If FilteredTableExport is set to True, then when a subject is specified to be exported, at least one line for every subject is exported.	True	no	-

ForbidProblemUploads	identifies problematic queries and does not pass the problematic query to the database. SAS performs the query locally. If your tables are large, this setting can cause additional performance issues. This custom property is especially useful when using Netezza	True	no	False
From_Subject_ID_ *subjectname1*_ To_Subject_ID_ *subjectname2*	is defined for each relationship between the valid subjects on the source tables. An entry is necessary for each direction of the relationship. These are valid relationships: ■ OneToOne ■ OneToMany ■ ManyToOne ■ ManyToMany* * not recommended for this reason: if the cells or lists of customers are created, then any duplicates are deleted.	From_Subject_ID_Customer_ To_Subject_ID_ Household= ManyToOne From_Subject_ID_Household_ To_Subject_ID_ Customer= OneToMany	yes	-
IDListSeparator	overwrites the default separator (<>) for the IDList parameter that is passed into the stored process in the case where the data item ID also contains this pattern. This value is used by the metacount stored process.	—- ++++ ****	no	<>
MAMeta	specifies a libref for the location of the metadata tables.		yes	-

MaxItemsForInQuery	converts ITEM IN(...) queries with > N items:
	SELECT ITEM IN(...< N) UNION
	SELECT ITEM IN(...< N)
	Use this attribute to override the specification that is imposed by a database (such as Oracle) that limits the number of items in a list.
		1000 (for Oracle)	no	0

MaxProcSummary	specifies discrete (NUM and CHAR) variable metadata. MAxProcSummary specifies the maximum number of variables to process within one execution of PROC SUMMARY. By reducing the number of variables, you can reduce memory requirements.	20	no	50

MaxValueColumnWidth	a numeric attribute that defines the width of the metadata columns Value and FormattedValue in each metadata table.	100	no	200
MaxVariableColumnWidth	a numeric attribute that defines the width of the metadata column Variable in each metadata table.	100	no	40
Metadata	Metadata = [Values | Counts | None] Values: distinct values only. Counts: default (distinct values and counts). None: equivalent to NoMetadata=True. The Values argument is for discrete variables only. The Metadata extended attribute can be propagated from upper level to lower level. For example, if you specify Metadata = x at the map level, then all of the folders inherit this setting. An exception is when a folder also has its own Metadata = y attribute that overrides the x setting.	Values Counts None	no	Counts
MetadataTable_Prefix_ Subject_ID_*subjectname* (for example, MetadataTable_Prefix_Subject_ID_Customer)	specifies the prefix name for the SAS data sets that contain the generated data and for the data items that do not have the NoMetadata extended attribute set to False. Only one extended attribute per subject should exist.	Customer	yes	-
Subject_Code_Subject_ID__ *subjectname*	associates a two-digit code with a subject. Only one extended attribute per subject should exist. For example, Subject_Code_Subject_ID_Customer=01 Subject_Code_Subject_ID_Account=02	01	yes	-
Subject_Default	is the default subject extended attribute to be used throughout the information map for any data items where an associated subject extended attribute has not been explicitly set.	Subject_ID_Customer	yes	-

Subject_ID_*subjectname* (for example, Subject_ID_Customer)	is a unique extended attribute that exists for each subject. The *subjectname* is used in many places for other extended attribute values. The value appears in the **Subject** drop-down list in the **Select Data Item** dialog box in SAS Customer Intelligence. The actual physical data field name or names that make up this subject are defined at the data item level.	Customer	yes	-
Verbose	If this extended attribute is set to True, then the extended attributes for the map, folder, and data items are written to the log files of the middle tier.	True False	no	False

Table 5.5 *Folder Level: All Custom Properties*

Label	**Description**	**Example Value**	**Required?**	**Default Value**
Histogram_Display-NumBins	defines the number of intervals used in the histogram in the Select Data Item dialog box in SAS Customer Intelligence. Unless specified at the data item level, this value applies to all Interval level data items within the folder.	8	no	If Histogram _NumBins is 64, then the default value is 16.
Histogram_NumBins	defines the number of bins to be used by the histogram stored process in order to support the histogram display of interval data items. Interval data items are specified in the Select Data Item dialog box in SAS Customer Intelligence. The value for this attribute is required to be an even power of 2. For example, 16 is an acceptable value because 16 is equal to 2 to the 4th power. This value applies to all of the interval level data items within the folder, unless the value is specified at the data item level.	32	no	16

Metadata	Metadata = [Values | Counts | None] Values: distinct values (discrete variables only). Counts: default (distinct values and counts). None: equivalent to NoMetadata=True. This extended attribute can be propagated from upper level to lower level. For example, if you specify Metadata = Counts at the map level, then all of the folders inherit this setting. An exception is when an attribute such as Values for a folder has been specified. In this example, the folder attribute (Values) overrides the setting of the map level (Counts).	Values Counts None	no	Counts
Subject_ID_*x* (for example, Subject_ID_1)	is a shortcut to define that for all of the data items in this folder, unless otherwise specified, the data item’s subject attribute is this value. Each Subject_ID_*x* defines a valid subject for selection rules when a user selects data items in this folder. The value must correspond to a subject extended attribute name that is defined at the information map level. More than one of these extended attributes can be defined in a folder, by incrementing the number. In a diagram node, the user chooses a category (folder) and a subject. Then the user selects the corresponding folder that has the chosen subject as a value of one of these Subject_ID_*x* extended attributes at either the folder level or data item level. If Subject_ID_1 is not set for either a given data item or the parent folder, then the Subject Default that is set at the Information Map level is assumed. For example, if Subject_ID_1=Subject_ID_Customer and Subject_ID_2=Subject_ID_Account are specified for the data item Account Balance, then Account Balance can be selected only if the user’s chosen subject is either Customer or Account.	Subject_ID_Customer	no	-

Table 5.6 *Data Item Level: All Custom Properties*

Label	**Description**	**Example Value**	**Required ?**	**Default Value**
AllowBlankValue	specifies that a blank value is allowed to be used as a distinct value when this attribute is set to true. Set this attribute to True for a Teradata database (or any database that is not configured to allow blank values).	True False	no	False

Contact_History	identifies the table in which contact history records are added for the specified subject. The contact history property is used by SAS Customer Intelligence in order to direct new records to the contact history table. The underlying physical column name or names of the data items that have UseInSubjectID equal to this value are the physical column name or names of the subject on Contact History. This attribute should be set once per subject.	Subject_ID_ Customer	yes for 1 data item derived from each contact history table	-
ExportIgnoresRefinement	affects the refinement of export behavior. If ExportIgnoresRefinement is set to True for a data item, then the data item ignores any refinement that has been added. This extended attribute is useful for aggregated items.	True	No	-
Histogram_DisplayNumBins	defines the number of intervals that are used in the histogram in the Select Data Item dialog box in SAS Customer Intelligence. Unless specified at the data item level, this value applies to all interval level data items within the folder.	8	no	If Histogram_NumBins is 64, then the default value is 16.
Histogram_Increment	specifies the increment value for the histogram stored process.	1	no	-
Histogram_NumBins	defines the number of bins to be used by the histogram stored process in order to support the histogram display of interval data items. Interval data items are specified in the Select Data Item dialog box in SAS Customer Intelligence. The value for this attribute is required to be an even power of 2. For example, 16 is an acceptable value because 16 is equal to 2 to the 4th power. This value applies to all of the interval level data items within the folder, unless the value is specified at the data item level.	32	no	16
Histogram_Start	specifies the start value for the histogram stored process.	1	no	-
Identifier	specifies a string that is entered by a user.	-	no	-
IsFilterItem	specifies whether the corresponding data item is a filter item that is applied in the WHERE clause.	True False	no	False

Level	specifies the level of value to be assigned to the data item. You can assign the following levels to a data item: ■ ID ■ Unary ■ Binary ■ Nominal ■ Interval ■ Ordinal Data items that are set to Level=ID are not used in SAS Customer Intelligence. See Table 5.7 on page 100 for the definition of each valid level of data item. See Figure 5.7 on page 102 for valid combinations of Level with data type.	Nominal	yes	-
Metadata	Metadata = [Values | Counts | None] Values: distinct values only. Counts: default (distinct values and counts). None: equivalent to NoMetadata=True. Values is for discrete variables only. The Metadata extended attribute can be propagated from upper level to lower level. For example, if you specify Metadata = x at the map level, then all of the folders inherit this setting. An exception is when a folder also has its own Metadata = y attribute that overrides the x setting.	Values Counts None	no	Counts
NoMetadata	specifies whether a discrete list of values should be generated in the metadata. If a data item whose level is nominal has many distinct values (for example, 1,000) to be generated for the metadata tables, then performance degrades. If you use NoMetadata, then the user must specify values manually in order to use the data item for selections. Values are case sensitive and must be enclosed in quotation marks for data items that are specified by character table columns. To prevent poor performance for nominal data items that have too many unique values, specify NoMetadata=True.	True	no	False
OutputColumnName	specifies the column name that replaces the physical column name that is used in the internally generated SQL query when this data item is selected for processing.	Any valid column name.	no	-
Precision	specifies the precision for the interval variable	100	no	1

Presented_Treatments	specifies that the table containing this data item is where the presented treatment is stored for the specified subject. This attribute should be set once per Subject_ID.	Subject_ID_Customer	yes for 1 data item from each presented treatment table	-
Responses	identifies the table in which the responses are stored for the specified subject. The Responses property is used by SAS Customer Intelligence to fetch the appropriate response records from the response table. The underlying physical column name or names of the data items that have UseInSubjectID equal to this value are the physical column name or names of the subject on responses. This attribute should be set once per subject.	Subject_ID_Customer	yes for one data item derived from each response table	-
UnivariateMethod	generates metadata for a data item in a univariate table. The default is the default SAS behavior. These are the valid values: InDatabase: uses the MEAN function to calculate the values Summary: uses PROC SUMMARY to calculate the values. Specify this value to avoid overflow errors when using PROC SQL to generate histogram metadata. AVGMinMax: calculates the average of the minimum and maximum values	InDatabase	no	default SAS behavior
UseInCluster	specifies whether this data item is used in the Cluster node. These are the valid values: ■ True ■ False Nominal and ordinal data items are clustered separately from interval data items. If UseInCluster is set to True in any nominal or ordinal data items, then at least three nominal or ordinal data items with the UseInCluster attribute set to True must exist. At least three interval data items with UseInCluster set to True must exist if any interval data items have UseInCluster set to True. To improve performance, do not cluster any nominal or ordinal data items with many values. Limit the data items that you cluster to those data items that are the most meaningful to cluster.	True	no	False

UseInOptimization	specifies whether the data item is used in an optimization of the subject ID. The valid value is the subject ID. **Note:** Data items that are passed to SAS Marketing Optimization have a limit of 32 characters.	Subject_ID_Customer	no	-
UseInSubjectIDTop	is the same as UseInSubjectID, except that it also specifies the order of this data item to be placed ahead of the other multi-part keys.	Subject_ID_Arrangement	no	-
UseInSubjectID	is set only for the data items that are derived from primary keys of a primary subject table. The value that is specified here is the Subject_ID_*x* extended attribute value that is assigned for this subject table at the information map level. For example, if this extended attribute is on data item Customer ID whose underlying physical column name is CUSTTABLE.CUST_ID, then this extended attribute defines that for subject Subject_ID_Customer, the physical column is CUSTTABLE.CUST_ID. If two physical fields make up a single subject, then a UseInSubjectID extended attribute would exist for both of the data items that are associated with those fields. Only one (or one set) of these attributes for each subject that is represented in the information map should exist. It does not matter which folder the data item with this attribute is in; it is the physical subject column or columns that are derived for all data items in the information map that is associated with that subject. The field name and type are used to generate the holding table when the selections are made at this subject level.	Subject_ID_Customer	yes for data items from the primary key of the subject tables	-
VarUsedWithFilter	specifies the data item name to be used together with a filter item. Only a Numeric Interval data item is a valid entry.	Customer Model Score No	no	-
Visible	If Visible is set to False for a data item, then the unfiltered targets of a filtered data item are hidden and metadata generation is faster. These hidden items can still be referenced as target items for filtered data items.	True False	no	True

Definitions of the Levels of Data Items

You can assign a data item to one of the following levels:

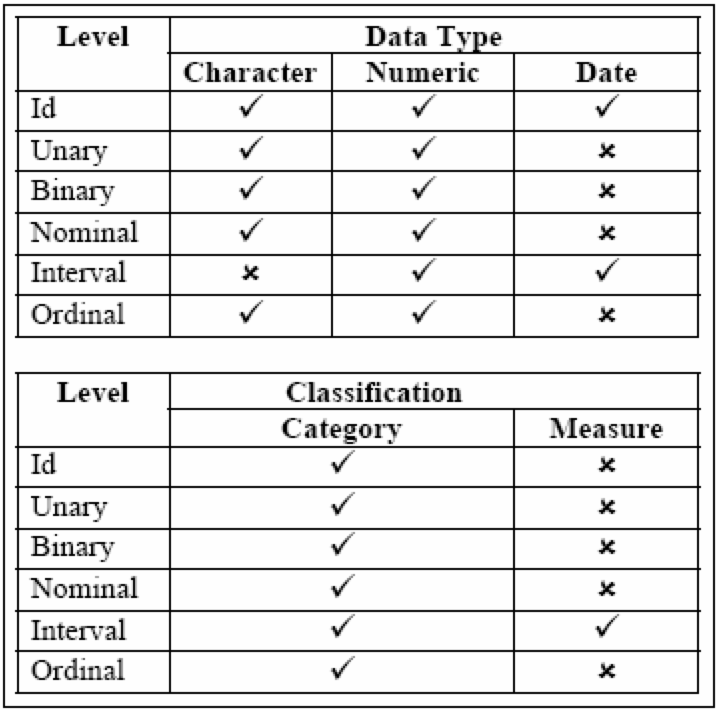
Table 5.7 Definitions of the Levels of Data Items

Level Value	Definition	Character	Numeric	Date

ID	identifies the data, but is not used in analysis.	yes	yes	yes
Unary	has one distinct value, including missing.	yes	yes	no
Binary	has two distinct values, including missing.	yes	yes	no
Nominal	has distinct values, but has no order (such as gender or marital status), including missing.	yes	yes	no
Interval	has many values, including infinite.	no	yes	yes
Ordinal	has distinct ordered values, such as income brackets, age groups, or scores.	yes	yes	no

The following display provides the valid combinations of the Level attribute with data type. **Classification** refers to how the data item is classified in SAS Information Map Studio.

Figure 5.7 Valid Data Type and Level Combinations



Level	Data Type		
	Character	Numeric	Date
Id	✓	✓	✓
Unary	✓	✓	✗
Binary	✓	✓	✗
Nominal	✓	✓	✗
Interval	✗	✓	✓
Ordinal	✓	✓	✗

Level	Classification	
	Category	Measure
Id	✓	✗
Unary	✓	✗
Binary	✓	✗
Nominal	✓	✗
Interval	✓	✓
Ordinal	✓	✗

Understanding the Counts Metadata

What Is the Counts Metadata?

The *counts metadata* is metadata that is specifically generated for SAS Customer Intelligence views, and is distinct from the SAS metadata in the SAS Metadata Repository. The counts metadata is typically a list of values and a count of how often each value occurs in the database. Occasionally the counts metadata consists of a list of how many values fall into each of a set number of sub-intervals. The sub-intervals depend on the type of level that is selected for a data item.

A counts metadata value is read from a library reference (libref) of an information map. A counts metadata value is displayed within the nodes of a SAS Customer Intelligence diagram.

Descriptions of the Counts Metadata Tables

The counts metadata is stored in a series of SAS tables stored in the location that is defined by the MAMeta extended attribute at the map level. These are the metadata tables:

subject charcounts
is a list of character data items and the associated counts by subject.

*subject*numcounts	is a list of numeric data items and the associated counts by subject.
subject histocounts	contains information that is used to generate histograms for numeric data items by subject.
subject datehistocounts	information that is used to generate histograms for the date data items by subject.
univariatestats	min, max, mean, and standard deviation for interval variables. This attribute must be updated before you update either *subject*histocounts or *subject*datehistocounts.
physicalmapping	variable and physical reference data that is based on the information map. Must be updated before updating any of the other tables in this list.

For the first four metadata tables in the previous list, one table per subject exists. Metadata is generated only for items whose subject is set at the folder level. For example, for a folder whose Subject_ID_*x* is not specified for the household subject, no metadata is generated for any data item in the folder at the household level.

Generating the Counts Metadata

Overview of Generating the Counts Metadata

You generate the counts metadata in SAS Customer Intelligence. For more information, see *SAS Marketing Automation: User’s Guide*.

When Should the Counts Metadata Be Generated?

Update the metadata cache only when the system does not have any active users. Generate or update the counts metadata in any of the following circumstances:

- The underlying data changes to ensure that all the possible values are available for use.
- A new folder is added to the information map.
- A new subject is added to a folder.
- New data items are added to the information map.
- Data item attributes that require support by the metadata are changed.

Update the Counts Metadata

You can manually update the counts metadata tables by using the custom property **CreateMetadataTable2** on the information map level. When you set this custom property to `True`, the generated metadata tables (from the underlying information map) are renamed by appending a 2 to the table names.

For example, customercharcounts.sas7bdat is renamed customercharcounts2.sas7bdat. In order to update the old metadata tables, replace the contents of the original tables with the files that are appended with a 2, and then clear the metadata cache. For more information, see "When to Enable the Clear Metadata Cache Option" on page 104.

When to Use PROC SQL to Generate Metadata

If the information map is not complex, but the underlying data is very large, then generate the metadata by using the SAS SQL procedure (PROC SQL). In the PROC SQL method, an SQL query is streamed to the SAS procedure as XML. The XMLgenerates the counts for discrete variables and the bin ranges for the histogram variables.

Accessing the Tables of the Counts Metadata

If an information map has three subjects (such as household, customer, and account), the resulting metadata is contained in these tables:

- accountcharcounts
- accountdatehistocounts
- accounthistocounts
- customercharchounts
- customerdatehistocounts
- customerhistocounts
- customernumcounts
- householddatehistocounts
- householdhistocounts
- householdnumcounts
- physicalmapping
- univariatestats

The data that is contained in these tables can be accessed like any other SAS table. For example, you can use the following SAS code to check the data item variable:

```
LIBNAME MAMETA '\\machinename\SAS\MA\Data\MAMeta';
proc sql;
      select * from MAMeta.customercharcounts
      where Variable='Gender';
quit;
```

When to Enable the Clear Metadata Cache Option

In some environments, the counts are stored in a separate set of tables. The tables are copied into the SAS Customer Intelligence metadata library. The SAS Customer Intelligence engine, however, keeps a copy of those tables in memory. When you make changes to the metadata tables, the engine does not recognize and update it. Users do not see the new values of the counts metadata in the diagrams in SAS Customer Intelligence.

To enable SAS Customer Intelligence to recognize the new metadata values, clear the cache on the **Metadata** tab in the business context. For more information, see *SAS Marketing Automation: User’s Guide*.

6

Common Data Model: Concepts

About the Common Data Model

What Is the Common Data Model?

The SAS Customer Intelligence Common Data Model is a common data repository for SAS Customer Intelligence solutions, including SAS Marketing Automation and SAS Real-Time Decision Manager.

Some solutions use the common data model to store data such as campaigns that have been conducted, communications that have been run, offers that have been presented, and customer responses.

All of the business contexts for your site can share a single version of the common data model. Alternatively, you can specify a separate version of the model for each business context. For more information, see "Managing Multiple Business Contexts" on page 108.

For descriptions of the tables and their columns in the common data model, see *SAS Customer Intelligence Common Data Model: Data Dictionary* at http://support.sas.com/documentation/solutions/ci/index.html. You must supply the following user name and password to view this site:

```
User Name: sas
Password: CIadmin123
```

A diagram of the model is included at the end of the data dictionary. Use the diagram to understand the relationships among the tables that comprise the model.

Updating the Content of the Common Data Model

The common data model is typically updated when you save a campaign after the campaign has been published, or each time you execute a campaign. A campaign is not published if the publish operation is not enabled, and if the location of the data model for the respective business context is not specified.

When a campaign is executed once or in a series of scheduled executions, each occurrence is assigned a unique RESPONSE_TRACKING_CD. All occurrences are displayed in the CI_CONTACT_HISTORY table of the common data model. You manage occurrences on the **History** tab of the Execution page of a campaign. If you delete the most recent occurrences, the sequence of occurrence numbers begins with the last successfully executed occurrence that is listed in the Executions table.

Managing Multiple Business Contexts

In an environment where multiple business contexts exist, you can deploy the common data model and capture data across business contexts in two ways.

- Specify a single instance of the common data model. Specify that all of the business contexts point to the same reporting libref. The **BUSINESS_CONTEXT** field in the CI_CAMPAIGN table captures the business context that is associated with the campaign. In this pattern, all of the campaign data across the enterprise resides in the same location.
- Create a separate instance of the common data model for each business context (recommended). In this pattern, the data of a business context resides at different locations, and each instance of the data model contains only the information for a single business context.

Reporting and the Common Data Model

When campaign data is published to the common data model tables, users can view this data in reports.

How the Common Data Model Is Organized

Overview

The tables of the common data model can be organized conceptually in the following categories:

- Campaigns. For more information, see "Campaigns" on page 109.
- Campaign groups. For more information, see "Campaign Groups" on page 110.
- Cells, packages, and treatments. For more information, see "Cells, Packages, and Treatments" on page 112.
- Communications. For more information, see "Communications" on page 111.
- Control groups. For more information, see "Control Groups" on page 114.
- History and responses. For more information, see "History and Responses" on page 114.
- SAS Marketing Optimization. For more information, see "SAS Marketing Optimization" on page 115.
- Integration with SAS Customer Intelligence 360. For more information, see "Integration with SAS Customer Intelligence 360" on page 116.

To implement these tables, see Chapter 7, "Implementing the Common Data Model ," on page 129.

Campaigns

CI_CAMPAIGN

The CI_CAMPAIGN table is the central table to which the other tables in the campaigns category connect. A campaign contains a planned set of one or more communications over one or more channels that are directed at a group of subjects such as customer, household, or individual.

CI_CAMPAIGN_EXT

The CI_CAMPAIGN_EXT table contains the additional user-defined fields if the CI_CAMPAIGN table is extended in order to meet user-specific requirements. For more information, see "About Extension (_EXT) Tables" on page 116.

CI_CAMPAIGN_STATUS

The CI_CAMPAIGN_STATUS table is a reference table that contains the status codes and status descriptions for a campaign. This table is automatically populated when the common data model is installed. For default populated values, see "About Lookup Tables" on page 123.

CI_CAMPAIGN_TYPE

The CI_CAMPAIGN_TYPE table is a reference table that contains the types of campaigns. This table is automatically populated with the values Selection and Decision when the common data model is installed.

CI_CAMP_PAGE

The CI_CAMP_PAGE table contains the checklist item for a campaign. A checklist item can be a brief, a diagram, a communication, an optimization, a schedule, creative details, or offer details.

For information about _UDF tables, see "UDF (User-Defined Field) Tables" on page 117.

CI_CHANGE_LOG

The CI_CHANGE_LOG table contains an audit trail that is recorded when a component (table) of a campaign is changed.

CI_CHANGE_TYPE

The CI_CHANGE_TYPE table is a reference table that contains the types of changes that are stored in the change log. This table is prepopulated when the common data model is installed.

CI_TREATMENT_CAMPAIGN_SET

The CI_TREATMENT_CAMPAIGN_SET table is a set of treatment campaigns applicable only in Real-Time Decision Manager campaigns. After each treatment campaign has determined the eligibility of an offer, the campaign returns the information to the treatment campaign set. The treatment campaign set arbitrates the offers and returns the best offers to the individual customers.

CI_TREAT_CAMP_SET_STATUS

The CI_TREAT_CAMP_SET_STATUS table is a reference table that contains the status codes and descriptions for a treatment campaign set. This table is prepopulated when the common data model is installed. The supplied values are __N (Not Marked for Deployment) and __D (Marked for Deployment).

CI_TREAT_CAMP_SET_X_CAMPAIGN

The CI_TREAT_CAMP_SET_X_CAMPAIGN table is an intersection table that models the many-to-many relationship between treatments and campaigns. When a treatment campaign is removed from the treatment campaign set, a clean publish operation removes the treatment campaign from this table.

Campaign Groups

CI_CAMPAIGN_GROUP

The CI_CAMPAIGN_GROUP table contains the name, description, business contexts, and checklist items for a group of selection campaigns. The campaign group enables the campaign manager to assign all budgets and scores across campaigns, and to schedule all of the campaigns within the campaign group at the same time. Either SAS Marketing Automation or SAS Marketing Optimization can access data about campaign groups.

CI_CAMPAIGN_GROUP_EXT

The CI_CAMPAIGN_GROUP_EXT table contains the additional user-defined fields when the CI_CAMPAIGN_GROUP table must be extended in order to meet user-specific requirements. For more information, see “About Extension (_EXT) Tables” on page 116.

CI_CAMPAIGN_GROUP_STATUS

The CI_CAMPAIGN_GROUP_STATUS table is a reference table that contains the status codes and descriptions for a campaign group. This table is prepopulated when the common data model is installed. Additional user-defined status codes can be specified.

CI_CAMPAIGN_GROUP_TYPE

The CI_CAMPAIGN_GROUP_TYPE table is a reference table that contains the types of campaign groups. This table is prepopulated when the common data model is installed.

CI_CAMP_GRP_PAGE

The CI_CAMP_GRP_ PAGE table contains the checklist item for a campaign group. A checklist item can be a brief, a diagram, a communication, an optimization, a schedule, creative details or offer details.

For information about _UDF tables, see “UDF (User-Defined Field) Tables” on page 117.

CI_CAMP_GRP_X_MO_OPT_SCENARIO

The CI_CAMP_GRP_X_MO_OPT_SCENARIO table is an intersection table that models the many-to-many relationship between campaign groups and SAS Marketing Optimization scenarios.

Communications

CI_CHANNEL

The CI_CHANNEL table is a reference table. It contains the channels, channel codes, and channel descriptions that are typically used in a campaign. A channel represents the means of communication (such as email) between a company and targeted customers. This table is automatically populated with common channel codes. For more information, see “About Lookup Tables” on page 123.

CI_COMMUNICATION

The CI_COMMUNICATION table contains the general descriptive information about a communication. A communication represents the delivery of a package of treatments over a channel to particular cells.

CI_COMMUNICATION_EXT

The CI_COMMUNICATION_EXT table contains additional user-defined fields when the CI_COMMUNICATION table must be extended in order to meet user-specific requirements.For more information, see "About Extension (_EXT) Tables" on page 116.

CI_COMMUNICATION_RECURR_TYPE

The CI_COMMUNICATION_RECURR_TYPE table is a reference table that contains the types of communication recurrences. This table is prepopulated with the values None, Hourly, Daily, Weekly, Monthly, and Other, when the common data model is installed.

CI_COMMUNICATION_STATUS

The CI_COMMUNICATION_STATUS table is a reference table that contains the status codes and status code descriptions for a communication. For example, the status code _11 means Exported; the status code _12 means Deployed. This table is prepopulated when the common data model is installed. Additional user-defined status codes can be specified.For the default values, see "About Lookup Tables" on page 123.

_UDF Tables

For information about _UDF tables, see "UDF (User-Defined Field) Tables" on page 117.

Cells, Packages, and Treatments

CI_CELL_PACKAGE

The CI_CELL_PACKAGE table contains columns to associate cells, communications, and treatments with a package in order to facilitate reporting. The table also includes summary counts for contacts and responses at a package level.

The DELETED_FLG field is set to Y when an occurrence is re-executed. Both executions have the same occurrence number. The previous occurrence is deleted when a new occurrence is added. The DELETED_FLG field is not set to Y when the last occurrence is deleted. DELETED_FLG field is set to Y when the occurrence is re-executed and has a new surrogate key.

CI_DYNAMIC_TREATMENT_ATTRIBUTE

The CI_DYNAMIC_TREATMENT_ATTRIBUTE table contains the associations of what dynamic treatments are part of a campaign. In dynamic treatments, values can change each time that a campaign is run.

This table uses the response tracking code to generate a value for the CELL_PACKAGE_SK column. In a DS2 process, the contact must precede the writing to the table, or the value of CELL_PACKAGE_SK will be -1.

CI_DYNAMIC_TREATMENT_ATTR_EXT

The CI_DYNAMIC_TREATMENT_ATTR_EXT table contains additional user-defined fields when the CI_DYNAMIC_TREATMENT_ATTRIBUTE table must be extended in order to meet user-specific requirements.For more information, see "About Extension (_EXT) Tables" on page 116. For information about the limits on the number of characters in a variable, see "About Populating the _EXT Tables" on page 116.

CI_PACKAGE

The CI_PACKAGE table contains a collection of one or more treatments. A package is used as a collection point for one or more treatments. Treatments are delivered to the channel as part of a package that is associated with a campaign.

CI_PACKAGE_X_TREATMENT

The CI_PACKAGE_X_TREATMENT table is an intersection table that models the many-to-many relationship between packages and treatments. A package can have multiple treatments. A single treatment can be a part of multiple packages.

CI_STAGING_DATA

If the common data model is used to store staged treatments, the CI_STAGING_DATA table associates staged treatments with a business context.

CI_TREATMENT

The CI_TREATMENT table contains the treatments that are associated with a campaign. A treatment represents a marketing message and its content that are delivered over a channel. Treatments can be combined into a package.

When a business user defines a treatment, that user can specify whether the treatment should be static or dynamic. Static treatments are named because their values do not change as the treatment or treatments are applied throughout the execution of a campaign.

The dynamic attribute of a treatment in a decision campaign can be specified as either a variable or a manually defined value.

CI_TREATMENT_EXT

The CI_TREATMENT_EXT table contains additional user-defined fields when the CI_TREATMENT table must be extended in order to meet user-specific requirements.For more information, see "About Extension (_EXT) Tables" on page 116.

_UDF Tables

For information about _UDF tables, see "UDF (User-Defined Field) Tables" on page 117.

Control Groups

CI_CONTROL_GROUP_TYPE

The CI_CONTROL_GROUP_TYPE table is a reference table that contains the types of control groups that are used for reporting. The supplied type is _ST for Standard. This table is prepopulated when the common data model is installed. Additional user-defined types can be specified.For more information, see Table 6.12 on page 127.

In the common data model, these associations are established.

- For champion/challenger settings, the champion is the control group that the challengers are compared with.
- For challenger/challenger settings, each challenger is a control group for all the other challengers in a node, so that any two challengers can be compared with each other.

CI_CTL_GRP_CELL_X_TEST_CELL

The CI_CTL_GRP_CELL_X_TEST_CELL table contains the intersection of a marketing cell that is being used as a control group with a marketing cell that is being used as a test cell in order to determine the performance of this control group comparison. The values for CONTROL_GROUP_CELL_SK and MARKETING_CELL_SK are added as rows in the table when a campaign is published. This table is used for holdout control groups as well as for champion/challenger and challenger/challenger control groups.

CI_MARKETING_CELL

The CI_MARKETING_CELL table contains a grouping of subjects that are targeted by a campaign. An example of a subject is customer, household, or individual.

History and Responses

CI_CONTACT_HISTORY

The CI_CONTACT_HISTORY table contains a contact record of the delivery of a particular package to a particular channel for a particular subject. For more information, see “About History Tables” on page 118.

CI_PRESENTED_TREATMENT_HISTORY

The CI_PRESENTED_TREATMENT_HISTORY table contains a record of the presentation of particular treatments for a particular channel for particular subjects. For more information, see “About History Tables” on page 118.

This table is used in a call center where an operator presents a subset of the delivered treatments to a customer. Data in this table is used for decision campaigns (for SAS Real-Time Decision Manager data) as opposed to selection campaigns (for SAS Marketing Automation data).

CI_CONTACT_HISTORY_STATUS

The CI_CONTACT_HISTORY_STATUS table is a reference table that contains the status codes and descriptions for a contact history record. This table is prepopulated when the common data model is installed. Additional user-defined status codes can be specified. For the default values, see “About Lookup Tables” on page 123.

CI_RESPONSE_HISTORY

The CI_RESPONSE_HISTORY table contains a record of the direct or inferred response to a particular treatment or package that has been made over a particular channel by a particular subject. For more information, see “About History Tables” on page 118.

CI_RESPONSE

The CI_RESPONSE table is a reference table that contains user-defined response codes and their associated descriptions.

CI_RESPONSE_CHANNEL_RESPONSE

The CI_RESPONSE_CHANNEL_RESPONSE table is a reference table that associates the external response codes of a specific channel with internal response codes in order to facilitate the reporting of customer responses.For the default values, see “About Lookup Tables” on page 123.

CI_RESPONSE_X_CELL_PACKAGE

The CI_RESPONSE_X_CELL_PACKAGE table is a reference table that contains the codes for the types of responses (for example, _CV for Converted, _RS for Responded). This table is prepopulated when the common data model is installed.

CI_RESPONSE_TYPE

The CI_RESPONSE_TYPE table is a lookup table that contains the codes for the types of responses such as _CV (which stands for Converted) and _RS (which stands for Responded).

SAS Marketing Optimization

CI_MO_OPTIMIZATION_SCENARIO

The CI_MO_OPTIMIZATION_SCENARIO table contains columns that associate marketing optimization scenarios with campaign groups. This table enables associations between SAS Marketing Optimization and SAS Marketing Automation data.

Integration with SAS Customer Intelligence 360

CI_EXTERNAL_CONTACT

The CI_EXTERNAL_CONTACT table contains a summary of contact events that are extracted from contact data that has been imported from SAS Customer Intelligence 360.

CI_EXTERNAL_CONT_RESPONSE_DETAIL

The CI_EXTERNAL_CONT_RESPONSE_DETAIL table contains details of contact or response impression events that are extracted from contact and response data that has been imported from SAS Customer Intelligence 360.

CI_EXTERNAL_RESPONSE

The CI_EXTERNAL_RESPONSE table contains a summary of response events that are extracted from response data that has been imported from SAS Customer Intelligence 360.

About Extension (_EXT) Tables

Purpose of _EXT Tables

Extension (_EXT) tables are typically created in order to contain any additional information that is specific to your business practices such as the metadata for campaign groups, campaigns, communications, or treatments.

About Populating the _EXT Tables

Overview

The _EXT tables are not initially structured and are initially empty. As part of the customizing process, you define the new column structures that you want to add to the _EXT tables. This is usually a one-time database administrator task.

When a campaign or communication is executed, those name/value pairs in the columns of a UDF table are transposed to the rows of the associated _EXT table. The rows of the _EXT tables are populated with data.

Note: For list type custom details, the value, rather than the display value, is used to populate the _EXT table.

The CI_DYNAMIC_TREATMENT_ATTR_EXT table is populated with the dynamic custom details from the CI_TREATMENT_*_UDF tables and the CI_TREATMENT_EXT table is populated with the static custom details from the CI_TREATMENT_*_UDF tables.

Note: The value of the CI_TREATMENT_CHAR_UDF.CHAR_UDF_VAL variable is limited to 500 characters.

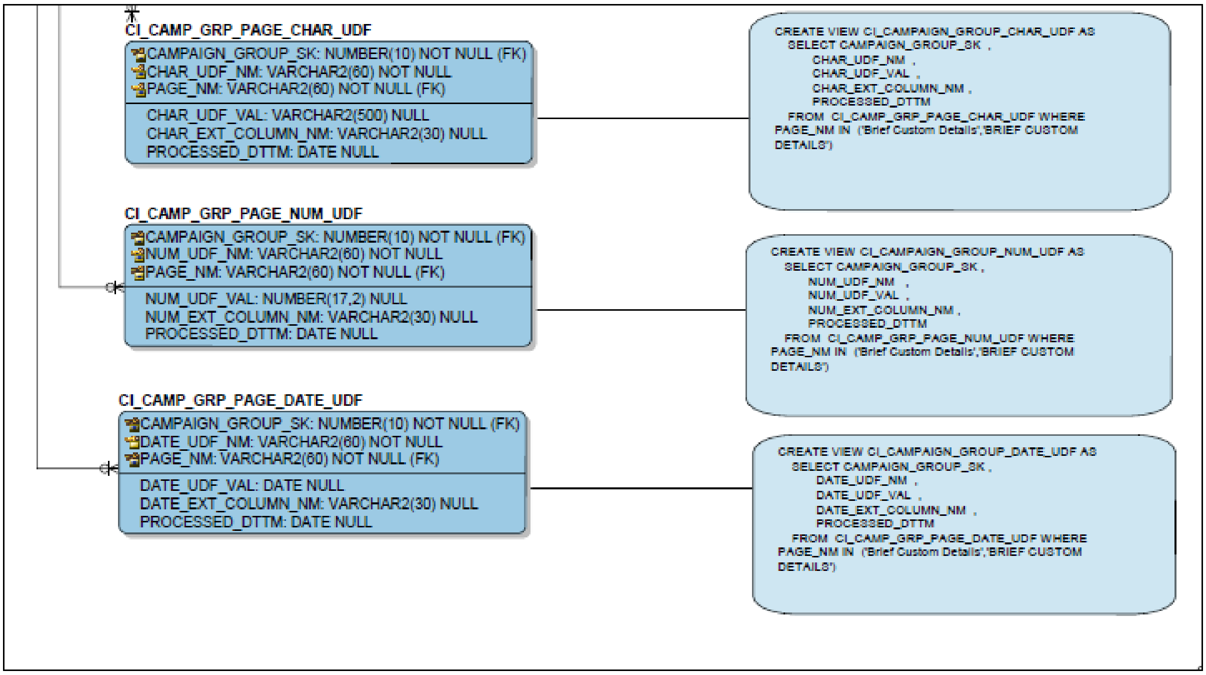
For more information, see "Customizing the Extension (EXT) Tables" on page 136.

UDF (User-Defined Field) Tables

A UDF table exists for each data type (character, numeric, and date). When a campaign is published, a row is added to the associated UDF table for each custom detail that is defined within the campaign.

For campaigns and campaign groups, views are associated with UDF tables.

Figure 6.1 Example of UDF Views


CI_CAMP_GRP_PAGE_CHAR_UDF
CAMPAIGN_GROUP_SK: NUMBER(10) NOT NULL (FK)
CHAR_UDF_NM: VARCHAR2(60) NOT NULL
PAGE_NM: VARCHAR2(60) NOT NULL (FK)
CHAR_UDF_VAL: VARCHAR2(500) NULL
CHAR_EXT_COLUMN_NM: VARCHAR2(30) NULL
PROCESSED_DTTM: DATE NULL
CI_CAMP_GRP_PAGE_NUM_UDF
CAMPAIGN_GROUP_SK: NUMBER(10) NOT NULL (FK)
NUM_UDF_NM: VARCHAR2(60) NOT NULL
PAGE_NM: VARCHAR2(60) NOT NULL (FK)
NUM_UDF_VAL: NUMBER(17,2) NULL
NUM_EXT_COLUMN_NM: VARCHAR2(30) NULL
PROCESSED_DTTM: DATE NULL
CI_CAMP_GRP_PAGE_DATE_UDF
CAMPAIGN_GROUP_SK: NUMBER(10) NOT NULL (FK)
DATE_UDF_NM: VARCHAR2(60) NOT NULL
PAGE_NM: VARCHAR2(60) NOT NULL (FK)
DATE_UDF_VAL: DATE NULL
DATE_EXT_COLUMN_NM: VARCHAR2(30) NULL
PROCESSED_DTTM: DATE NULL

```
CREATE VIEW CI_CAMPAIGN_GROUP_CHAR_UDF AS
   SELECT CAMPAIGN_GROUP_SK ,
         CHAR_UDF_NM ,
         CHAR_UDF_VAL ,
         CHAR_EXT_COLUMN_NM ,
         PROCESSED_DTTM
   FROM  CI_CAMP_GRP_PAGE_CHAR_UDF WHERE
PAGE_NM IN ('Brief Custom Details','BRIEF CUSTOM
DETAILS')
```

```
CREATE VIEW CI_CAMPAIGN_GROUP_NUM_UDF AS
   SELECT CAMPAIGN_GROUP_SK ,
         NUM_UDF_NM ,
         NUM_UDF_VAL ,
         NUM_EXT_COLUMN_NM ,
         PROCESSED_DTTM
   FROM  CI_CAMP_GRP_PAGE_NUM_UDF WHERE
PAGE_NM IN ('Brief Custom Details','BRIEF CUSTOM
DETAILS')
```

```
CREATE VIEW CI_CAMPAIGN_GROUP_DATE_UDF AS
   SELECT CAMPAIGN_GROUP_SK ,
         DATE_UDF_NM ,
         DATE_UDF_VAL ,
         DATE_EXT_COLUMN_NM ,
         PROCESSED_DTTM
   FROM  CI_CAMP_GRP_PAGE_DATE_UDF WHERE
PAGE_NM IN ('Brief Custom Details','BRIEF CUSTOM
DETAILS')
```

The following views are associated with campaign and campaign group UDF tables.

Table 6.1 Views Associated with UDF Tables

Table	View
CI_CAMP_PAGE_CHAR_UDF WHERE PAGE_NM = 'Brief Custom Details'	CI_CAMPAIGN_CHAR_UDF
CI_CAMP_PAGE_NUM_UDF WHERE PAGE_NM = 'Brief Custom Details'	CI_CAMPAIGN_NUM_UDF
CI_CAMP_PAGE_DATE_UDF WHERE PAGE_NM = 'Brief Custom Details'	CI_CAMPAIGN_DATE_UDF
CI_CAMP_GRP_PAGE_CHAR_UDF WHERE PAGE_NM = 'Brief Custom Details'	CI_CAMPAIGN_GROUP_CHAR_UDF

CI_CAMP_GRP_PAGE_NUM_UDF WHERE PAGE_NM = 'Brief Custom Details'	CI_CAMPAIGN_GROUP_NUM_UDF
CI_CAMP_GRP_PAGE_DATE_UDF WHERE PAGE_NM = 'Brief Custom Details’	CI_CAMPAIGN_GROUP_DATE_UDF

About History Tables

Overview of History Tables

The CI_CONTACT_HISTORY, CI_RESPONSE_HISTORY, and CI_PRESENTED_TREATMENT_HISTORY tables have the following requirements or properties:

- They are deployed in the same location as the common data model.
- They must be registered in the SAS Metadata Repository.
- They must be included in a SAS Information Map in order to make the contact history and response history data available for use.
- They are typically connected to the table or tables of the primary marketing data by an outer join.
- The tables contain contact history for SAS Marketing Automation and SAS Real-Time Decision Manager.

Estimating the Size of History Tables

Use either of the following methods to estimate maximum table size:

- Multiply the average number of customers who are targeted by the average number of campaigns that you want to initiate. Then multiply the result by the number of bytes in the total number of rows in a history table.
- Multiply the estimated number of rows in each history table by the size of each row (in bytes). Then multiply the result by the number of years that you want to retain the history table data for each table.

Note: The CI_CONTACT_HISTORY, CI_RESPONSE_HISTORY, and the CI_PRESENTED_TREATMENT_HISTORY tables are the only tables in the common data model that might increase in size as responses are recorded.

The CI_CONTACT_HISTORY Table

What the CI_CONTACT_HISTORY Table Contains

The CI_CONTACT_HISTORY table contains the following columns:

CELL_PACKAGE_SK
: is the primary reference key that associates the contact history with a cell package.

CONTACT_DTTM
: is the date and time at which the subject (such as Customer, Account, or Household) is contacted.

CONTACT_DT
: represents the mm/dd/yyyy value of CONTACT_DTTM.

CONTACT_HISTORY_STATUS_CD
: is the reference key code that associates the contact history with a contact history status. See the CI_CONTACT_HISTORY_STATUS table for the supplied values of the status codes.

EXTERNAL_CONTACT_INFO_ID1 and EXTERNAL_CONTACT_INFO_ID2
: are columns for SAS Real-Time Decision Manager contact processing.

OPTIMIZATION_BACKFILL_FLG
: is a column for use by SAS Marketing Optimization. The initial value for this field is set to NULL, which implies N. This initial value avoids the need to write out each individual record and greatly improves performance.

PACKAGE_HASH_VALUE
: contains a hash value generated to supply a single lookup value for a combination of the dynamic custom details for a package.

RESPONSE_TRACKING_CD
: contains the response code.

SUBJECT_ID
: is the primary reference key of a unique identifier such as Subject ID (for example, Customer, Account, or Household) that associates the CI_CONTACT_HISTORY table with an external table. You need to replace the placeholder value that is assigned for this column during installation with a valid key to the appropriate external table.

Note: SAS Technical Support does not recommend adding fields to the table. If you need additional contact-related fields, then create a separate table and link the information to the history table by using a key.

Structure of the CI_CONTACT_HISTORY Table: Subject Levels

Create a separate table for each subject level for which contacts are made. For example, a customer contact history table and a household contact history table might both need to exist when communications might be directed to the individual or to the household. A contact history table is required only for the potential subject levels that apply to a communication in SAS Customer Intelligence Studio.

For more information, see “Update the Placeholder SUBJECT_ID Columns” on page 134.

The CI_CONTACT_HISTORY table contains one row for each subject for each occurrence of a communication. There is a separate record in the CI_CONTACT_HISTORY table for each time that a selected customer is contacted. Each record represents a unique combination of communication, communication occurrence, and subject (such as **Customer**). Note that occurrence = 1 if the communication is not a recurring communication.

The CI_RESPONSE_HISTORY Table

What the CI_RESPONSE_HISTORY Table Contains

The CI_RESPONSE_HISTORY table contains information about customer behavior in response to a marketing campaign. There is one row for each subject every time that a response is recorded.

The CI_RESPONSE_HISTORY table contains these columns:

CELL_PACKAGE_SK
: is the primary reference key that associates the response with a cell package.

EXTERNAL_RESPONSE_INFO_ID1 and EXTERNAL_RESPONSE_INFO_ID2
: contains channel-specific data that is typically the reference to external channel information such as a transaction number.

INFERRED_RESPONSE_FLG
: is populated by an external program and is the Actual Response that an individual customer provided as a result of a contact (an offer or a treatment).

PROCESSED_DTTM
: is the last date and time that a record was processed.

RESPONSE_CHANNEL_CD
: is the reference response code that indicates the response that originates directly from the channel. This foreign key references the RESPONSE_CHANNEL_CD column in the CI_RESPONSE_CHANNEL_RESPONSE table. The RESPONSE_CHANNEL_CD column is also the role name that corresponds to the CHANNEL_CD column of the CI_CHANNEL table.

RESPONSE_DT
: contains the MM/DD/YYYY value of RESPONSE_DTTM.

RESPONSE_DTTM
: is a primary key that is the date and time of the response.

RESPONSE_SK
: is the primary reference key code that associates the response with the response code. This column refers to the CI_RESPONSE table that is a lookup table for the response codes and their descriptions.

RESPONSE_TRACKING_CD
: contains the response code.

RESPONSE_VALUE_AMT
: is a flag that indicates the type of correlated response. Valid values are Y (inferred response) and N (direct response).

SUBJECT_ID
: is the primary reference key of a unique identifier such as Subject ID (for example, Customer, Account, or Household) that associates the CI_RESPONSE_HISTORY table with an external table. You must replace the placeholder value that is assigned for this column during installation with a valid key to the appropriate external table.

TREATMENT_HASH_VAL
contains a hash value generated to supply a single lookup value for a combination of the dynamic custom details for a package.

TREATMENT_SK
is the primary reference key that associates the response with a treatment.

About Types of Responses

Responses occur from the sales that result from a specific marketing offer. Both direct and inferred responses can be recorded in the CI_RESPONSE_HISTORY table.

Direct Responses

Direct responses are responses that are attributed to a specific direct marketing campaign. Direct responses are captured at the time of purchase by matching a promotion code or key code to the targeted customer names. The code is provided to the customer in the offer. The promotion code is returned in the response by the customer and captured at the touchpoint so that there is a direct link between the campaign and the response. The promotion code is a value that is external to SAS Customer Intelligence. Therefore, you must decide how to manage this link between the external promotion code and a campaign in SAS Customer Intelligence.

Inferred Responses

Inferred responses are those responses that are indirectly attributable to a marketing campaign, as specified by the purchase behavior of a customer. For example, inferred responses might include the responses of all of the customers in the target population of direct marketing Campaign X who bought the product during the campaign. The objective of tracking the inferred responses is to measure the total potential effect of the direct marketing efforts.

Both direct and inferred responses are defined by a response interval that includes a start time and an end time. Responses that are returned outside the defined response interval are not attributed to the campaign ROI. The only responses that do not require a response interval are specified by either trigger-based or event-based campaigns.

A customer who is targeted for a campaign might respond in a number of ways. Possible types of direct and inferred responses that you might track are listed in the following table.

Table 6.2 Sample Types of Responses

Response Type	Direct	Inferred
Acceptance	Provides a promotion code at the time of purchase. Clicks through an email to make a purchase.	Purchases without providing a promotion code (captured in billing data).

Spill (to a similar product in the same category)	Clicks through an email to purchase a similar product. Provides a promotion code at time of purchase and does not purchase the product that is offered, but purchases another product in the same category.	Does not provide a promotion code and does not purchase the product that is offered, but purchases another product in the same category (captured in billing data).
Inquiry	Provides a promotion code and asks for more information, but does not purchase. Opens and clicks through an email, but does not purchase.	Does not provide a promotion code and asks for more information, but does not purchase (captured in touchpoint data).
Decline (often not captured)	Provides a promotion code and does not purchase either the product that is offered or a product in the same category. Receives an email, but does not open it.	Does not provide a promotion code and does not purchase either the product that is offered or a product in the same category. There is no record of the purchase in the billing data.
Halo	Customers who are not targeted for a campaign are not captured in contact history. Therefore, their responses to that campaign are not captured. Forwards the email offer to a friend.	Customers who are not targeted for a campaign are not captured in contact history. Therefore, their responses to that campaign are not captured.

The CI_PRESENTED_TREATMENT_HISTORY Table

What the CI_PRESENTED_TREATMENT_HISTORY Table Contains

The CI_PRESENTED_TREATMENT_HISTORY table contains the following columns:

CELL_PACKAGE_SK
is the primary reference key that associates the presented treatment history with a cell package.

EXTERNAL_CONTACT_INFO_ID1 and EXTERNAL_CONTACT_INFO_ID2
are columns for SAS Real-Time Decision Manager confirm contact processing.

PRESENTED_TREATMENT_DT
contains the MM/DD/YYYY value of PRESENTED_TREATMENT_HIST_DTTM.

PRESENTED_TREATMENT_HIST_DTTM
is the primary key that contains the date and time of the presented treatment.

RESPONSE_TRACKING_CD
contains the response code.

SUBJECT_ID
is the primary reference key of a unique identifier such as Subject ID (for example, Customer, Account, or Household) that associates the CI_PRESENTED_TREATMENT_HISTORY table with an external table. You must replace the placeholder value that is assigned for this column during installation with a valid key to the appropriate external table.

TREATMENT_HASH_VAL
is the hash value for a dynamic treatment. The value is S or blank if the treatment is static.

TREATMENT_SK
is the primary reference key that associates the presented treatment history with a treatment.

About Lookup Tables

Lookup tables contain information about campaigns in SAS Marketing Automation and SAS Real-Time Decision Manager. These lookup tables, also called reference tables, are prepopulated with typical industry values:

- CI_CAMPAIGN_GROUP_STATUS
- CI_CAMPAIGN_GROUP_TYPE
- CI_CAMPAIGN_STATUS
- CI_CAMPAIGN_TYPE
- CI_CHANGE_TYPE
- CI_CHANNEL
- CI_COMMUNICATION_RECURR_TYPE
- CI_COMMUNICATION_STATUS
- CI_CONTACT_HISTORY_STATUS
- CI_CONTROL_GROUP_TYPE
- CI_RESPONSE
- CI_RESPONSE_TYPE

Note: These values are supplied when the common data model is created. Here are the tables, with their prepopulated values. Your lookup tables might contain either different or additional values if they have been customized.

Table 6.3 CI_CAMPAIGN_GROUP_STATUS Table: Prepopulated Values

Campaign Group Status Code	Campaign Group Status Description
__1	Not Ready

__2	Ready to Optimize

Table 6.4 CI_CAMPAIGN_GROUP_TYPE Table: Prepopulated Values

Campaign Group Type Code	Campaign Group Type Description
OI_	Inbound Optimization Group
OO_	Outbound Optimization Group
O__	Other Optimization Group
_I_	Inbound Campaign Group
_O_	Outbound Campaign Group
___	Other

Campaigns that are migrated from a previous release retain their original campaign status codes. The codes are not converted to the values in the following table.

Table 6.5 CI_CAMPAIGN_STATUS Table: Prepopulated Values

Campaign Status Code	Campaign Status Description
1_1	Initiating
1_2	Initiation Complete
1_3	Designing
1_4	Design Complete
2_1	Approval Requested
2_2	Approval Denied
2_3	Approval Approved
3_1	Ready to Execute
3_2	Scheduled
3_3	Executing
3_4	Execute Complete
3_5	Marked for Deployment

Table 6.6 *CI_CAMPAIGN_TYPE Table: Prepopulated Values*

Campaign Type Code	Campaign Type Description
I	Decision
O	Selection
T	Treatment

Table 6.7 *CI_CHANGE_TYPE Table: Prepopulated Value*

Change Type Code	Change Type Description
_NV	New Version

Table 6.8 *CI_CHANNEL Table: Prepopulated Values*

Channel Code	Channel Name	Channel Description
_AG	Agent	Agent channel
_AT	ATM	ATM channel
_BT	Batch	Batch channel
_CA	Calendar	Calendar channel
_CC	Call Center	Call center channel
_CT	Catalog	Catalog channel
_EL	Electronic	Electronic channel
_EM	Email	Email channel
_EV	Event	Event channel
_FX	Fax	Fax channel
_GI	General insert	General insert channel
_MC	Multichannel	Multiple potential channels
_ML	Mail	Mail channel
_MP	Mobile phone	Mobile phone channel
_OP	Off the page	Off the page channel
_PA	Print Ad	Print advertisement channel

_PB	Point of Sale Branch	Point of Sale (POS) branch channel
_PG	Pager mail	Pager mail channel
_PH	Phone	Telephone channel
_PM	Phone mail	Phone mail channel
_RD	Radio	Radio channel
_SC	SAS Conversation Center	SAS Conversation Center channel
_SF	Point of Sale Sales Force	Point of Sale (POS) sales force channel
_SI	Statement Insert	Statement insert channel
_SM	Statement Message	Statement message channel
_SO	Social Media	Social Media channel
_SP	Sales Promo	Sales promotion channel
_TV	TV	Television channel
_WB	Web	Web channel
_WC	Web Campaign	SAS Web Analytics Campaign

Table 6.9 CI_COMMUNICATION_RECURR_TYPE Table: Prepopulated Values

Communication Recurr Type Code	**Communication Recurr Type Description**
__H	hourly
__D	daily
__W	weekly
__M	monthly
__N	none
__U	unknown

Table 6.10 CI_COMMUNICATION_STATUS Table: Prepopulated Values

Communication Status Code	Communication Status Description
_11	Executed
_30	Failed
_32	Future

Table 6.11 CI_CONTACT_HISTORY_STATUS Table: Prepopulated Values

Contact History Status Code	Contact History Status Description
_11	Executed
_13	Excluded Mandatory Contacts
_20	Control Group
_30	Failed

Table 6.12 CI_CONTROL_GROUP_TYPE Table: Prepopulated Values

Control Group Type Code	Control Group Type Description
_AU	Automatic
_CL	Challenger
_CP	Champion
_ST	Manual Manual control groups are created by selecting the **Cell represents a control group** option in the Cell node.

Table 6.13 CI_RESPONSE Table: Prepopulated Values

Response Code	Response Name
_EC	Email - Clicked Link
_EO	Email - Opened
_ER	Email - Reply

Table 6.14 *CI_RESPONSE_TYPE Table: Prepopulated Values*

Response Type Code	Response Type Description
_CV	Converted
_RS	Responded

7

Implementing the Common Data Model

Overview of Steps

This section contains the steps for creating the table structure of the common data model. The common data model is typically created as part of the SAS Customer Intelligence installation. The common data model is used by SAS Marketing Automation and SAS Real-Time Decision Manager. For conceptual information about the common data model, see Chapter 6, “Common Data Model: Concepts,” on page 107.

To implement the common data model, perform the following steps:

1 Verify that the user and group authorizations are specified for the users who start the Customer Intelligence Common Web Service in order to access the

metadata for the business context. For more information, see Chapter 3, “Security,” on page 47.

2 Create the table structure for common data model tables in your database. For more information, see “Creating the Table Structure of the Common Data Model” on page 130.

3 Set up the history tables. For more information, see “Setting Up the History Tables” on page 134.

4 Customize the _EXT tables. To add the user-defined fields for the associated UDF tables, For more information, see “Customizing the Extension (EXT) Tables” on page 136.

5 Set up the lookup tables. For more information, see “Setting Up the Lookup Tables” on page 138.

6 Create the library definition for the common data model. For details, see SAS Intelligence Platform: Data Administration Guide.

7 Edit your business contexts to include the location of the metadata for the common data model tables.

Creating the Table Structure of the Common Data Model

Overview of DDL Scripts

SAS Customer Intelligence provides the data definition language (DDL) scripts that are contained in a program called ci_cdm_ddl_<*db*>.sas. You execute these scripts to create the column structure of the common data model tables in your SAS environment. The SAS system administrator modifies these DDL scripts to customize some of the tables.

To set up the common data model tables, open ci_cdm_ddl_<*db*>.sas (where <*db*> is the database that you are using, as specified in .“How to Find Your DDL Scripts for Creating the Common Data Model” on page 131). Then follow these steps:

- Customize the history tables. Scroll down to BEGIN HISTORY SECTION and follow the instructions for modifying the history tables. For more information, see “Setting Up the History Tables” on page 134.
- Customize the user-defined fields for all UDF tables. Scroll down to BEGIN USER DEFINED SECTION and follow the instructions for modifying the UDF tables.
- Customize the extension tables:
  - CI_CAMPAIGN_EXT
  - CI_CAMP_CHECKLIST_ITEM_EXT
  - CI_CAMPAIGN_GROUP_EXT
  - CI_CAMP_GRP_CLIST_ITEM_EXT

- □ CI_COMMUNICATION_EXT
- □ CI_DYNAMIC_TREATMENT_ATTR_EXT
- □ CI_TREATMENT_EXT

Note: CI_CAMPAIGN_EXT columns that are meant to hold dates must use a database type that holds date and time.

Scroll down to BEGIN EXTENSION SECTION and follow the instructions for modifying the extension tables.

After customizing the ci_cdm_ddl_<*db*>.sas file, execute it via SAS to place the tables in the database.

How to Find Your DDL Scripts for Creating the Common Data Model

The following table lists the name of the program that contains the DDL script for each database type:

Database	Program Name
Amazon Redshift	ci_cdm_ddl_redshift.sas
DB2	ci_cdm_ddl_db2.sas
Greenplum	ci_cdm_ddl_greenplum.sas
Netezza	ci_cdm_ddl_netezza.sas
Oracle	ci_cdm_ddl_oracle.sas
PostgreSQL	ci_cdm_ddl_postgres.sas
SAP HANA	ci_cdm_ddl_hana.sas
SQL Server on UNIX	ci_cdm_ddl_sqlserver_sqlsvr.sas
SQL Server - OLE DB Connection on Windows	ci_cdm_ddl_sqlserver_oledb.sas
SQL Server - ODBC Connection (Windows only)	ci_cdm_ddl_sqlserver_odbc.sas
Teradata	ci_cdm_ddl_teradata.sas

Find the program at the following default locations:

- Under Windows: `Program Files\SASHome\SASFoundation\9.4\cicsvr\sasmisc`
- Under UNIX: `/SASHome/SASFoundation/9.4/misc/cicsvr`

Specify Your Database Connection Values

Overview of Specifying Database Connection Values

To specify the connections between your database and the common data model tables, open ci_cdm_ddl_<*db*>.sas , where <*db*> is the database that you are using. For more information, see "How to Find Your DDL Scripts for Creating the Common Data Model" on page 131. Make the changes that apply for the applicable database type as follows.

Amazon Redshift Example

```
   %let user = <User Name> ;          /* Redshift User     */
   %let pwd = <Password> ;            /* Redshift Password */
   %let dsn = <Database> ;            /* Redshift Database */
   %let schema = <Schema> ;           /* Redshift Schema   */
```

DB2 Example

```
%let user = <User Name>;          /* Other than Default User */
%let pass = <Password>;           /* DB2 Password            */
%let dsn  = <Data Source>;        /* DB2 Data Source         */
```

Greenplum Example

```
%let user = ; /* Other than Default User */
%let pass = ; /* Greenplum Password */
%let db = ; /* Greenplum Database */
%let server = ; /* Greenplum Server */
%let port = ; /* Greenplum Port */
%let schema = ; /* Greenplum Schema */
```

Netezza Example

```
%let lib  = <library>;           /* For example, MAREPORT   */
%let user = <user>;              /* Netezza User            */
%let pwd  = <password>;          /* Netezza Password        */
%let db  = <database>;           /* Netezza Database        */
%let server = <server>;          /* Netezza Server          */
```

Oracle Example

```
%let path = <Oracle TNS Entry>;   /* From tnsnames.ora     */
%let user = <User Name>;          /* Oracle User/Schema    */
%let pass = <Password>;           /* Oracle Password       */
```

For example, if the user name is scott, the password is tiger, and the Oracle TNS Entry computer is cicommon.acme.com, then your %LET statements would be similar to the following statements:

```
%let path = cicommon.acme.com;
%let user = scott;
%let pass = tiger;
```

If the SAS session encoding is UTF-8, the length of staged treatment names might exceed 128 bytes. In this case, there are execution errors. In order to accommodate these longer names, add a DBCONINIT option to the CONNECT statement to specify character semantics for the Oracle database. Character semantics are specified in the following example:

```
   CONNECT TO ORACLE (USER=&USER PASS=&PASS PATH=&PATH DBCONINIT="begin
   execute immediate('ALTER SESSION SET NLS_LENGTH_SEMANTICS=CHAR');end;");
```

For Oracle, if the common data model is in a separate database instance from the customer's subject data (for example: customer, household, account), then the tnsnames.ora file must be updated with the new connection information about the subject data files. The tnsnames.ora file is located on the SAS computer where the Oracle client is installed.

PostgreSQL Example

```
%let server = ; /* Postgres Server */
%let port = ; /* Postgres Port */
%let user = ; /* Postgres User/Schema */
%let pass = ; /* Postgres Password */
%let database = ; /* Postgres Database */
%let schema = ; /* Postgres Schema */
```

Mixed case or uppercase names for tables, columns, or schemas are not supported for PostgreSQL.

SAP HANA Example

```
%let user = <User Name> ;         /* SAP Hana User           */
%let pass = <Password> ;          /* SAP Hana Password       */
%let server = '<Server>' ;        /* SAP Hana Server         */
%let instance = <Instance>;       /* SAP Hana Instance       */
%let schema = <Schema> ;          /* SAP Hana Schema         */
```

SQL Server – Linux Example

```
%let user = ; /* Other than Default User */
%let pwd = ; /* SQL Server Password */
%let dsn = ; /* SQL Server Data Source */
%let schema = ; /* SQL Server Schema */
```

SQL Server – OLE DB Connection Example

```
%let user = <User Name>;          /* Other than Default User */
%let pwd  = <Password>;           /* SQL Server Password     */
%let dsn  = <Data Source>;        /* SQL Server Data Source  */
%let schema = <Schema>;           /* SQL Server Schema       */
%let catalog = <Catalog>;         /* SQL Server Catalog      */
```

SQL Server – ODBC Connection Example

```
%let user = <User Name>;          /* Other than Default User */
%let pwd  = <Password>;           /* SQL Server Password     */
%let dsn  = <Data Source>;        /* SQL Server Data Source  */
%let schema = <Schema>;           /* SQL Server Schema       */
```

If the SAS session encoding is UTF-8, replace `VARCHAR` with `NVARCHAR` throughout the file, as in the following example.

```
CAMPAIGN_NM         NVARCHAR(60) NULL ,
    CAMPAIGN_DESC        NVARCHAR(256) NULL ,
    CAMPAIGN_FOLDER_TXT  NVARCHAR(1024) NULL ,
    CAMPAIGN_OWNER_NM    NVARCHAR(60) NULL ,
```

Teradata Example

```
%let lib    = <Target Library>;  /* For example, MAREPORT    */
%let user   = <User Name>;       /* Teradata User            */
%let pass   = <Password>;        /* Teradata Password        */
%let server = <Server>;          /* Teradata Server or TDPID */
%let database = <Database>;      /* Teradata Database        */
```

Setting Up the History Tables

Overview: Setting Up the History Tables

These three history tables must be created according to subject type within the environment:

- CI_CONTACT_HISTORY records the contacts who are identified.
- CI_PRESENTED_TREATMENT_HISTORY records the specific treatments that were actually presented to the consumer. This table is automatically populated by SAS Real-Time Decision Manager.
- CI_RESPONSE_HISTORY records the responses to specific contacts or presentations of treatments.

The following changes are required to set up these tables for subject and business context. The section of the open ci_cdm_ddl_<*db*>.sas program code for history tables appears in BEGIN HISTORY SECTION.

Note: When you run this code in order to set up a demonstration, one change to this program is required: specify the customer-specific target source connection values in the %LET statements at the beginning of the program.

Update the Placeholder SUBJECT_ID Columns

Placeholder Columns

Each of the tables in BEGIN HISTORY SECTION by default contains the primary key column, SUBJECT_ID, that is used as a placeholder column. A history table can belong to one or more subjects, as well as to one or more business contexts. You must update SUBJECT_ID with a customer-specific subject key column or columns that typically point to an external table of custom business values.

Update a Single Subject

1. SUBJECT_ID is a placeholder column for the subjects. Replace this column with the key column or columns of the subject or subjects. For example, if the subject that is referenced is Customer and if the key column is CUSTOMER_ID that is referenced as VARCHAR(10), then replace the SUBJECT_ID column with this column:

   ```
   CUSTOMER_ID VARCHAR(10) NOT NULL
   ```

   Note: If the subject has a composite key, then the multiple columns of the composite key replace the SUBJECT_ID column.

2. Replace SUBJECT_ID column in the primary key constraint statements.

Update Multiple Subjects

1. Duplicate each of the three history tables for each subject that track history, and assign a subject-specific name to each table. For example, if one of the multiple subjects that are referenced is Customer, then the history tables might have these names:

   - CI_CUST_CONTACT_HISTORY
   - CI_CUST_RESPONSE_HISTORY
   - CI_CUST_PRESENTED_TREAT_HIST

2. Rename the constraint names to match the new subject-specific table names.

3. Follow the same steps in“Update a Single Subject” on page 135 for the set of tables for each subject.

4. Repeat the previous steps for each subject.

Update a Single Subject in Multiple Business Contexts

A single business context requires no change other than the modifications described above for the single or multiple subject updates. If a multiple business context setup is required, you can use one of the following techniques:

- Run this entire SAS program using a different schema for each business context. Every common data model table should be duplicated for each schema.
- For each schema, duplicate the three tables in the history section, once for each subject in all business contexts, and assign an identifiable business context name to each table.

If a company is divided into two separate business contexts, duplicate and name the three tables in the history section according to the corresponding business context. Here are examples:

- CI_AP_CUST_CONTACT_HISTORY
- CI_AP_CUST_RESPONSE_HISTORY

- CI_AP_CUST_PRESENTED_TREAT_HIST
- CI_NA_CUST_CONTACT_HISTORY
- CI_NA_CUST_RESPONSE_HISTORY
- CI_NA_CUST_PRESENTED_TREAT_HIST

Note: Table names must conform to the restrictions (such as length and special characters) of the target data source. For example, the maximum length of a table name in Oracle is 30 characters.

Update Multiple Subjects in Multiple Business Contexts

Multiple subjects within multiple business contexts can also exist. Use a naming convention to easily identify the combination of subject and business context. Here are examples:

- CI_AP_CUST_CONTACT_HISTORY
- CI_AP_CUST_RESPONSE_HISTORY
- CI_AP_CUST_PRESENTED_TREAT_HIST
- CI_AP_HHLD_CONTACT_HISTORY
- CI_AP_HHLD_RESPONSE_HISTORY
- CI_AP_HHLD_PRESENTED_TREAT_HIST
- CI_NA_CUST_CONTACT_HISTORY
- CI_NA_CUST_RESPONSE_HISTORY
- CI_NA_CUST_PRESENTED_TREAT_HIST

Customizing the Extension (EXT) Tables

Overview

These steps are required in order to customize an extension table:

1. Modify the database table definition for the extension table. This step is performed by the system DBA.
2. Identify which custom details (UDFs) are associated with which column of the extension table. This step is performed by the administrator in SAS Management Console.

Note: User-defined fields (UDFs) in the common data model are represented as custom details in SAS Customer Intelligence and in SAS Management Console.

Modify the Definition of an Extension Table

Overview

The extension tables contain one column by default that is a key column that is associated with a main table in the common data model. For example, the extension table, CI_CAMPAIGN_EXT, contains the single column called CAMPAIGN_SK that is a key column in the associated table, CI_CAMPAIGN.

To create the new UDF columns for reporting, add a column to the extension table. Here is an example of SAS code in which a column called CAMPAIGN_ADDITIONAL_DESC is added to the CI_CAMPAIGN_EXT extension table in an Oracle database.

Example: Modifying the Oracle Database Table Definition of an Extension Table

```
/*==================================================================*/
/* Enter Customer Specific Target Source Connection Values - Oracle  */
/*==================================================================*/

%let path = <Oracle TNS Entry> ;  /* From tnsnames.ora      */
%let user = <User Name> ;         /* Oracle User for schema */
%let pass = <Password> ;          /* Oracle Password        */

PROC SQL NOERRORSTOP;

CONNECT TO ORACLE (USER=&USER PASS=&PASS PATH=&PATH);

EXECUTE (ALTER TABLE CI_CAMPAIGN_EXT
  ADD CAMPAIGN_ADDITIONAL_DESC VARCHAR (50)) BY ORACLE;
  DISCONNECT FROM ORACLE;

QUIT;
```

Set Up the User-Defined Fields in SAS Customer Intelligence

Custom details, or user-defined fields, are defined in SAS Customer Intelligence. In order to indicate that a specific UDF (user-defined field) should be added to the extension table, specify the name of the column that corresponds to the custom detail. This custom detail must conform to SAS column-naming conventions. The name can contain uppercase and lowercase letters (A through Z), numeric digits (0 through 9), and underscores (_). The first character in the name must be a letter (A through Z) or an underscore. The name cannot contain blank spaces or special characters.

Populating the Extension Tables

The extension tables are populated automatically when a campaign is published or a campaign is executed. Using the specific UDF tables as input, the system performs simple row-to-column mapping. One column in the extension tables is populated for each row that exists in the associated UDF tables.

This example code shows the values of the specific UDF table that return a value of *East*. The value *East* is also published to the CAMPAIGN_ADDITIONAL_DESC column in the extension table.

```
CAMPAIGN_SK = 1000000001
CHAR_UDF_NM = 'Sales Unit'
CHAR_UDF_SEQ_NO = 000002
CHAR_UDF_CHECKLIST_FLG = 'N'
```

```
CHAR_UDF_VAL = 'East'
CHAR_EXT_COLUMN_NM = 'CAMPAIGN_ADDITIONAL_DESC'
PROCESSED_DTTM = 01-APR-2013 00:00:00
```

If the column does not exist in the extension table, then no error is reported. However, no data is written. You can either create campaign definitions that have extension column names before the tables are modified, or modify the tables before you create the campaign definitions.

Setting Up the Lookup Tables

Prepopulate the Lookup Tables

After the lookup tables for the common data model are created, you next prepopulate the lookup tables with default values. To prepopulate the lookup tables and assign start-up codes, run the ci_cdm_load_codes.sas program. Find ci_cdm_load_codes.sas at the following locations:

- Under Windows: **Program Files\SASHome\SASFoundation\9.4\cicsvr\sasmisc**
- Under UNIX: **/SASHome/SASFoundation/9.4/misc/cicsvr**

Edit ci_cdm_load_codes.sas to set the correct LIBNAME for your database. The following example sets the LIBNAME for an Oracle database.

```
%let lib= CDMLIB;
LIBNAME &lib. ORACLE PATH=&path. USER=&user. PASSWORD=&pass.;
```

The ci_cdm_load_codes.sas program must be run in the same session as the ci_cdm_ddl_.sas script that creates the tables in order to refer to database connection values can be referenced. The `&lib.` macro variable must be set to a valid migration library. This program uses macros that are already defined in SAS server tier. The macros must be executed when the session is connected to the machine that is used with the common data model.

For a list of lookup tables and their prepopulated values, see "About Lookup Tables" on page 123.

Resolving Translated Data

The SASMSG function has been added to each INSERT statement in ci_cdm_ddl_<*db*>.sas to properly resolve the translated data that contains characters in Unicode escape representation. The SASHELP.ci tables that are created by SMD files contain these characters. These characters are resolved by the SASMSG function when the message text is read from the table. These characters are used in the table to enable you to store data for all languages in one SAS data set. The tables are also created in an encoding that supports ASCII characters that are used in Unicode escape representation to prevent problems with data loss during transcoding. These problems can occur if the language of the locale is incompatible with the session encoding.

Registering Data Sources

Updating the Libref for the Common Data Model

Re-register the Common Data Model

An administrator registered the common data model during the installation of SAS Customer Intelligence. However, if you change the location of your common data model tables, then you must re-register the common data model by updating the libref that references the common data model. For details, see SAS Intelligence Platform: Data Administration Guide.

Specify the Common Data Model Library Location for a Business Context

To configure a business context so that it points to the library location for the common data model, specify the data options, database upload options, and reporting options on the **Settings** tab. For more information, see *SAS Marketing Automation: User's Guide*.

Do not specify FASTLOAD, BULKLOAD, or MULTILOAD options in the LIBNAME statement for the common data model library. Those options are available only in the bulk-load facility. For more information, see "Activating the Bulk Load Facility" on page 209.

Note: SAP HANA schema names are case-sensitive. Specify the proper case in the LIBNAME statement.

Saving, Publishing, and the Common Data Model

About Saving versus Publishing

When users publish campaign data, they populate or update the tables of the common data model.

A new campaign designer saves a campaign after creating it. SAS Customer Intelligence does not automatically publish it when it is saved, because any view of the partial data, if the data were updated to the common data model, would be incorrect. The Save operation, because it does not update the common data model, also uses fewer system resources than the Publish operation. You can specify an option in SAS Customer Intelligence to publish a campaign when you save it.

After all of the components of the campaign have been completed and the common data model is ready to be updated, the designer can manually publish the campaign . If the **Automatically publish campaigns on subsequent save** option has been set for the business context, the campaign data is published every time that the campaign is saved after the Publish operation. When any communication within a SAS Marketing Automation campaign is executed, the system automatically publishes the campaign. In subsequent Publish operations, the common data model is updated with any changes that were made since it was last published. The common data model is updated every time that an occurrence runs. Future occurrences are not recorded in the common data model when a campaign is executed or when the campaign is automatically published on subsequent save. Future occurrences are recorded when a campaign is published manually.

Note: If a campaign is published, these consequences might result:

- Performance is reduced during each subsequent Save operation.
- If the campaign designer saves a partially completed campaign , any view of the data in the common data model reflects this partial state.

Analysts can view the reports of the campaign data from the common data model.

Full and Incremental Publish Operations

A full Publish operation occurs when you first manually publish a campaign from within SAS Customer Intelligence. When you first manually publish a campaign , all publishable data is sent to the common data model. Subsequent requests to publish the campaign from within SAS Customer Intelligence result in an incremental Publish operation.

An incremental Publish operation updates only those fields in the common data model that have changed or have never been published. An incremental Publish operation occurs under these conditions:

- You request that a campaign be published from SAS Customer Intelligence and the campaign has been previously published.
- An execution of a SAS Marketing Automation campaign (or a communication within the campaign) occurs.
- A previously published campaign is saved and the **Automatically publish campaigns on subsequent saves** option is selected for the business context.

Migrating Tables

The migration of the common data model from 5.4 and 5.4.1 to 6.1 and later requires the running of the ci_cdm_migrate_to_61_<*db*>.sas script. This is an in-place script that takes place in one schema or database. No migration of data from one schema or database to another schema or database is provided for in this script. You should make a complete backup version of the 5.4 common data model before you run the migration script.

Within each migration script the connection values and history tables must be customized.

As the script executes, there are instances in which a current 5.4 table is dropped and then re-created. In these instances a backup version of the table is created and left intact after the script completes in order to save the original data.

Find the migration scripts at the following locations:

- Under Windows: `Program Files\SASHome\SASFoundation\9.4\cicsvr\sasmisc`
- Under UNIX: `/SASHome/SASFoundation/9.4/misc/cicsvr`

In 6.4, a change was made in the way that treatment custom details of type Numeric Range or Date Range are published in the common data model. For information about migrating tables that contain treatment custom details to 6.5 from previous releases, see “Migrating Files from a Previous Release” in *SAS Marketing Automation: User’s Guide*.

8

Displaying Reports in the Reports Workspace

Overview of Displaying Reports

The Reports workspace in SAS Customer Intelligence Studio contains templates for reports that display data from the common data model. SAS LASR Analytic

Server is an analytic platform that provides a secure, multi-user environment for concurrent access to data that is loaded into memory.

Data is extracted from the common data model to a staging library. Autoload loads data from the staging library to the SAS LASR Analytic Server. After the report templates are imported, reports are created from the data on the SAS LASR Analytic Server.

For information about accessing SAS Customer Intelligence Studio through SAS Visual Analytics and setting reporting options in business contexts, see "SAS Visual Analytics" on page 163. *SAS Marketing Automation: User's Guide*. For information about using SAS Visual Analytics and the SAS LASR Analytic Server, see "SAS Visual Analytics" on page 163.

The following server and libraries are required for displaying reports.

SAS LASR Analytic Server
: Customer Intelligence LASR Analytic Server is the SAS LASR Analytic Server that is installed with SAS Customer Intelligence for use in reporting. You can use this server and the default values, edit the default values, or create your own SAS LASR Analytic Server. For information about Customer Intelligence LASR Analytic Server, see "Customer Intelligence LASR Analytic Server" on page 145.

LASR library
: The autoload process loads data from the data directory into a LASR library. Customer Intelligence LASR Library is installed with SAS Customer Intelligence. You can use this library and the default values, edit the default values, or create your own LASR library. For more information about Customer Intelligence LASR Library, see "Customer Intelligence LASR Library" on page 147.

staging library
: The staging library contains the data that is extracted from the common data model. Customer Intelligence Staging Library is installed with SAS Customer Intelligence. For more information, see "Customer Intelligence Staging Library" on page 148.

Customer Intelligence LASR Analytic Server and Customer Intelligence LASR Library are configured to be used with a single common data model. There can be multiple business contexts at your site as long as they share the same common data model. Sites that use more than one common data model for reporting need additional configuration.

To display reports:

1. Configure the autoload process. For more information about configuring autoload, see "Configure Autoload" on page 150.

2. Schedule the autoload process. For more information, see "Schedule Autoload" on page 153.

3. Create a SAS job that extracts data from the common data model into the staging library. For more information, see "Extract Data" on page 154.

4. Set a schedule for the job to extract data on a regular basis.For more information, see "Schedule the Job" on page 158.

In order for the Reports workspace to be displayed on any device, including mobile devices, SAS Visual Analytics Administration and Reporting must be installed at your site. For more information, see *SAS Visual Analytics:*

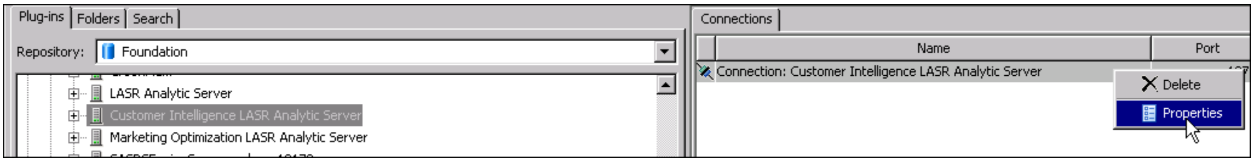
Installation and Configuration Guide at http://support.sas.com/documentation/onlinedoc/va/index.html.

Note: Do not block ReadMetadata access for the SAS Trusted User (for example, sastrust@saspw). The SAS Trusted User is a member of the SAS System Services group. To preserve access, grant the ReadMetadata permission to the SAS System Services group. For more information about SAS Visual Analytics and permissions, see *SAS Visual Analytics: Administration Guide* at http://support.sas.com/documentation/solutions/va/index.html.

Customer Intelligence LASR Analytic Server

Customer Intelligence LASR Analytic Server is located in the Server Manager plug-in in SAS Management Console. Default values were supplied during SAS Customer Intelligence installation and configuration. To view and edit the default values, select **Customer Intelligence LASR Analytic Server**, right-click the server connection, and select **Properties**.

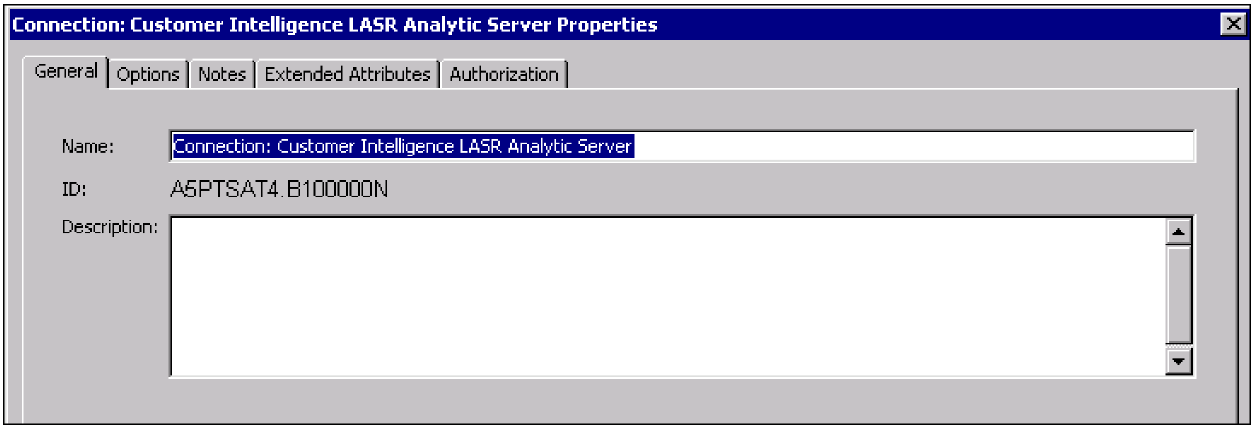
Figure 8.1 Connection Properties


Plug-ins
Folders
Search
LASR Analytic Server
Customer Intelligence LASR Analytic Server
Marketing Optimization LASR Analytic Server
Connections
Name
Port
Connection: Customer Intelligence LASR Analytic Server
Delete
Properties

Repository: Foundation

The **General** tab displays the name of the server connection.

Figure 8.2 Server Connection Name


Connection: Customer Intelligence LASR Analytic Server Properties
General
Options
Notes
Extended Attributes
Authorization

Name: Connection: Customer Intelligence LASR Analytic Server

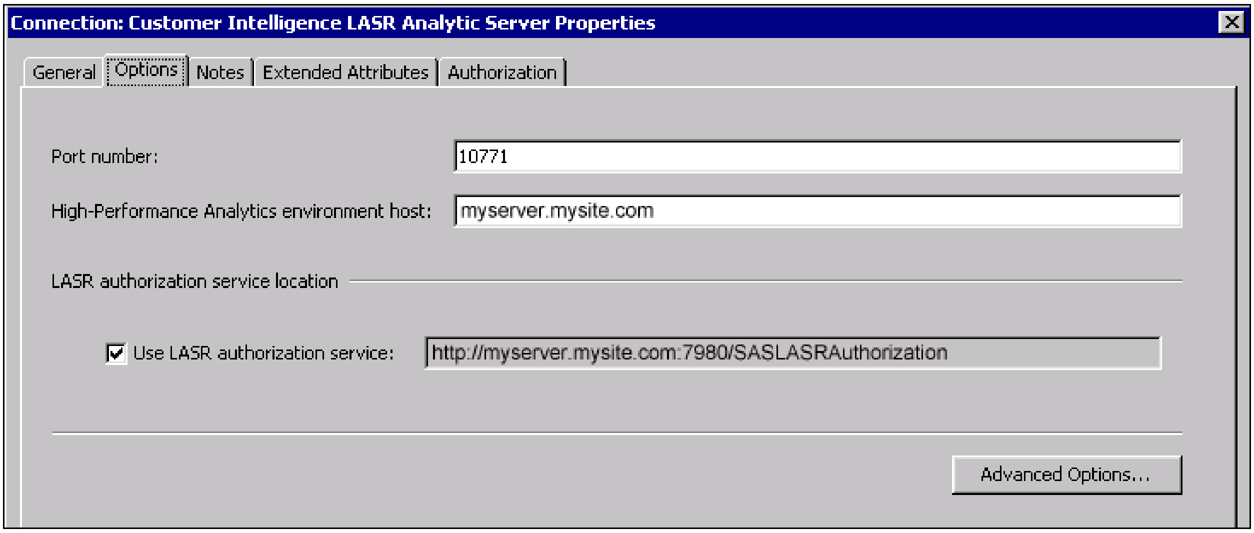
ID: A5PTSAT4.B100000N

Description:

The **Options** tab specifies the port number, and the name of the High-Performance Analytics environment host. The port number is the TCP/IP port number that the Customer Intelligence LASR Analytic Server listens on. The High-Performance Analytics environment host is the host path where files that define the cluster are located. This field is applicable to a distributed server only.

Use LASR authorization service is selected.

Figure 8.3 Server Options



Connection: Customer Intelligence LASR Analytic Server Properties

Port number: 10771

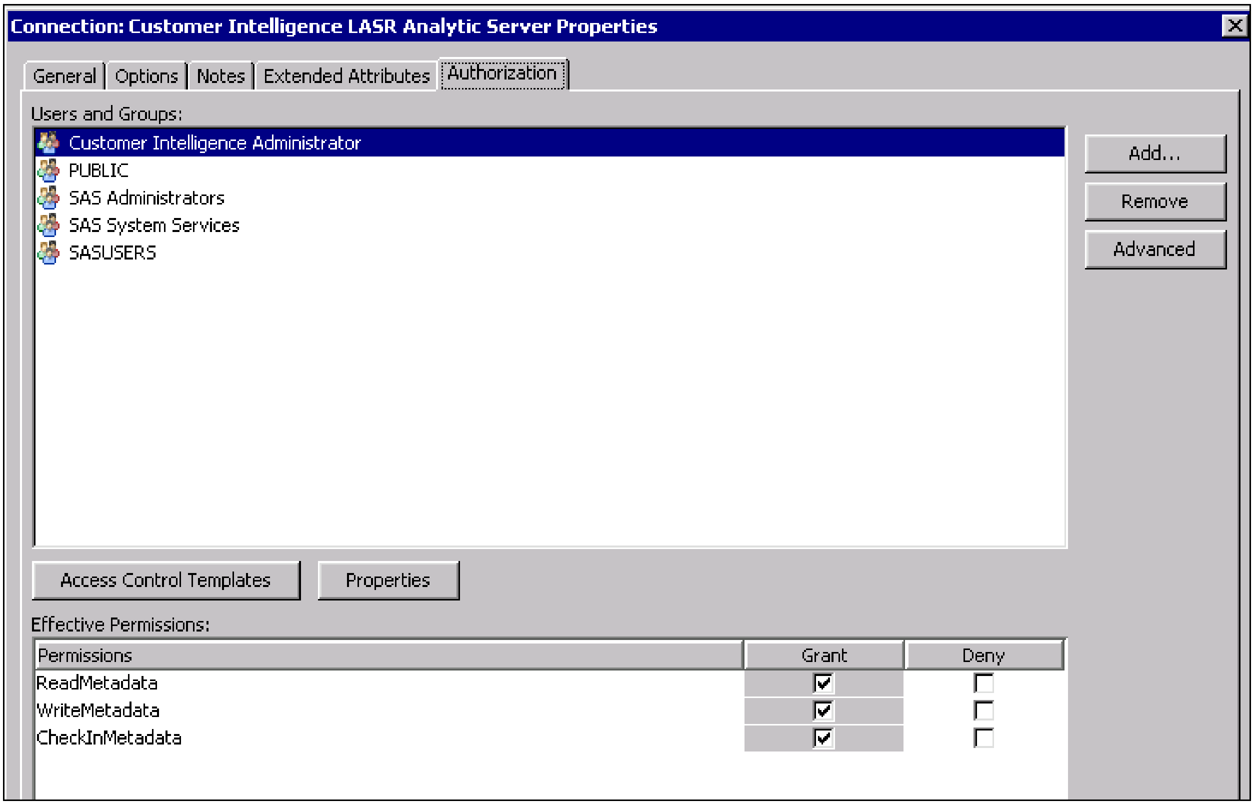
High-Performance Analytics environment host: myserver.mysite.com

LASR authorization service location

☑ Use LASR authorization service: http://myserver.mysite.com:7980/SASLASRAuthorization

On the **Authorization** tab, **ReadMetadata**, **WriteMetadata**, and **CheckinMetadata** permissions are assigned to the **Customer Intelligence Administrator** group.

Figure 8.4 Customer Intelligence Administrator Permissions



Connection: Customer Intelligence LASR Analytic Server Properties

Users and Groups:

- Customer Intelligence Administrator
- PUBLIC
- SAS Administrators
- SAS System Services
- SASUSERS

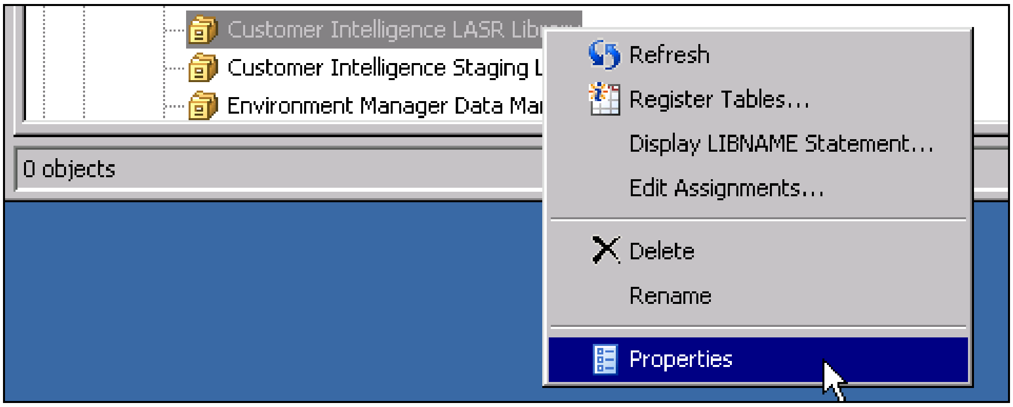
Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	☑	□
CheckInMetadata	☑	□

Customer Intelligence LASR Library

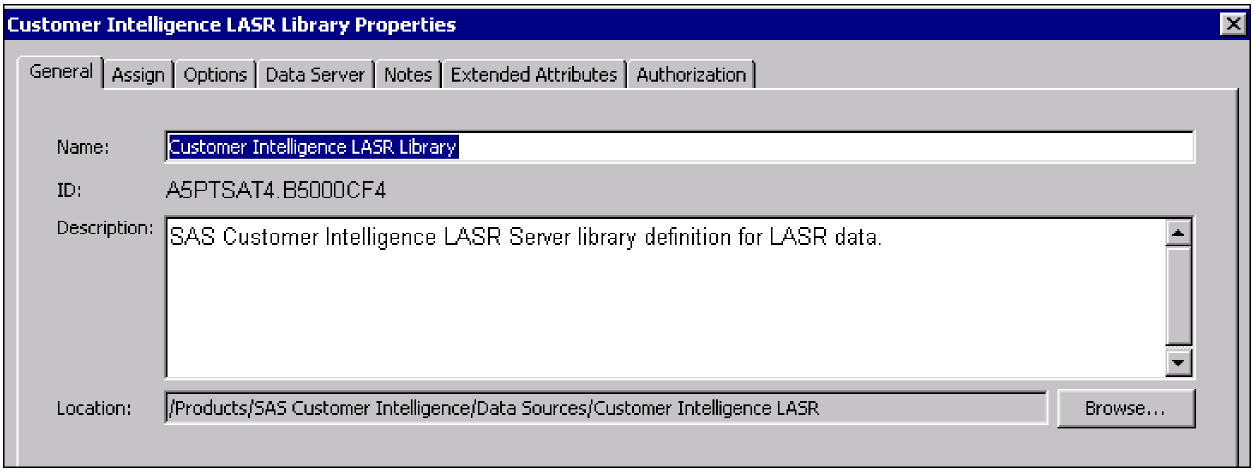
The autoload process loads data from the data directory into the Customer Intelligence LASR Library. The Customer Intelligence LASR Library is located in the `Libraries` folder in the Data Library Manager plug-in in SAS Management Console. Default values were supplied during SAS Customer Intelligence installation. To view and edit the default values, right-click **Customer Intelligence LASR Library** and select **Properties**.

Figure 8.5 LASR Library Properties


Customer Intelligence LASR Libr
Customer Intelligence Staging L
Environment Manager Data Mar
0 objects
Refresh
Register Tables...
Display LIBNAME Statement...
Edit Assignments...
Delete
Rename
Properties

The **General** tab displays the name and description for the library.

Figure 8.6 LASR Library Name and Description



Name: Customer Intelligence LASR Library

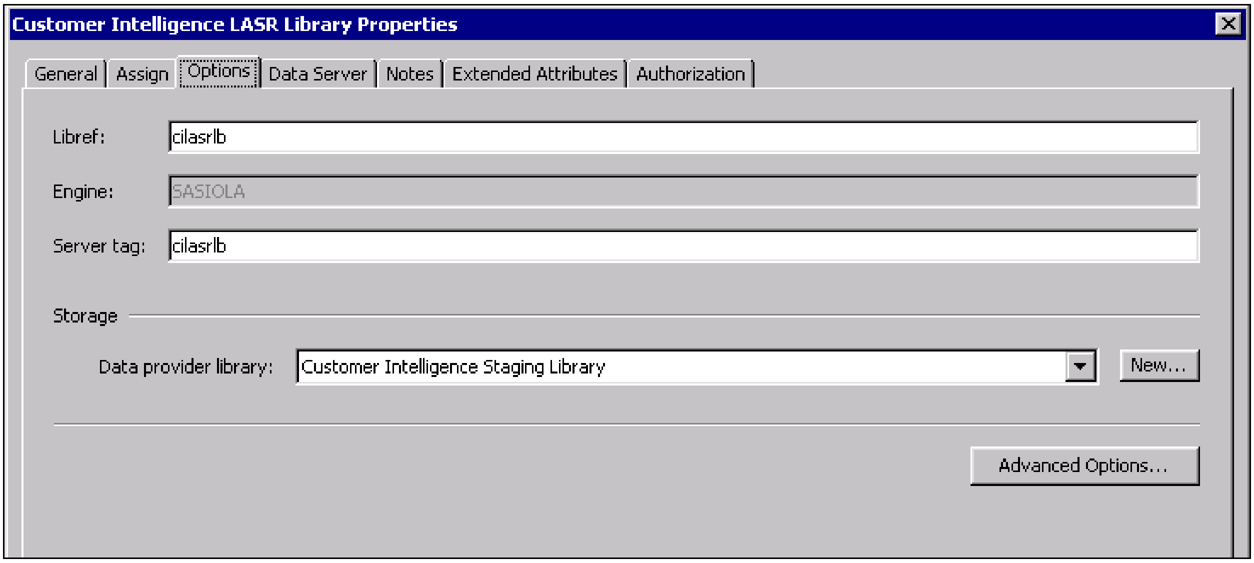
ID: A5PTSAT4.B5000CF4

Description: SAS Customer Intelligence LASR Server library definition for LASR data.

Location: /Products/SAS Customer Intelligence/Data Sources/Customer Intelligence LASR

The **Options** tab specifies the libref and server tag. A server tag is a text string that is associated with a table that is loaded into memory on a Customer Intelligence LASR Analytic Server instance. The server tag is specified in the LIBNAME statement or as a data set option. The server tag and the table name are used together to match the name that is used for tables in the Customer Intelligence LASR Analytic Server. The data provider library is the Customer Intelligence Staging Library that was installed with the product. For more information, see“Customer Intelligence Staging Library” on page 148.

Figure 8.7 LASR Library Options


Customer Intelligence LASR Library Properties

Libref: cilasrlb

Engine: SASIOLA

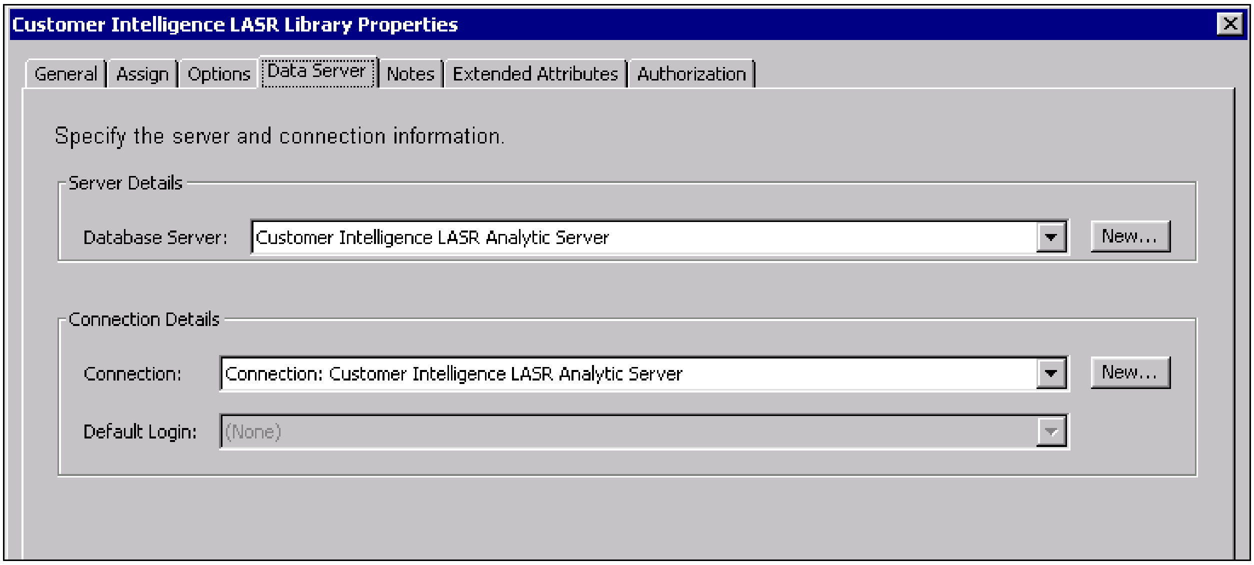
Server tag: cilasrlb

Storage

Data provider library: Customer Intelligence Staging Library

The **Data Server** tab specifies the database server and connection. The server is the Customer Intelligence LASR Analytic Server that was installed with the product. For more information, see “Customer Intelligence LASR Analytic Server” on page 145.

Figure 8.8 LASR Library Data Server


Customer Intelligence LASR Library Properties

Specify the server and connection information.

Server Details

Database Server: Customer Intelligence LASR Analytic Server

Connection Details

Connection: Connection: Customer Intelligence LASR Analytic Server

Default Login: (None)

For information about modifying extended attributes to enable autoload, see “Edit LASR Library Extended Attributes” on page 150.

Customer Intelligence Staging Library

Data is extracted from the common data model and loaded into a staging library whose contents are a data directory.

LASR libraries reference a specific set of data. A LASR library enables the data to be loaded into a LASR Analytic Server. The data is autoloaded from the data directory into the LASR library. The Customer Intelligence LASR Library and the Customer Intelligence Staging Library can be installed with the product. The staging library is located in the `Libraries` folder in the Data Library Manager plug-in in SAS Management Console.

The default path for the root of the staging data directory is `/<config-dir>/AppData/SASCustomerIntelligence/CustomerIntelligenceCommon/Autoload/cilasrlb`. You can change this location during configuration.

The following directories are created under the root directory:

- Append
- Formats
- Logs
- Unload

During configuration, the SAS Deployment Wizard configuration creates the data directories on the SAS Customer Intelligence LASR Analytic Server tier and associates the paths with the SAS Customer Intelligence Staging library in metadata.

If SAS Marketing Automation and SAS Customer Intelligence LASR Analytic Server are installed on different SAS server tiers, the root data directory must be on a shared drive that is accessible by both compute tiers so that SAS Marketing Automation and SAS Visual Analytics can be successfully integrated. If the compute tiers are on different operating systems, such as UNIX and Windows, map the drives so that they are accessible from both tiers.

To map the drives:

1. Log on to SAS Management Console as an Unrestricted user.

2. In the Libraries folder of the Data Library Manager plug-in, right-click Customer Intelligence Staging Library and select **Properties**.

3. Click the **Assign** tab. The server context (for example, SASVAApp) that is associated with the Customer Intelligence LASR Analytic Server is selected in the **Selected servers** field. Move the server context that is associated with the SAS Marketing Automation compute tier from the **Available servers** field to the **Selected servers** field.

4. Click the **Options** tab. The directory in the **Selected items** field specifies the physical location that is on the Customer Intelligence LASR Analytic Server compute tier (for example, SASVAApp). If this path is the same path that is on the SAS Marketing Automation compute tier (for example, SASApp), go to Step 8 on page 150. If this path is different from that path, continue with the next step.

5. Click **New**. In the New Path Specification window, select the SAS Marketing Automation compute tier server context (for example, SASApp) from the **SAS server** list.

6. Click **Browse** and select your staging data directory. A connection is opened to the server that you selected from the **SAS server** list.

7. Issue a LIBNAME statement to ensure that SAS successfully connects to the physical path on each tier. This is an example of a LIBNAME statement:

```
libname stgPath("<tier 1 pathname>","<tier 2 pathname>");
```

8 If the directory paths are the same for the SAS Marketing Automation compute tier and the Customer Intelligence LASR Analytic Server compute tier, open a SAS Display Manager session on one of the compute tiers. Issue a LIBNAME statement to ensure that SAS successfully connects to the physical path. Here is an example:

```
libname stgPath "…/<config-dir>/AppData/SASCustomerIntelligence/
CustomerIntelligenceCommon/Autoload/cilasrlb";
```

The physical path in this example is the default directory. If you configured a different directory when you ran the SAS Deployment Wizard, use the full physical path in the LIBNAME statement.

Configure Autoload

Overview of Configuring Autoload

After the data is extracted from the common data model and loaded into a staging library, autoload retrieves the data and loads it into the LASR library. Autoload periodically synchronizes in-memory data against tables in a designated directory. To enable autoload to synchronize the extracted data with the content of the LASR library, edit LASR library extended attributes, modify the autoload scripts, and set the correct access permissions. The account that executes the autoload script must have a user ID in the group that has access permissions.

For information about autoload and SAS LASR Analytic Server, see the documentation for SAS LASR Analytic Server at http://support.sas.com/documentation/onlinedoc/lasrserver/index.html.

Edit LASR Library Extended Attributes

To enable autoload, edit the values on the **Extended Attributes** tab of the LASR library. For more information about modifying extended attributes for LASR libraries, see *SAS Visual Analytics: Administration Guide* at http://support.sas.com/documentation/onlinedoc/va/index.html.

For information about the location of the **Extended Attributes** tab, see “Customer Intelligence LASR Library” on page 147.

Modify Autoload Scripts

The autoload scripts directory contains autoload, SAS, and scheduling scripts. Verify that the correct values are set in Autoload.sas.

1 Make a copy of the following directory.

On UNIX, the directory is **/*<configuration-directory>*/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/VALIBLA**.

On Windows, the directory is **\*<configuration-directory>*\Applications\SASVisualAnalytics\VisualAnalyticsAdministrator\VALIBLA**.

For more information about the autoload scripts directory and the configuration directory, see *SAS Visual Analytics: Administration Guide* at http://support.sas.com/documentation/onlinedoc/va/index.html.

2 Paste the copy of the directory in the directory for the staging library.

On UNIX, the directory is **/*<configuration-directory>*/AppData/SASCustomerIntelligence/MarketingAutomation**.

On Windows, the directory is **\*<configuration-directory>*\AppData\SASCustomerIntelligence\MarketingAutomation**.

This default location can be modified during configuration or migration.

If the SAS Visual Analytics Administration and Reporting (VAAR) LASR Analytic Server and the Customer Intelligence LASR Analytic Server are on different machines, copy the directory from the VAAR LASR Analytic Server to the machine where the Customer Intelligence LASR Analytic Server is installed.

Do not paste the copy under **SASVisualAnalytics**. The contents of this location are overwritten with the next update to SAS Visual Analytics.

Rename the copied directory. For example, you could name the directory LASRCI.

3 In the new directory, edit AutoLoad.sas. Change the name of the LASR library to Customer Intelligence LASR Library or the LASR library that you created. For more information, see "Overview of Displaying Reports" on page 143.

```
%LET AL_META_LASRLIB=Customer Intelligence LASR Library;
```

4 Edit the following file.

On Windows, the file is runsas.bat.

On UNIX, the file is runsas.sh.

Change the pathname for AUTOLOAD_ROOT to the pathname for the new directory. The following example is for the UNIX environment.

```
AUTOLOAD_ROOT="/<configuration-directory>/AppData/SASCustomerIntelligence/MarketingAutomation/LASRCI"
```

The user ID that executes runsas.bat or runsas.sh must have access permissions to the LASR library, SAS LASR Analytic Server, and target metadata folder that is specified as the **VA.Default.MetadataFolder** value on the **Extended Attributes** tab of the LASR library. For more information, see "Edit LASR Library Extended Attributes" on page 150.

5 Edit the following file.

On UNIX, the file is schedule.sh. Change the pathname for RUNSAS_PATH to include the name of the new directory. The following example is for the UNIX environment:

```
RUNSAS_PATH="/<configuration-directory>/AppData/SASCustomerIntelligence/MarketingAutomation/
LASRCI/runsas.sh"
```

On Windows, the file is schedule.bat. Add the `set RUNSAS_PATH=` option to specify the name of the new directory. The following example is for the Windows environment:

```
set RUNSAS_PATH="\<configuration-directory>\AppData\SASCustomerIntelligence\MarketingAutomation\
LASRCI\runsas.bat"
```

By default, the TIME_INTERVAL_MINUTES value is set to 15 minutes. This is the interval at which autoload synchronizes the data. You can edit this value. If the SAS LASR Analytic Server stops, autoload restarts the server and refreshes the data.

6 Edit the following file.

On UNIX, the file is unschedule.sh. Change the pathname for RUNSAS_PATH to include the name of the new directory. The following example is for the UNIX environment:

```
RUNSAS_PATH="/<configuration-directory>/AppData/SASCustomerIntelligence/MarketingAutomation/
LASRCI/runsas.sh"
```

On Windows, the file is unschedule.bat. Add the `set RUNSAS_PATH=` option to specify the name of the new directory. The following example is for the Windows environment:

```
set RUNSAS_PATH="\<configuration-directory>\AppData\SASCustomerIntelligence\MarketingAutomation\
LASRCI\runsas.bat"
```

If the SAS Visual Analytics Administration and Reporting (VAAR) LASR Analytic Server and the Customer Intelligence LASR Analytic Server are on different machines, take the following steps:

1 Copy the SAS High-Performance Analytics **include** directory to the directory for the staging library.

On UNIX, the directory is **/*<installation-directory>*/SASVisualAnalyticsHighPerformanceConfiguration/7.1/Config/Deployment/Code/Autoload/include**.

On Windows, the directory is ***<installation-directory>*\SASVisualAnalyticsHighPerformanceConfiguration\7.1\Config\Deployment\Code\Autoload\include**.

2 Edit the AutoLoad.sas file. Change the pathname for INCLUDELOC to the pathname for the new include directory. The following example is for the UNIX environment:

```
%LET INCLUDELOC=/<configuration-directory>/AppData/SASCustomerIntelligence/MarketingAutomation/include
```

Test the Autoload Configuration

Test your autoload configuration to verify that autoload is enabled for the LASR library and that the access permissions are correct. From the directory that you created in “Modify Autoload Scripts” on page 150, execute the following file.

On Windows, the file is runsas.bat.

On UNIX, the file is runsas.sh.

Examine the autoload log file for errors.

Schedule Autoload

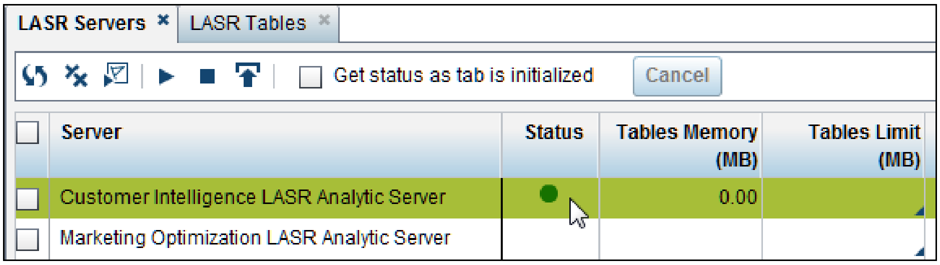
After you have configured autoload, execute the scheduling script so that your changes take effect. From the directory that you created in “Modify Autoload Scripts” on page 150, execute the following file.

On Windows, the file is schedule.bat.

On UNIX, the file is schedule.sh.

When autoload starts for the first time, the script starts the SAS LASR Analytic Server. To verify that the server is running, open the Manage Environment page of SAS Visual Analytics Administrator. The server status is displayed on the **LASR Servers** tab.

Figure 8.9 Server Status



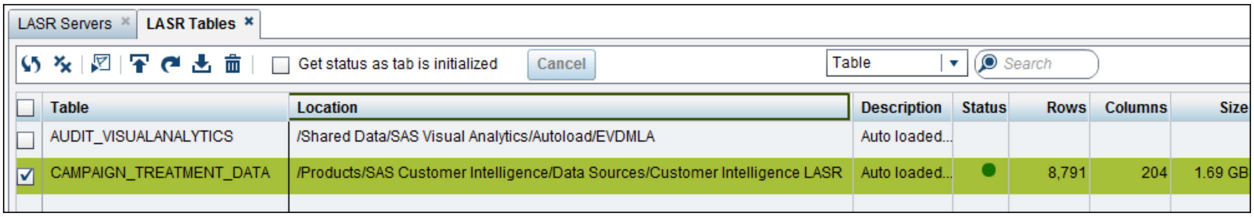
□ Get status as tab is initialized

□	Server	Status	Tables Memory (MB)	Tables Limit (MB)
□	Customer Intelligence LASR Analytic Server	●	0.00	
□	Marketing Optimization LASR Analytic Server			

In some situations, you might need to select a default application server before autoload can start the SAS LASR Analytic Server. For more information, see “Select the SAS Visual Analytics Application Server” on page 159.

The table status is displayed on the **LASR Tables** tab.

Figure 8.10 Campaign Status



□ Get status as tab is initialized

□	Table	Location	Description	Status	Rows	Columns	Size
□	AUDIT_VISUALANALYTICS	/Shared Data/SAS Visual Analytics/Autoload/EVDMLA	Auto loaded...				
☑	CAMPAIGN_TREATMENT_DATA	/Products/SAS Customer Intelligence/Data Sources/Customer Intelligence LASR	Auto loaded...	●	8,791	204	1.69 GB

When you validate autoload for the first time, the list of campaign tables is empty because no data has been extracted.

Extract Data

Overview of Extracting Data

A SAS job extracts the data from the common data model and loads the data into the staging library. The previous version of the data is replaced. The SAS job does the following.

- Assigns librefs to the common data model, the LASR library, and the staging library
- Creates a subject table and the associated contact history, response history, and treatment history tables from the common data model
- Defines additional fields to extract from the common data model
- Uses the %CI2LASR macro to extract the data from the common data model. For information about the contents of the table that contains the extracted data, see Appendix 1, “%CI2LASR Output Table Data Dictionary,” on page 235.

When autoload runs, the data is loaded into a LASR library.

If any parameters in the job are not valid, the table is not created.

Note: If SAS Marketing Automation and SAS Visual Analytics are installed on different SAS server tiers, the extracted data must be on a shared drive that is accessible by autoload. Specify the shared drive information in the VA.Autoload.Location extended attribute for the LASR library. For more information, see “Customer Intelligence Staging Library” on page 148. The job that extracts data from the common data model must run on the SAS server tier where SAS Marketing Automation is installed.

Example Job

Here is an example of a SAS job that extracts data from the common data model. The output is located in a file named CAMPAIGN_TREATMENT_DATA. All templates use the contents of this file to generate reports.

```
/************************************************************/
/* Assign SAS libref for the CI Common Data Model Library   */
/* -------------------------------------------------------- */
/* Note you can find libname statements in SMC.             */
/* Select the library object, right-click, and              */
/* select Display LIBNAME Statement.  Cut and paste the     */
/* libname statement from the dialog box.                   */
/************************************************************/

LIBNAME CICDM ORACLE
        PATH=pathname  SCHEMA=myschema  AUTHDOMAIN="mydomain" ;

/************************************************************/
/* Assign SAS libref for library for the extracted data     */
/* -------------------------------------------------------- */
/* Library for autoload data directory location.  This is   */
/* the pathname that is specified as the                    */
/* VA.Autoload.Location extended attribute for the          */
/* LASR library.                                            */
/*                                                          */
/************************************************************/

LIBNAME BASELIB BASE "pathname";

/************************************************************/
/* In order to extract data from the CI Common Data Model   */
/* it is necessary to know what subjects are of interest.   */
/* For each subject, it is necessary to know what tables    */
/* hold the Contact History, Response History, and          */
/* Treatment History data, so it can be summarized.         */
/*                                                          */
/* Use the %CISUBJ macro to create a Subject Table.  Each   */
/* call to the %CISUBJ macro adds a subject to the table.   */
/* If the table does not exist, it is created.              */
/************************************************************/

%ciSubj (subjectTable=WORK.CI_SUBJECT,
         subject=Customer,
         contact_histTable=CI_CONTACT_HISTORY_CUST,
         response_histTable=CI_RESPONSE_HISTORY_CUST,
         presentedTreatment_Table=CI_PRESENTED_TREATMENT_CUST);

%ciSubj (subjectTable=WORK.CI_SUBJECT,
         subject=AllAcct,
         contact_histTable=CI_CONTACT_HISTORY_ACCT,
         response_histTable=CI_RESPONSE_HISTORY_ACCT,
         presentedTreatment_Table=CI_PRESENTED_TREATMENT_ACCT);

%ciSubj(subjectTable=WORK.CI_SUBJECT,
        subject=Household,
        contact_histTable=CI_CONTACT_HISTORY_HHD,
        response_histTable=CI_RESPONSE_HISTORY_HHD,
        presentedTreatment_Table=CI_PRESENTED_TREATMENT_HHD);

/************************************************************/
/* Additional user-requested fields.  Note that             */
/* the syntax for this macro variable consists of items for */
/* a SQL select clause.  Each item must have a trailing     */
/* comma.                                                   */
/* Only tables that are already included in the extract can */
/* be used to request fields that are not included in the   */
/* extract.                                                 */
/*                                                          */
/* Each extracted field is specified in the form:           */
/*                                                          */
/*    <table name>.<column name> AS alias_column_name       */
/*        'descriptive label you will see in VA' ,          */
/*                                                          */
/************************************************************/


%let ci_user_req_fields =
   ci_campaign.max_budget_offer_amt
       as campaign_max_offer
             'Campaign max budget offer',
   ci_communication.max_budget_offer_amt
       as communication_max_offer
             'Communication max budget offer',
;
/************************************************************/
/* Extract subject data from the common data model and      */
/* load it into a staging library.  All parameters are      */
/* required.                                                */
/************************************************************/

%ci2lasr(subjectTable=work.CI_SUBJECT,
         CIDMLIB=CICDM,
         extract_libref=BASELIB,
         outputTable=CAMPAIGN_TREATMENT_DATA);
```

The %CISUBJ Macro

The %CISUBJ macro builds a table of subjects and related tables for contact history, response history, and treatment history.

```
/*****************************************************************/
/* NAME: cisubj.sas                                              */
/* VERSION: 6.5                                                  */
/* DESCRIPTION: build table of subjects and related tables for   */
/*              contact, response, and treatment history data    */
/*                                                               */
/* Parameters:                                                   */
/*   subject= name of subject                                    */
/*   contact_histTable= name of contact history table            */
/*   response_histTable= name of response history table          */
/*   presentedTreatment_Table= name of treatment history table   */
/*   subjectTable= name of subject table to create/update        */
/*                 (default is WORK.CI_SUBJECT)                  */
/*                                                               */
/* PRODUCT: MA                                                   */
/* USAGE:                                                        */
/*                                                               */
/*  %ciSubj (                                                    */
/*      subjectTable=WORK.CI_SUBJECT,                            */
/*      subject=Customer,                                        */
/*      contact_histTable=CI_CONTACT_HISTORY_CUST,               */
/*      response_histTable=CI_RESPONSE_HISTORY_CUST,             */
/*      presentedTreatment_Table=CI_PRESENTED_TREATMENT_CUST);   */
/*                                                               */
/*  Note that the subject table is consumed by the macros        */
/*  %CICOUNT and %CI2LASR, using the libref of the common data   */
/*   model                                                       */
/*****************************************************************/
```

The %CICOUNT Macro

The %CICOUNT macro stores the updated counts for each subject in the given subject table from the contact history, response history, and presented treatment history tables. The %CICOUNT macro is called by the %CI2LASR macro.

```
/*****************************************************************/
/* NAME: cicount.sas                                             */
/* VERSION: 6.5                                                  */
/* DESCRIPTION: accumulate metrics for each subject in the given */
/*              subject table from the contact history, response */
/*              history, and presented treatment history tables  */
/*              for the subject.                                 */
/*                                                               */
/* Parameters:                                                   */
/*   CIDMLIB= libref of common data model                        */
/*   contact_histTable= name of contact history table            */
/*   subjectTable= name of subject table to create/update        */
/*                 (default is WORK.CI_SUBJECT)                  */
/*   resultTable= name of summary data table to create           */
/*                 (default is WORK.CI_CELL_PACKAGE_COUNTS)      */
/*                                                               */
/* PRODUCT: MA                                                   */
/* USAGE:                                                        */
/*                                                               */
/*  %cicount (                                                   */
/*      CIDMLib=CICDM,                                        */
/*      subjectTable=WORK.CI_SUBJECT,                            */
/*      resultTable=WORK.CI_CELL_PACKAGE_COUNTS);                */
/*                                                               */
/*  Note that the subject table is consumed by the %CI2LASR      */
/*  macro using the libref of the common data model              */
/*                                                               */
/*****************************************************************
```

The %CI2LASR Macro

The %CI2LASR macro extracts data from the common data model.

```
/*******************************************************************/
/* NAME: ci2lasr.sas                                               */
/* VERSION: 6.5                                                    */
/* DESCRIPTION: extract data from the common data model and        */
/*   accumulate counts using %CICOUNT                              */
/* Parameters:                                                     */
/*  extract_libref=  libref of staging library                     */
/*  CIDMLIB=         libref for CDM                                */
/*  subjectTable=    name of subject table                         */
/*  outputTable=     name of extracted table                       */
/* PRODUCT: MA                                                     */
/* USAGE:                                                          */
/*                                                                 */
/*  %ci2lasr(                                                      */
/*             extract_libref=%(),                                 */
/*              CIDMLib=%str(),                                        */
/*              subjectTable=%str(),                                   */
/*              outputTable=%str(),                                    */
/*              mode=SIZE | EXTRACT);                                  */
/*                                                                     */
/*********************************************************************/
```

mode=EXTRACT is the default. If you specify `mode=SIZE`, the table is not created. Instead, a data set named WORK.CI_EXTRACT_SIZE is created with the following columns.

Row_extract_count	the number of rows that the table would contain.
Column_count	the number of columns in each row.
Row_byte_size	the number of uncompressed bytes in each row.
Total_table_byte_size	the number of uncompressed bytes in the table. This is the result of multiplying row size times row count.

Schedule the Job

You use batch scheduling or SAS Management Console to schedule the job to extract data on a regular basis. The extracted data is a snapshot of the contents of the common data model at the time of the extraction. During the process of extracting and loading the data, users are not able to view or create reports. The best time to run this job is when no campaigns are executing and there are no users logged on to SAS Customer Intelligence Studio. Autoload loads the most recently extracted data.

The user ID that schedules the job must have scheduling permissions.

To schedule the job from the command line in the Windows environment, create a BAT file that executes the job. In the following example, the job has been saved in a file named extract.sas.

```
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" -sysin c:\mysasjobs\extract.sas
```

Enter the following command on the Windows command line. In this example, the BAT file is named sasjobextract.bat. The job is executed every day at 1:00 am.

```
schtasks create /sc DAILY /st 1:00:00 /tn C:\myprogramfiles\sasjobextract.bat
```

To schedule the job from the command line in the UNIX environment, create a cron table that sets the schedule. In the following example, the job has been saved in a file named extract.sas. The job is executed every day at 1:00 am.

```
0 1,* * 0-6 sas /mysasjobs/extract.sas
```

Enter the following command at the UNIX prompt. In the following example, the cron table file is named sasjobextract.

```
crontab sasjobextract
```

You can also schedule the job through the Schedule Manager plug-in in SAS Management Console.

For more information about scheduling and executing SAS jobs, see *Scheduling in SAS* at http://support.sas.com/documentation/onlinedoc/sasmc/index.html.

For more information about scheduling tasks in the UNIX and Windows environments, see the documentation for your operating system.

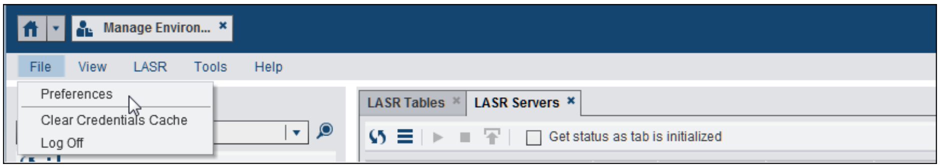
Select the SAS Visual Analytics Application Server

Application servers might be installed on different SAS server tiers. In this situation, you must select the SAS Visual Analytics application server as the default before you can start the SAS LASR Analytic Server. To select the application server:

1 Open the Manage Environment page of SAS Visual Analytics Administrator.

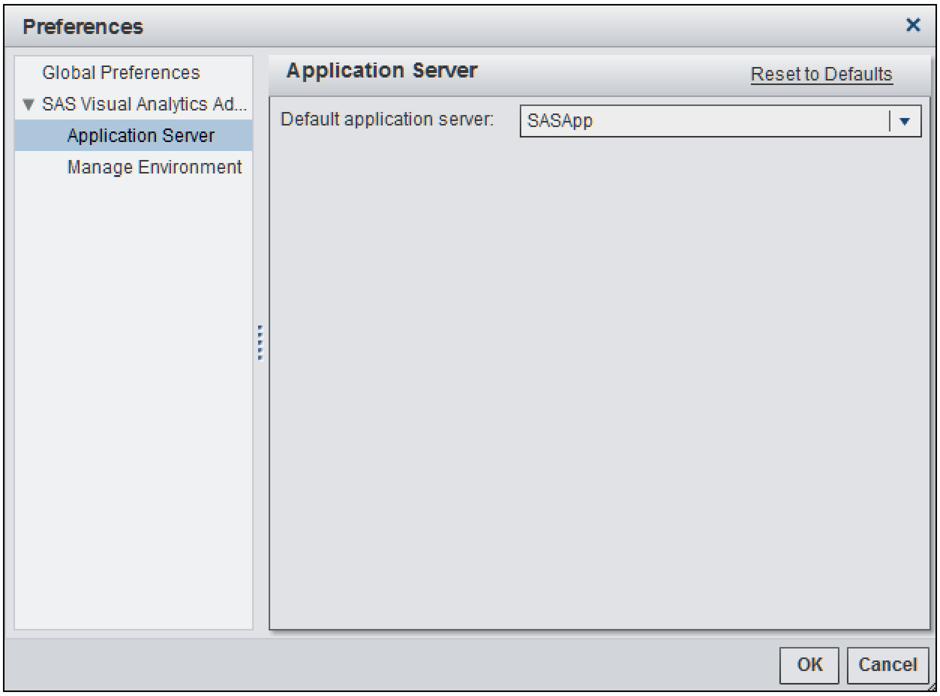
2 Select **File** ▶ **Preferences.**

Figure 8.11 Manage Environment Preferences



3 On the Application Server page of the Preferences window, select the application server for SAS Visual Analytics from the **Default application server** list.

Figure 8.12 Select Default Application Server



Default application server: SASApp

Reports and Templates in SAS Management Console

Report Templates

SAS Customer Intelligence Studio provides templates that you can use to display reports in the Reports workspace. During installation, the SAS Package maps the table to an existing table on the SAS LASR Analytic Server. Before you install the report templates, extract data and create the table. For more information, see "Extract Data" on page 154.

To install the report templates, take the following steps.

1 Copy the SAS package that contains the report templates from the following location on the SAS Customer Intelligence Application Server tier to a location on your SAS Management Console client machine.

On UNIX, the location of the report templates package is:

```
<SASHome>/SASFoundation/9.4/misc/cicsvr/Config/Deployment/Packages/cireporttemplates.spk
```

On Windows, the location of the report templates package is:

```
<SASHome>\SASFoundation\9.4\cicsvr\sasmisc\Config\Deployment\Packages\cireporttemplates.spk
```

SASHome is the directory where SAS is installed on your system.

2 On the **Folders** tab of SAS Management Console, right-click `SAS Folders/Products/SAS Customer Intelligence/Report Templates` and select **Import SAS Package**.

3 In the Import SAS Package wizard, the location on your SAS Management Console client machine that you specified in Step 1.

4 Follow the rest of the steps in the wizard to import the report templates into the Report Templates folder. On the Tables page of the wizard, specify `CAMPAIGN_TREATMENT_DATA` as the target table.

Report templates have the following extended attributes:

ci.external
: specifies that the report is not filtered by business context and that the location of the LASR library is not verified. This attribute enables you to generate and view reports against any LASR library that is currently in memory, including LASR libraries that are not part of SAS Customer Intelligence. The report displays data from all the business contexts in the common data model.

If both the ci.template and ci.external attributes are specified for a SAS Visual Analytics report, then the ci.external attribute and the ci.template attribute is set for all report instances.

ci.template
: identifies the report document as a report template. All templates are displayed in the Select Template window when you create a new report in SAS Customer Intelligence Studio. This is the only attribute that is assigned

to the templates that are provided with SAS Customer Intelligence Studio. The report displays only data from the current business context.

This attribute must be added in SAS Management Console to any SAS Visual Analytics report that you want to display in the Reports workspace. For more information, see "Display Extended Attributes" on page 162.

If a report template has both the ci.external and the ci.template extended attributes, the report displays data from all of the business contexts in the common data model. However, the report is visible only in the current business context.

By default, a report template is visible in all business contexts. To limit the template to a single business context, see "ci.bc.name" in "Report Instances" on page 161.

Report Instances

Report instances are located in the SAS Management Console folder that you specify when you save a report in SAS Customer Intelligence Studio. Reports have the following extended attributes.

ci.report

identifies the file as a report. This extended attribute is required to display the report in the Reports workspace in SAS Customer Intelligence Studio.

This attribute is added automatically when the file is saved in SAS Customer Intelligence Studio.

ci.bc.name

identifies a business context. This extended attribute is required to display the report in the Reports workspace in SAS Customer Intelligence Studio. You cannot specify more than one business context. When this attribute is specified, the report is visible only if the value of the attribute matches the business context for the current user.

This attribute is added automatically when the report instance is saved in SAS Customer Intelligence Studio.

You can use SAS Management Console to add this attribute to any SAS Visual Analytics report that you want to display in the Reports workspace and limit to users of a particular business context. For more information, see "Display Extended Attributes" on page 162.

The attribute is required for all report instances. It is optional for report templates.

ci.source.template.name

specifies the name of the template from which the report was created. The template name is displayed with the report in the Reports workspace in SAS Customer Intelligence Studio.

This attribute is added automatically when the file is saved in SAS Customer Intelligence Studio.

ci.source.template.path

specifies the pathname of the template from which the report was created.

This attribute is added automatically when the file is saved in SAS Customer Intelligence Studio.

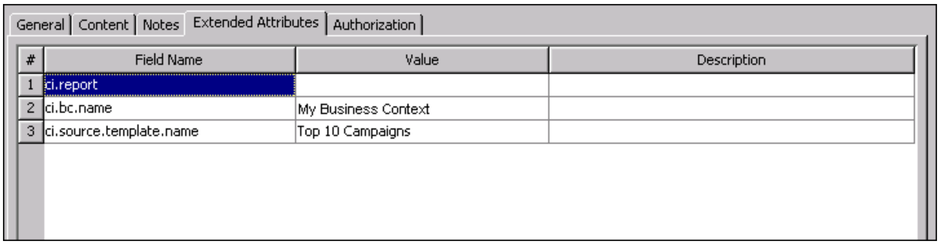
For information about designing reports, see *SAS Visual Analytics: User's Guide* at http://support.sas.com/documentation/onlinedoc/va/index.html.

Display Extended Attributes

To list the extended attributes of a template or a report:

1. Right-click the filename, and select **Properties**.
2. Click the **Extended Attributes** tab.

Figure 8.13 Example of Extended Attributes for Report



#	Field Name	Value	Description
1	ci.report		
2	ci.bc.name	My Business Context	
3	ci.source.template.name	Top 10 Campaigns	

How Responses Are Calculated in Reports

Response values for a communication are processed as TOTAL_RESPONSE_CNT, TOTAL_SUCCESSFUL_RESPONSE_CNT, and TOTAL_RESPONSE_VALUE.

TOTAL_RESPONSE_CNT
: is the total number of subjects that responded to the package. This value is a count of subjects such as Customer, Household, or Account.

TOTAL_SUCCESSFUL_RESPONSE_CNT
: is the total number of responses in which the response type is Converted.

TOTAL_RESPONSE_VALUE
: is the total monetary value of the response.

The TOTAL_RESPONSE_CNT value might be deceptively large if there are many possible responses. For example, a single customer might account for five responses to a single communication:

- Open email
- Click link
- View product
- Add to cart
- Purchase

TOTAL_SUCCESSFUL_RESPONSE_CNT depends on how successful responses are defined for the communication. If more than one response is marked as **Converted** in a Communication node, the number of responses might reflect a higher number than expected.

SAS Visual Analytics

Documentation for SAS Visual Analytics

Documentation for SAS Visual Analytics is available at http://support.sas.com/documentation/onlinedoc/va/index.html.

Note: A user ID and password are required to access this documentation. Licensed customers can request the user ID and password from SAS Technical Support.

Provide Access to SAS Visual Analytics

In the standard configuration, you can provide access to SAS Visual Analytics functionality as follows:

- Add regular users to the Visual Analytics Users group.
- Add administrators to the Visual Analytics Data Administrators group.

The user roles for SAS Visual Analytics are documented in the *SAS Visual Analytics: Administration Guide*.

Start a SAS LASR Analytic Server

Instructions

1. On the SAS Visual Analytics home page, click **Manage Environment**.
2. In the main menu bar in SAS Visual Analytics Administrator, select **LASR** ▶ **Manage Servers**.
3. Select a server, right-click, and select **Start**.

Troubleshooting

If the server does not start, make sure the following requirements are met:

- The SAS Application Server that is used must contain a workspace server that runs on the same machine as the SAS LASR Analytic Server. To determine which SAS Application Server is being used:

  1. From the main menu bar in SAS Visual Analytics Administrator, select **File** ▶ **Preferences**.
  2. In the Preferences window, select the **Application Server** node (below the **SAS Visual Analytics Administrator** node). The **Default application server** drop-down list indicates which server is in use.

  TIP In the standard configuration, the SASApp application server is used by SAS Visual Analytics.

- You must use an account that is known to the workspace server's host.

Windows Specifics: The account must have the **Log on as a batch job** privilege.

UNIX Specifics: For a distributed SAS LASR Analytic server, the account must have passwordless SSH access to all of the machines in the cluster.

For more information, see the *SAS Visual Analytics: Administration Guide*.

9

Using SAS Stored Processes for SAS Marketing Automation

Stored Process Connections

You can configure and use a SAS stored process for selection campaigns by creating a Process node or a custom tool. SAS Customer Intelligence Studio users can drag the Process node or a custom node onto a diagram of a selection campaign. For more information about creating and using stored processes, see *SAS Marketing Automation User's Guide*. See also SAS Stored Processes: Developer's Guide.

When you name your stored process, make sure that the name is different from the stored process names that are used by the SAS Marketing Automation core. For details, see http://support.sas.com/kb/18/484.html.

To improve the performance of the stored process server, see Troubleshooting and Tuning SAS Stored Process Server and SAS Pooled Workspace Server load-balancing for SAS Marketing Automation.

For information about exporting and importing SAS stored processes, see "Stored Processes" on page 70.

Configure Stored Processes for Pooled Workspace Server

By default, stored processes run on a stored process server. To configure stored processes to run on a pooled workspace server, specify the following option in the start-up script for the SAS Web Application Server.

```
-Dsas.ci.stp.workspace.enabled
```

For more information on pooled workspace servers, see *SAS Intelligence Platform: Overview* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Modify SAS Code to Conform to SAS Stored Process Syntax

You can modify SAS code to run as a stored process for SAS Marketing Automation. In this method, you can add a few lines of SAS code in order to conform to SAS Stored Process syntax. This modification is identical to the modifications that are made by the New Stored Process wizard. The registered stored process can be used in a new Process node. SAS Customer Intelligence Studio displays all registered stored processes for which the keyword is `MAUser`.

Modify the SAS code in the previous section at the beginning and the end of the code block to define the code as a stored process:

```
%stpbegin;
%maspinit;

proc sql noprint;
select VALUE into :campname from &MATABLEFORMACRO where
     CATEGORY='CAMPAIGNINFO' and NAME='CAMP_NAME';
select VALUE into :commname from &MATABLEFORMACRO where
     CATEGORY='COMMUNICATIONINFO' and NAME='COMM_NAME';
select count(*) into :commcount from &intable;
quit;
%let CampName=&campName;
%let CommName=&commName;
%let CommCount=&commCount;

filename tweet temp;
filename twtOut temp;
data _null_;
   file tweet;
   nd = date();
   nt = time();
   put "status=Campaign &campname ran with &commcount customers for
      communication &commname on " nd date9. " at " nt time8.0 ".";
run;
proc http
    in=tweet
    out=twtOut
    url="http://twitter.com/statuses/update.xml"
    method="post"
    proxyhost="inetgw.unx.sas.com"
    proxyport=80
    webusername="&twUser"
    webpassword="&tw_Password";
run;
data &outTable;
   set &inTable;
run;

%macount(&outTable);
%mastatus( &_stpwork.status.txt );
```

```
%stpend;
```

10

Scheduling SAS Marketing Automation Campaigns

Overview of Scheduling

Scheduling jobs from SAS Marketing Automation is only a part of the overall scheduling process that takes place for all SAS jobs across your enterprise computing environment. See *Scheduling in SAS* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.htmlfor details about how to automate the scheduling and execution of SAS jobs in general, including jobs from SAS Marketing Automation.

SAS jobs are continually executed by SAS components in the SAS Business Intelligence environment. You can schedule a campaign, campaign group, or communication as a job for SAS Marketing Automation.

Many factors, such as personnel and priorities, affect job scheduling efficiency within an enterprise. To control the resources that are used by various individuals such as marketing campaign designers, your SAS environment uses a third-party scheduler. A scheduler application such as Platform LSF or an OS scheduling tool such as the Windows AT command or UNIX AT command automatically assigns the resources for job execution as specified by known criteria.

See *SAS Marketing Automation User's Guide* for details about scheduling a campaign in SAS Customer Intelligence Studio.

Scheduling Campaigns

The scheduling process is as follows.

1 A user creates a job (such as a campaign) using SAS Customer Intelligence Studio.

2 A user schedules the job by using either SAS Customer Intelligence Studio or the Schedule Manager plug-in for SAS Management Console, depending on the user’s rights to perform scheduling tasks.

3 The third-party scheduler (such as Platform LSF) triggers the job to be executed, and passes the instruction to the Launcher.

4 The Launcher sends the command in the format that the SAS Marketing Automation engine on the application server understands. The SAS Marketing Automation engine executes the job.

Note: If the middle tier is running in a different time zone from the client tier, the schedule is executed in the time zone of the middle tier. For more information, see “Architecture of the SAS Intelligence Platform” on page 2.

Verify the Scheduling Server Name

To schedule jobs from SAS Customer Intelligence Studio, verify that the correct name is specified for the third-party scheduling server.

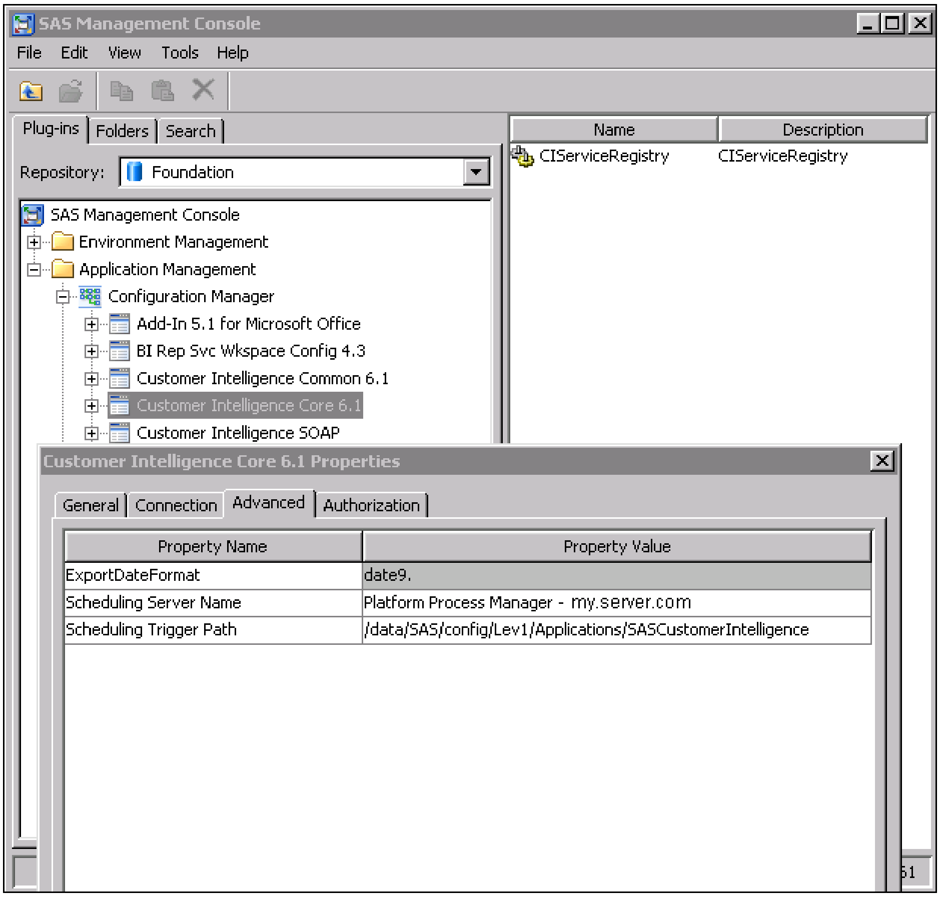
To specify the name and authorizations:

1 In SAS Management Console, on the **Plug-ins** tab, select **Application Management ▶ Configuration Manager**.

2 Right-click **Customer Intelligence Core 6.5**, and select **Properties**. Select the **Advanced** tab.

- If the third-party scheduler is Platform LSF, then specify all instances of **Scheduling Server Name** as `Platform Process Manager`.

Figure 10.1 *Customer Intelligence Core Properties*


SAS Management Console
File Edit View Tools Help
Plug-ins Folders Search
Repository: Foundation
SAS Management Console
Environment Management
Application Management
Configuration Manager
Add-In 5.1 for Microsoft Office
BI Rep Svc Wkspace Config 4.3
Customer Intelligence Common 6.1
Customer Intelligence Core 6.1
Customer Intelligence SOAP
Name
Description
CIServiceRegistry
CIServiceRegistry
Customer Intelligence Core 6.1 Properties
General Connection Advanced Authorization

Property Name	Property Value
ExportDateFormat	date9.
Scheduling Server Name	Platform Process Manager - my.server.com
Scheduling Trigger Path	/data/SAS/config/Lev1/Applications/SASCustomerIntelligence

- If the third-party scheduler is an OS scheduling tool, then specify **Scheduling Server Name** as `Operating System Services`.

Overview of Third-Party Schedulers

Platform LSF is the recommended third-party scheduler for SAS Marketing Automation jobs. However, Platform LSF is an optional application and is not bundled with SAS Marketing Automation.

Credentials are provided by the Customer Intelligence Scheduling Users group.

To schedule jobs from SAS Customer Intelligence Studio, you can also use an OS scheduling tool (such as the AT command) in either Windows or UNIX.

Comparing Platform LSF with an OS Scheduling Tool

Platform LSF does the following:

- provides the most flexibility for scheduling campaign groups.

- can schedule the optimization and campaign executions separately.
- can set an offset for individual campaigns so that they execute a specified time after the optimization.
- can add and schedule new campaigns to the group without rescheduling the entire campaign group.
- supports the separate flows that are created for each member campaign in Schedule Manager.
- supports the separate flows that are created for optimization.

An OS scheduling tool does the following:

- supports only a single time-based trigger, and does not support multiple criteria or job dependencies.
- requires that the optimization and campaign executions be scheduled as a single unit.

For information about the limitations of operating system scheduling, see *Scheduling in SAS* at http://support.sas.com/documentation/onlinedoc/sasmc/index.html

11

Command-Line Utilities

Encode Passwords

For greater security, encode passwords when you execute the command-line utilities. The parameters, including the encoded password, should be passed as a single parameter to the `sasciutils` command.

In the following example, parameters are passed to the Release Tables utility as a single parameter.

```
echo '-releasetables -userid myuserid -password
"{SAS002}3CD4EA1E0AF5A79500BC8ACC" -utilitylogfile c:/temp/cleanup.log
-deldate "Jun 01, 2013"'| sasciutils
```

For more information about encoding passwords, see *SAS Intelligence Platform: Migration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/tabs/install94.html.

Release Tables

Overview of the Release Tables Utility

The Release Tables utility releases some of the disk space that is occupied by out-of-date tables in the MATables library. Use this utility to delete the tables belonging to campaigns that are no longer needed. The affected campaigns are reset and their counts are cleared.

The user ID that executes Release Tables utility must have Write or Update permission for all of the campaigns whose counts should be cleared.

The Release Tables utility requires the Log on to Customer Intelligence applications capability. For more information, see “About Predefined Roles for SAS Customer Intelligence” on page 51.

CAUTION! Counts for released tables cannot be recovered When you delete out-of-date tables with this utility, the counts are cleared and cannot be recovered unless the campaign is executed again and new counts are generated. As a best practice, use the `-listonly` option first so that you can review and confirm the tables that you want to delete.

How to Use the Release Tables Utility

On Windows, the Release Tables utility is contained in the sasciutils.exe file. The typical location of this file is on the middle tier, at **`C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`**.

From the directory that contains the sasciutils.exe file, execute the command with the following syntax:

sasciutils_console.exe -*argument1* -*argument2*

Use the console version of the sasciutils.exe file so that debugging information is displayed in the command window. To create a sasciutils_console.exe version, make a copy of the sasciutils.exe file and rename it sasciutils_console.exe.

On UNIX, the Release Tables utility is contained in the sasciutils file. The typical location is on the middle tier, at **`SASHome/SASCustomerIntelligenceUtilities/6.5`**. From the directory that contains this file, execute the following command:

./sasciutils -*argument1* -*argument2*

The command takes the following arguments:

-releasetables
: specifies the Release Tables utility.

-userid
: is the user ID of the account that is used to execute the utility. This ID must have Write or Update permissions for all of the campaigns whose counts should be cleared.

-password
: is the password for the user ID.

-utilitylogfile
: specifies the location to which error messages and success messages are logged. If you specify a directory path, all directories in the path must exist before you execute the utility.

-deldate
: is the date to be used for determining whether a campaign is reset. Tables that were last modified before this date are deleted. The date must be in form of `month dd, yyyy`. The format must be in the MEDIUM ISO format that Java uses for the country where the product is installed.

-bcname
: specifies the name of the single business context to be processed. If this argument is not used, all business contexts are processed. If the business context name contains spaces, the name must be enclosed in double quotation marks (").

-listonly
: directs the utility not to delete the tables or reset the counts for the campaigns. All logging is performed. This argument is useful for determining which campaigns would be affected by the utility.

Release Tables Utility Examples

The following example produces a log file that lists the campaigns that would be affected in the NorthWestBanking business context if the Release Tables utility were executed. The log file will list campaigns with **last modified** dates that are prior to April 28, 2014.

This is the example on Windows:

```
"C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5\sasciutils_console.exe"
-releasetables -userid MyUserID -password MyPassword
-utilitylogfile c:/temp/tableCleanup.log -deldate "Apr 28, 2014"
-listonly -bcname NorthWestBanking
```

This is the same example on UNIX:

```
SASHome/SASCustomerIntelligenceUtilities/6.5/sasciutils
-releasetables -userid MyUserID -password MyPassword
-utilitylogfile /tmp/tableCleanup.log -deldate "Apr 28, 2014"
-listonly -bcname NorthWestBanking
```

The following example clears the counts and releases the tables for all campaign with a **last modified** date that is prior to June 1, 2016. The user has Update permission for all business contexts.

If you have permission to execute Windows programs, execute the following command from the **`C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`** directory:

```
sasciutils_console.exe -releaseTables -userid MyUserId
-password MyPassword -utilitylogfile c:/temp/cleanup.log
-deldate "Jun 01, 2016"
```

If you have permission to execute this utility from a UNIX shell, execute the following command from the **`SASHome/SASCustomerIntelligenceUtilities/6.5`** directory:

```
./sasciutils -releaseTables -userid MyUserId
-password MyPassword -utilitylogfile /tmp/cleanup.log
-deldate "Jun 01, 2016"
```

Add Channels

Overview of the Add Channels Utility

The Add Channels utility enables you to create additional channel definitions. The new channels will be available in definitions that are created after the channels have been added.

In order to use the Add Channels utility, you must have Write permission for all business contexts. This permission enables you to publish to the common data model and write to metadata. The sascisvc@saspw user ID usually has both authorizations.

To avoid performance issues, add only the channels that are actually needed.

Note: Removing a custom channel is not recommended. Any campaigns that use this channel will be affected. If you need to remove a channel, please contact SAS Technical Support at support.sas.com.

How to Use the Add Channels Utility

On Windows, the Add Channels utility is contained in the sasciutils.exe file. The typical location of this file is on the middle tier, at **`C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`**.

From the directory that contains the sasciutils.exe file, execute the command with the following syntax:

sasciutils_console.exe -addchannel *argument1 -argument2*

Use the console version of the sasciutils.exe file so that debugging information is displayed in the command window. To create a sasciutils_console.exe version, make a copy of the sasciutils.exe file and rename it sasciutils_console.exe.

On UNIX, the Add Channels utility is contained in the sasciutils file. The typical location is on the middle tier, at `SASHome/SASCustomerIntelligenceUtilities/6.5`. From the directory that contains this file, execute the following command:

sasciutils -addchannel *argument1 -argument2*

The command takes the following arguments:

-addchannel
: specifies the Add Channels utility.

-userid
: is the user ID of the account that is used to execute the utility. This user ID must have Write access to all of the business contexts that are created on the system. The sascisvc@saspw ID has this permission.

-password
: is the password for the user ID.

-utilitylogfile
: specifies the location to which error messages and success messages are logged. If you specify a directory path, all directories in the path must exist before you execute the utility.

-code
: is the channel code of the channel to be added. The code must be 1 to 3 characters in length. It must not begin with an underscore (_). The code cannot include leading or trailing blank spaces, forward slash (/) characters, backslash (\) characters, or any control characters.

-name
: is the new name to assign to the channel code. If the name contains spaces, it must be enclosed in double quotation marks. The length of the name that you provide must be between 1 and 30 characters. Names cannot include leading or trailing blank spaces, forward slash (/) characters, backslash (\) characters, or any control characters.

-skipcdm
: indicates that the new channel definition should not be published to the common data model.

  The channel is created in metadata, but no attempt is made to publish the channel to the common data model. Use this option if you have several business contexts and if one or more of the associated common data models is not accessible when you execute the Add Channels utility. This channel should not be used in campaigns until the channel can be published to the common data model.

  If you do not use this option, and if one or more of the common data models is not accessible, publish failures will occur when you execute the Add Channels utility. The channel is added to the metadata but it is not published to the common data models that are not accessible. If errors occur, check the SASCustIntelCore6.5.log and the Stored Process Server logs for more details.

Add Channels Utility Example

The following example adds a new channel with a code of **D2D** and name of **Door to Door Solicitation**.

```
"C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5
\sasciutils_console.exe"
-addChannel -userid MyUserId -password MyPassword
-utilitylogfile c:/tmp/channel.log -code D2D
-name "Door to Door Solicitation" -skipcdm
```

Update Export Paths

Overview of the Update Export Paths Utility

Campaigns and export definitions contain embedded paths for delimited export files. User-defined stored processes might also contain paths for exported files. If you migrate files from 5.4 or 5.4.1 to 6.1 and later releases, these paths are not updated automatically. The default paths might no longer exist. You can update the default paths for delimited export files.

This utility is optional. It can be used after migration to the current release.

How to Use the Update Export Paths Utility

After you have imported campaigns into the new environment, use the Update Export Paths utility to specify default paths for delimited export files.

On Windows, the Update Export Paths utility is contained in the sasciutils.exe file. The typical location of this file is on the middle tier, at **`C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`**. From the directory that contains the sasciutils.exe file, execute the command with the following syntax:

sasciutils.exe -useraction UpdateExportPaths -*argument1* -*argument2*

On UNIX, the Update Export Paths utility is contained in the sasciutils and sasciutils_console files. The typical location of these files is on the middle tier, at **`SASHome/SASCustomerIntelligenceUtilities/6.5`**. From the directory that contains these files, execute one of the following commands:

sasciutils -useraction UpdateExportPaths -*argument1* -*argument2*

sasciutils_console -useraction UpdateExportPaths -*argument1* -*argument2*

The command takes the following arguments:

-useraction UpdateExportPaths
: specifies the Update Export Paths utility.

-userid
: is the user ID for the metadata server.

-password
: is the password for the user ID.

-metaserver
is the fully qualified host name of the metadata server.

-metaport
is the metadata server port number. The default is 8561.

-metadomain
is the authentication domain (for example, DefaultAuth).

-oldpath
is the existing export directory path name.

-newpath
is the new export directory path name.

-matchrootpath
compares the old and new path names and specifies that new root path replace the old root path. For example, suppose that there are two old directory paths:

```
/data/export/campaign
/data/export/communications
```

The following arguments replace the root path in the path names:

```
-oldpath /data/export -newpath /install/export -matchrootpath
```

The result is that both of the directory paths now have the new root path name:

```
/install/export/campaign
/install/export/communications
```

This argument is optional. If you do not specify -matchrootpath, you must execute the Update Export Paths utility multiple times to replace the path for each directory.

-validatepath
validates the new directory path name. This argument is optional.

-logfile
specifies the log output file.

-threads
specifies the number of objects that are affected simultaneously. The default is 1.

-osccount
specifies the number of objects to migrate before reconnecting to Customer Intelligence core. This argument is optional.

-bclist
specifies a comma-separated list of business context names to process. All business contexts are processed by default.

-debug
enables the display of debugging text. This argument is optional.

Update Export Paths Utility Example

The following example updates the export path for the MAExport directory. A log file is created in the specified location. Debugging text is displayed. Two objects are affected simultaneously.

```
C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5\sasciutils.exe
-useraction UpdateExportPaths -userid myid -password mypassword -metaserver
metaserver21 -metaport 8561 -oldpath /local/install/SASServer/CI_Assets/Generic
Assets/MAExport -newpath D:\CI_Assets\Generic Assets\MAExport -logfile
c:\temp\mydir\UpdatePaths.log -debug -threads 2
```

Batch Export and Import Tools

Overview of the Batch Export and Import Tools

The batch export and import tools enable you to perform promotions from an operating system command line or from a batch script. You can use these tools to export and import campaigns and other objects between folders.

For information about the batch export and import tools, see the “Using the Batch Export and Import Tools” chapter in *SAS Intelligence Platform: System Administration Guide* at SAS Intelligence Platform Product Documentation.

How to Use the Batch Export Tool

The following code is an example of a Windows script that calls the batch export tools to export an object:

```
@ECHO OFF

SET SASHOME=C:\Program Files\<sas-config-dir>
SET SASFOLDER=/<path-to-object>
SET PROFILE=<metadata-server-connection-profile>

SET CURRENT=%CD%

IF %1.==. GOTO INVALID
IF %2.==. GOTO INVALID

CD %SASHOME%\SASPlatformObjectFramework\9.4

ExportPackage -profile "%PROFILE%" -subprop -package
"%USERPROFILE%\%~2.spk" -log "%USERPROFILE%\export_%~2.log"
-objects "%SASFOLDER%%~2(%~1)" %~3 %~4 %~5 %~6 %~7 %~8 %~9
GOTO END

:INVALID
GOTO END

:END
CD %CURRENT%
PAUSE
```

Modify the code and place it in a text file named ExportCIObject.cmd. From the directory that contains this file, execute the command with the following syntax:

```
ExportCIObject object-type object-name [-includeDep, -options]
```

The command takes the following arguments:

object-type
is the public object type. For more information, see Table 11.1 on page 181.

object-name
is the name of the object.

-includeDep
includes dependencies.

-options
are the other batch export tool options.

After you have exported the objects, modify the substitution properties file. For more information, see *SAS Intelligence Platform: System Administration Guide* at SAS Intelligence Platform Product Documentation.

Table 11.1 *Customer Intelligence Public Object Types*

Public Object Type	Name	Description
CIBusinessContext	Customer Intelligence business context	An environment with associated resources that is used to partition campaign, campaign group, and treatment artifacts from one another when running on the same server
CICalculatedItem (not visible in SAS Management Console **Folder** view)	Customer Intelligence calculated item	A Customer Intelligence calculated item
CICampaign	Campaign	A planned set of one or more communications that are directed at a selected group of customers.
CICampaignDefinition_Selection	Selection campaign definition	A template that specifies information about the underlying structure of a selection campaign
CICampaignGroup	Campaign group	A collection of campaigns
CICampaignGroupDefinition	Campaign group definition	A template that specifies information about the underlying structure of a campaign group
CICellNode (not visible in SAS Management Console **Folder** view)	Campaign cell node	A node that represents a group or a subgroup that is targeted by a campaign
CICommunicationDefinition	Campaign communication definition	A template that defines information about a campaign communication, such as its export definition, code, channel, and custom details
CICommunicationNode (not visible in SAS Management Console **Folder** view)	Campaign communication node	A campaign node that represents a specific marketing activity or communication with the consumer

CICustomDetailsGroup	Custom detail group	A collection of custom details that can be reused in more than one campaign, communication, treatment, or reply
CICustomDetailsTag	Custom detail tag	A user-friendly name for a custom detail
CIDiagram	Campaign diagram	A collection of nodes that make up a campaign process
CIDocument (not visible in SAS Management Console **Folder** view)	Campaign document	A collection of reports that summarize the information in campaigns
CIEnvironmentData	Customer Intelligence environment definition	The definition of a Customer Intelligence environment
CIExportDefinition	Campaign export definition	A collection of campaign information about the format of the data to be exported, the types of data to be exported, and other options
CIMasterSeedDefinition	Campaign seed definition	The format of a campaign seed list
CIMetadataGenerationDefinition	Customer-Intelligence-generated metadata definition	The format of customer-intelligence-generated metadata
CINodeDefinition (not visible in SAS Management Console **Folder** view)	Campaign node definition	The definition of a node in a diagram
CIReportExport	Customer Intelligence report export	A Customer Intelligence report link
CIResponseDefinition	Campaign response definition	A template that defines information about a campaign response
CISeedList	Campaign seed list	A list of individuals or organizations to whom communications are sent to verify that the campaign communications have been processed correctly
CISurrogateKey	Customer Intelligence surrogate key definition	A map of Customer Intelligence surrogate keys
CITreatment	Campaign treatment	A type of marketing communication. A treatment includes the format, creative content, and offer

How to Use the Batch Import Tool

The following code is an example of a Windows script that calls the batch import tools to import the objects in a package:

```
@ECHO OFF
```

```
SET SASHOME=C:\Program Files\<sas-config-dir>
SET SASFOLDER=<destination-pathname-of-imported-object>
SET PROFILE=metadata-server-connection-profile

SET CURRENT=%CD%

IF %1.==. GOTO INVALID

CD %SASHOME%\SASPlatformObjectFramework\9.4

ImportPackage -subprop "%USERPROFILE%\%~1.subprop" -newOnly -includeACL
-profile "%PROFILE%" -package "%USERPROFILE%\%~1.spk"
-log "%USERPROFILE%\import_%~1.log"
-target "%SASFOLDER%" %~2 %~3 %~4 %~5 %~6 %~7 %~8 %~9
GOTO END

:INVALID
GOTO END

:END
CD %CURRENT%
PAUSE
```

Modify the code and place it in a text file named ImportCIObject.cmd. From the directory that contains this file, execute the command with the following syntax:

```
ImportCIObject package-name  [-options]
```

The command takes the following arguments:

package-name
is the name of the package that contains the exported objects.

-options
are the batch import tool options.

Batch Export and Import Tool Examples

The following example exports an object that is named `my campaign` . Because the name contains spaces, it is enclosed in quotation marks. The public object type is CICampaign. Dependencies are included in the export.

```
ExportCIObject CICampaign "my   campaign " -includeDep
```

The following example imports a package that is named `my campaign` . Because the name contains spaces, it is enclosed in quotation marks. The `.spk` package extension is not included in the name.

```
 ImportCIObject "my   campaign "
```

Load Treatments

Overview of the Load Treatments Utility

The Load Treatments utility loads internal SAS Customer Intelligence treatments from external tables. The utility reads treatments and their custom details from SAS data sets. This utility does not update treatments within campaigns.

If an error is returned for a particular treatment, that treatment is skipped. The utility continues to load the rest of the treatments.

You can run the Load Treatments utility more than once on the same tables. The utility updates existing treatments that have the same name and folder. If there any differences from the previous version of a treatment, the treatment version number is incremented.

The treatment name and folder name form the unique ID of a treatment. You cannot use the Load Treatments utility to rename a treatment or to move a treatment to a new folder. Instead, delete the treatment and run the Load Treatments utility to load the treatment with the changed name or folder.

If there are multiple entries with the same name or folder in the treatment table, the last entry overwrites the first entry. The version number might be incremented.

If columns in the input tables are wider than the associated column in the common data model, the values are truncated in the common data model.

The Load Treatments utility creates a new version of the treatment with the data found in the treatment and custom detail tables that you specify on the command line. Only custom details that are in the custom detail table that you specify are in the new version of the treatment. You can use the Load Treatments utility to remove old custom details from a treatment by not specifying the table that contains those customer details.

How to Use the Load Treatments Utility

The Load Treatments utility is contained in the sasciutils.exe file on Windows and in the sasciutils file on UNIX. To write the results to the console rather than to the log, use sasciutils_console.exe or sasciutils_console.

On Windows, the sasciutils.exe file is typically installed on the middle tier in `C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`. From the Windows directory that contains the sasciutils.exe file, execute the command with the following syntax:

sasciutils -loadtreatments -*argument1* -*argument2*

On UNIX, the sasciutils file is typically installed on the middle tier in `./SASHome/SASCustomerIntelligenceUtilities/6.5`. From the UNIX directory that contains the sasciutils file, execute the command with the following syntax:

sasciutils -loadtreatments -*argument1* -*argument2*

The command takes the following arguments:

-loadtreatments
specifies the Load Treatments utility

-domain
specifies the domain of the user ID. This argument is optional.

-userid
is the user ID of the account that is used to execute the utility.

-password
is the password for the user ID.

-bcname
is the name of the business context.

-libname
is the library name of the tables that contain the treatments (for example, "Treatment Tables").

-treatmenttablename
is the name of the treatment table. The treatment table, custom detail table, and treatments to delete table must be in the same library.

-customdetailtablename
is the name of the custom detail table. The treatment table, custom detail table, and treatments to delete table must be in the same library.

-treatmentstodeletetablename
deletes the specified treatment table. The treatment table, custom detail table, and treatments to delete table must be in the same library.

Treatment Tables

Treatment tables should contain the following fields.

Table 11.2 Treatment Table Fields

Field	Description	Example
NAME	The name of the treatment. The name must be 60 characters or fewer in length. It cannot contain any forward slashes (/), backslashes (\), or control characters.	Little Red Corvette
FOLDER	The location relative to the business context root data folder. The specified folder must already exist. Otherwise, an error is returned for that treatment.	Treatments\September
DESCRIPTION	The treatment description	Toy car
CODE	The treatment code	LRC001
ASSET_LINK_LABEL	The display value of the treatment reference	Little Red Corvette

ASSET_LINK_URL	The URL of the treatment reference	www.sas.com/lrc.img
START_DATE	The activation date in the treatment properties	
END_DATE	The expiration date in the treatment properties	
HIDDEN	Specifies whether a treatment is hidden. `0` specifies that the treatment is not hidden. Any number other than `0` specifies that the treatment is hidden. If this field is not present, the treatment is not hidden.	1

Treatment Custom Detail Tables

Treatment custom detail tables should contain the following fields. For a particular treatment, the order of custom details in the table is the order of custom details in the loaded treatment. Multiple custom details with the same name, whether they have the same type, cannot be in the same treatment.

Table 11.3 Treatment Custom Detail Table Fields

Field	Description	Example
TREATMENT_NAME	The name of the treatment. The name must be 60 characters or fewer in length. It cannot contain any forward slashes (/), backslashes (\), or control characters.	Little Red Corvette
TREATMENT_FOLDER	The location of the treatment relative to the business context root data folder. The specified folder must already exist. Otherwise, an error is returned for that treatment.	Treatments\September
CUSTOM_DETAIL_NAME	The name of the treatment custom detail. The custom detail name must be unique within a treatment.	Size
TYPE	The default values are defined in UtilResources.properties. See below for more information.	Text

STRING_VALUE	This value is used as a default or initial value if the type is `Text` or `List`. If type is `List`, and multiple values are allowed, then this value is a list of strings that is separated by the list delimiter character that is specified in SAS Customer Intelligence. If the type is List, the value in the STRING_VALUE column must be in the specified list table. Otherwise, an error is returned.	1/4 in.
DOUBLE_VALUE	The value is used as a default or initial value if the type is `Numeric` or `Check box`. If type is `Check box`, then zero is `false` and nonzero is `true`. An error is returned if the DOUBLE_VALUE column is not numeric.	0.5
DATE_VALUE	This value is used as the default or initial date value if the type is `Date`. An error is returned if the DATE_VALUE column is not numeric.	15639
LINK_LABEL	This value is used as the default or initial display value if the type is `Link`.	Small Car
LINK_URL	This value is used as the default or initial URL value if type is `Link`.	www.sas.com/smallcar.img
REQUIRED_FLAG	Specifies whether a value is value required for this custom detail. `Y` specifies that the value is required. Otherwise, the value is not required. If the custom detail is static, the REQUIRED_FLAG is set to `Y`.	Y

COLUMN_NAME	The column name used for this custom detail in the UDF extension tables in the common data model. The values in the COLUMN_NAME field of the custom detail have the same restrictions as when the treatment is created in SAS Customer Intelligence. The length of a name must be between 1 and 60 characters. Names cannot include leading or trailing blank spaces, forward slash (/) characters, backslash (\) characters, or any control characters.	CAR_SIZE
TAG_NAME	Specifies whether a custom detail tag name is associated with this custom detail. `Y` specifies that a tag name is associated. Otherwise, a tag name is not associated.	CarSize
STATIC_FLAG	Specifies whether a custom detail is static or dynamic. If the custom detail is dynamic, the value of the custom detail can be overridden. `Y` specifies that the custom detail is static. Otherwise, the custom detail is dynamic. An error is returned if a UDF is static and has no default value. If the custom detail is static, the REQUIRED_FLAG is set to `Y`.	N
ALLOW_MULTIPLE_LIST_VALUES_FLG	If the type is `List`, this flag determines whether multiple items can be selected as the value of this custom detail. `Y` specifies that multiple items can be selected. Otherwise, multiple items cannot be selected.	

DYNAMIC_LIST_FLAG	If the type is `List`, this flag determines whether the available values for the list are read from a table each time that the value is selected or read in only once.
LIST_LIB_NAME, LIST_TABLE_NAME, LIST_VALUE_COLUMN, LIST_DISPLAY_VALUE_COLUMN	This field is used if the type is `List`. This field specifies where to read the values for the list. The values in the LIST_VALUE_COLUMN must be unique. This field can be empty.

The values for the **Type** field in the Treatment Custom Detail table are defined in **`CIUtilities/Source/Java/com/sas/analytics/crm/util/client/UtilResources.properties`**. The default values are as follows.

```
Utilities.LoadTreatments.numericType.txt = Numeric
Utilities.LoadTreatments.dateType.txt = Date
Utilities.LoadTreatments.textType.txt = Text
Utilities.LoadTreatments.checkboxType.txt = Check box
Utilities.LoadTreatments.listType.txt = List
Utilities.LoadTreatments.linkType.txt = Link
```

Type values are case-sensitive.

Treatments to Delete Table

The Treatments to Delete table should contain the following fields.

Field	**Description**
NAME	The name of the treatment. The name must be 60 characters or fewer in length. It cannot contain any forward slashes (/), backslashes (\), or control characters.
FOLDER	The location of the treatment relative to the business context root data folder. The specified folder must already exist. Otherwise, an error is returned for that treatment.

List Values Tables

In the list values tables, the name of the list value column must match the value of LIST_VALUE_COLUMN in the custom detail table. The name of the display value column must match the value of LIST_DISPLAY_VALUE_COLUMN in the custom detail table.

Load Treatment Utility Example

The following example loads the treatments from a CI_TREATMENT_IMPORT table that is in a library named TREATMENTLIBNAME. The CI_TREATMENT_CUST_DTLS table is also loaded. The results are printed to load_treatments.out.

```
sasciutils -loadtreatments -domain mydomain -userid myuserid -password
mypassword -bcname mybusinesscontext -libname "Treatment Tables"
-treatmenttablename CI_TREATMENT_IMPORT -customdetailtablename
CI_TREATMENT_CUST_DTLS > load_treatments.out
```

Write Response Definition Data to XML

Overview of the Response Definition Data Utility

The Response Definition Data utility writes response definition surrogate key (SK) and code data to XML so that response definitions can be migrated to the next release. You execute this utility after migration if writing the response definition data was omitted from the migration process.

How to Use the Response Definition Data Utility

The Response Definition Data utility is contained in the sasciutils.exe file on Windows and in the sasciutils file on UNIX. To write the results to the console rather than to the log, use sasciutils_console.exe or sasciutils_console.

On Windows, the sasciutils.exe file is typically installed on the middle tier in **`C:\Program Files\SASHome\SASCustomerIntelligenceUtilities\6.5`**. From the Windows directory that contains the sasciutils.exe file, execute the command with the following syntax:

sasciutils -useraction DumpResponseSK *argument1 -argument2*

On UNIX, the sasciutils file is typically installed on the middle tier in **`./SASHome/SASCustomerIntelligenceUtilities/6.5`**. From the UNIX directory that contains the sasciutils file, execute the command with the following syntax:

sasciutils -useraction DumpResponseSK -*argument1 -argument2*

The command takes the following arguments:

-useraction DumpResponseSK
: specifies the Response Definition Data utility.

-userid
: is the user ID of the account that is used to execute the utility.

-password
: is the password for the user ID.

-metaserver
: specifies the Domain Name System (DNS) name of the metadata server.

-metaport
: specifies the port for the metadata server.

-metadomain
specifies the domain for authentication (for example, DefaultAuth). This argument is optional.

-responseskfile
specifies the response SK map path and output XML filename.

-logfile
specifies the log path and output log filename.

-threads
specified the number of simultaneous objects that are affected. The default is 1.

-debug
enables debugging text. This argument is optional.

Response Definition Data Utility Example

The following example writes the output to `/install/temp/myresponsedump.xml`. Four threads are affected. The metadata server is myserver.na.sas.com. The port is 8561. The domain is DefaultAuth. A log file is written to `/install/temp/responsedump.log`. Debugging text is turned on.

```
sasciutils -useraction DumpResponseSK -userid mypassword -password myuserid
-metaserver myserver.na.sas.com -metaport 8561 -metadomain DefaultAuth
-responseskfile /install/temp/myresponsedump.xml -logfile
/install/temp/responsedump.log -threads 4 -debug
```

Execute Campaigns and Manage Information Maps

Overview of the Launcher

The Launcher application is a command-line utility that you use to perform the following tasks:

- Execute communications, campaigns, and campaign groups
- Query information maps and generate information map metadata tables
- Generate input variable reports.

In addition, the Launcher application runs automatically to schedule and execute campaigns and communications that are sent to the scheduler from SAS Customer Intelligence Studio.

The executable for the Launcher application is sasmalauncher.exe. On Windows, this file is typically located in **`C:\Program Files\SASHome\SASMarketingAutomationLauncher\6.5`**. On UNIX, it is typically located in **`SASHome/SASMarketingAutomationLauncher/6.5`** .

Launcher Command Syntax

The Launcher command uses the following syntax:

```
sasmalauncher.exe -d directive -u userid -p password
 -x business context[-n -q -v] [--] directive arguments
```

The command takes the following options:

`-d camp | campaigngroup | comm | file | imap`
specifies a directive

Note: If the `drgm` directive was used in a Launcher command in a previous release, it is converted to the `camp` directive in Release 6.5.

`-u` *userID*
specifies the ID of the user who is responsible for executing this command.

`-p` *password*
specifies the password for the user ID.

–g
assigns a user ID to identify the campaign. There might be a policy at your site to require passwords to be changed frequently. When the password is changed for the user who has scheduled an execution, the campaign continues to execute as scheduled. Each time a campaign is sent to the scheduler, a new ID is generated to identify the campaign.

`-x`
specifies the name of business context that corresponds to the campaign or communication to be executed. Spaces are allowed; the business context name is not case-sensitive.

`-v`
specifies informative messages in the log. If not specified, only errors are displayed.

`-n`
specifies that arguments are given by using the path and name rather than by identifier. The path and name must be provided in URI (Universal Resource Indicator) format unless option `-q` is specified.

`-q`
specifies that arguments are either quoted or have no spaces. URI translation of arguments is suppressed when this option is specified.

`--`
specifies the end of the options. All parameters after this are directive arguments.

directive arguments
is a list of arguments, separated by spaces.

See Table 11.6 on page 195 , Table 11.7 on page 195 and Table 11.8 on page 196 for the directives that are associated with each argument.

These are the possible arguments:

business context ID
uses the settings of the specified business context to regenerate metadata. Any saved settings are overridden by the directive arguments.

campID
campaign ID

camp path
folder path and campaign name

clearCache
information map cache to be cleared after map is generated

commID
communication ID

comm name
name of the communication in the campaign

file path
the path to the designated file

file op
the file operation. The values are `delete` or `write`.

file contents
holds the file contents to be written

genType
describes the method used. `1` is SQL. `2` is PROC SUMMARY.

groupId
campaign group ID

group mode
indicates the mode of campaign group execution. The values are `all`, `optimize`, or `execute`. The default is `all`.

group path
folder path and campaign group name

imap absolute path
specifies the absolute path of the information map (for example, **`/CI Assets/Information Maps/MABusinessContext`**)

For a list of arguments to *imap absolute path*, see Table 11.4 on page 193.

imap SBIP URL
specifies the SBIP URL of the information map. The SBIP URL uses the proprietary SAS Business Intelligence Platform (SBIP) protocol.

For a list of arguments to *imap SBIP URL*, see Table 11.4 on page 193.

imap URL
information map URL

typeArgs
space-delineated list of table-type arguments

The *campID* and *commID* arguments are identifiers that are assigned by the SAS Marketing Automation application server. They are not intended to be entered by the user on the command line, but they can be used if the `-n` option is not used.

When specified with *imap absolute path* or *imap SBIP URL*, `imap` takes arguments in the following table. The order is important for the `-method`, `-character`, `-numeric`, `-histogram`, `-datehistogram`, and `-univariate` options. The last entry on the command line takes precedence.

Table 11.4 *Arguments with imap absolute path, imap SBIP URL, or business context ID*

Argument	Parameter	Description

`-x` or `-bcname`	*business context name*	business context name for which metadata should be generated
`-method`	`SQL` | `Summary`	use SQL or the SAS SUMMARY procedure as the query method for all tables
`-character`	`SQL` | `Summary`	use SQL or the SAS SUMMARY procedure as the query method for character variable tables
`-numeric`	`SQL` | `Summary`	use SQL or the SAS SUMMARY procedure as the query method for numeric variable tables
`-histogram`	`SQL` | `Summary`	use SQL or the SAS SUMMARY procedure as the query method for histogram tables
`-datehistogram`	`SQL` | `Summary`	use SQL or the SAS SUMMARY procedure as the query method for date histogram tables
`-univariate`	univariate options. See Table 11.5 on page 194.	override information map settings and generate univariate metadata
`-mameta`	*library name*	override the setting for the MAMeta library
`—maxconcurrent`	*whole number*	override the **Maximum number of concurrent processes per metadata generation** setting in environment variables

The following table lists the options that can be used with the `–univariate` argument.

Table 11.5 *Options for -univariate Argument*

Option	Description
`Default`	let the application determine the best way to generate metadata
`InDatabase`	force MIN/MAX/MEAN generation, even for columns that might fail with numeric overrun errors

`Summary`	use PROC SUMMARY to calculate the MIN/MAX/MEAN. The column is brought into a SAS session to perform the calculation.
`AVGMinMax`	use the formula (MIN+MAX)/2 to calculate the mean.

In the following table, the `-n` option has not been specified.

Table 11.6 Arguments When the -n Option Has Not Been Specified

Directive	Arguments	Description
`camp`	*campID*	executes a campaign
`campaigngroup`	*groupID* [*groupMode*]	executes a campaign group
`comm`	*campID commID*	executes a communication
`imap`	*imap URL* [*clearCache*] [*typeArgs*] [*genType*]	generates all information map metadata tables, or only the specified types. This is the format that is automatically generated by SAS Customer Intelligence Studio.
`imap`	*imap absolute path* | *imap SBIP URL* | *business context ID*[–`x` or –`bcname`] [`-method`] [`-character`] [`-numeric`] [`-histogram`] [`-datehistogram`] [`-univariate`] [`-table`] [`-dataitem`] [`-mameta`] [`-maxconcurrent`]	generates information map metadata tables, using the specified arguments.

You can enter the *campPath*, *setPath*, and *commName* arguments on the command line if the `-n` option has been specified. If an application server that is not the standard is used, the application server also uses these arguments with the `-n` option.

In the following table, the `-n` option has been specified. All objects are identified by name or folder and name.

Table 11.7 Arguments When the -n Option Has Been Specified

Directive	Arguments	Description
`camp`	*camp path*	executes a campaign
`campaigngroup`	*groupID* [*group mode*]	executes a campaign group

`comm`	*camp path comm name*	executes a communication. *comm name* is case-sensitive if there is more than one communication with the same name in the campaign (for example, COMMI1 and comm1).

The `-n` option does not affect the `file` directive.

Table 11.8 *Arguments for the File Directive*

Directive	Arguments	Description
`file`	*file path* delete	deletes the file
`file`	*file path* write *file contents*	writes the file

Examples of Generating Information Map Metadata Tables

Here is an example of using the `imap` directive to generate information map metadata tables.

```
sasmalauncher.exe -d imap mybusinesscontextid
```

Information map metadata tables are generated for a business context named “mybusinesscontextid.” This command has the same result as clicking **Run Now** on the **Metadata** tab of the Business Context Properties window.

Here is an example of using the `imap` directive and SQL and the SAS SUMMARY procedure to generate metadata.

```
sasmalauncher.exe -d imap /CI Assets/Information Maps/Mybusinesscontext
-method Summary -character SQL
```

Except for character metadata, the SUMMARY procedure is used to generate metadata in the library that is defined for the information map. Character metadata is generated by SQL and is stored in the same library.

The metadata generation process might take a long time. You can generate metadata daily for specific data items that change frequently. You can then generate metadata weekly for other data items that change less often. Here is an example of generating metadata for specific data items.

```
sasmalauncher.exe -d imap /CI Assets/Information Maps/Mybusinesscontext
-method Summary -dataitem root.Gender -dataitem root.State
```

Each data item must be listed separately with the `–dataitem` option. The name of the data item must be fully qualified. Data item names are case-sensitive.

Here is an example of generating metadata for a specific table.

```
sasmalauncher.exe -d imap /CI Assets/Information Maps/Mybusinesscontext
-method Summary -table CMDM.MYTABLE
```

View the Launcher Command within a Job Flow

A Launcher command is automatically generated within the job flow that the user creates using the Schedule Manager plug-in for SAS Management Console.

To view the automatically generated Launcher command:

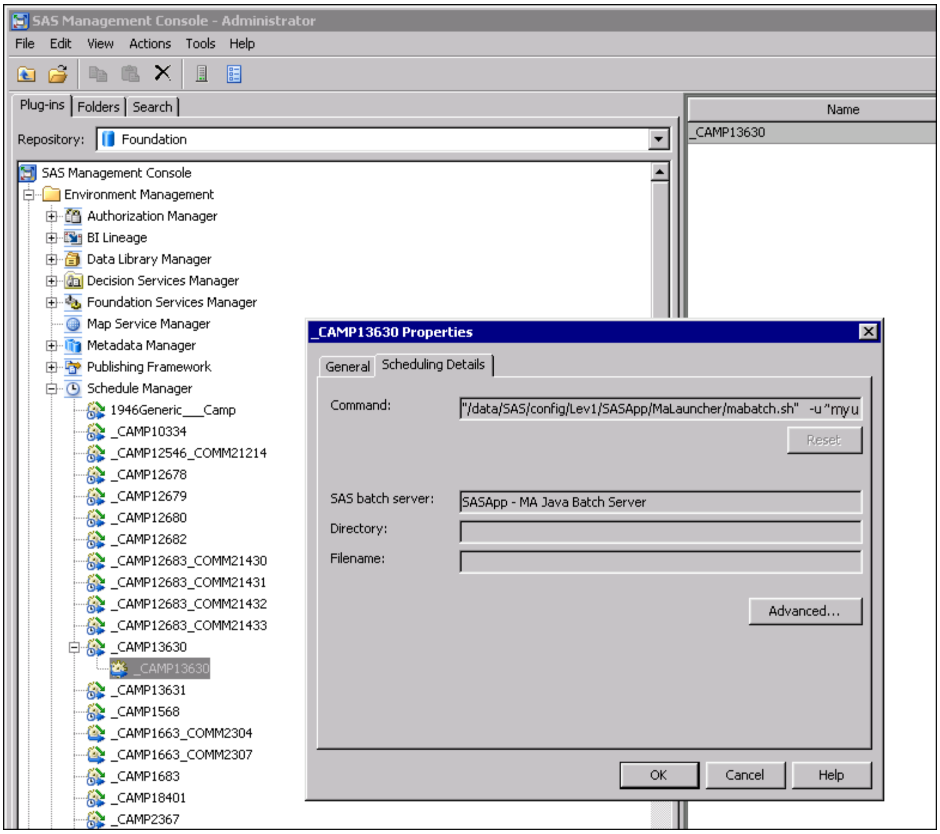
1 In SAS Management Console, create a flow using the Schedule Manager plug-in.

2 Select the flow. The name of the flow and a description appear in the right panel.

3 Double-click the name of the flow. The flow properties appear in the Properties window.

4 Select the **Scheduling Details** tab.

The system-generated Launcher command appears in the **Command** field, as shown:



_CAMP13630 Properties

Command: "/data/SAS/config/Lev1/SASApp/MaLauncher/mabatch.sh" -u "myu

SAS batch server: SASApp - MA Java Batch Server

Directory:

Filename:

Specify the Domain Name

The authentication process on Windows might require a domain name in addition to the user ID and password in the Launcher command. For example, you might be asked for a domain name when you schedule campaign groups for execution. You can require a domain name with the user ID by specifying the following option in `LevConfig\Web\WebAppServer`

\SASServer6_*cluster_number*\conf\wrapper.conf on every SASServer6 node where SAS Marketing Automation is deployed.

```
-Dsas.ci.CampaignGroupScheduleWithDomainQualifiedUserid
```

The domain name is then required with the `-u` option of the Launcher command with the following syntax.

```
-u userid@domain
```

Example of a System-Generated Launcher Command

Here is an example of a system-generated Launcher command:

```
c:\program files\SASHome\SASMarketingAutomationLauncher\6.5\sasmalauncher.exe -u
sasadm -p "my pass" -x "Functional Small Teradata"
-n -v -q -d comm  "Churn reduction\sub folder 1\Campaign 3" "mail_1"
```

The options `-n` and `-q` are specified in order to indicate that the campaign communication is executed by path and name, and that quotation marks are used to specify the arguments.

The Launcher is started by user sasadm with a non-encoded password of **my pass** in the business context **Functional Small Teradata**.

Communication **mail_1** in the campaign **Churn reduction\sub folder 1\Campaign 3** is being executed.

Verbose logging is specified.

Enable Logging for the Launcher

Overview of Logging

By default, the launcher logs are stored in the home directory of the user who issues the launcher command. The Launcher supports two types of logging:

General logging
: logs each execution of the Launcher. Logging information can be written to a different file for each execution or it can be appended to an existing file.

Command logging
: logs a particular execution of the Launcher. Command logging enables you to create logs for specific campaigns. The following considerations apply to command logging:

- Command logging runs in Append mode by default.
- Command logging can be used only if general logging is enabled.
- Command log files are written to the same directory as general log files. If a directory other than the default has been specified for general logging, then command logs are also written to the specified directory.
- Command log filenames are created based on the type of operation, the object name or ID, and the last directory name in the path.

The sasmalauncher.ini file is the Launcher initialization file that contains start-up and configuration parameters for the Launcher. To configure the Launcher's logging parameters, including log filenames and directories:

1. Locate the sasmalauncher.ini file.
2. Open sasmalauncher.ini. You see Java code similar to the sample file excerpt that is displayed.
3. Add command lines that specify the logging parameters that you want to change. Add each new command argument at the end of the file.

Locate sasmalauncher.ini

sasmalauncher.ini is located by default in the SAS Customer Intelligence installation directory:

- UNIX: **`.../SASMarketingAutomationLauncher/6.5`**
- Windows: *<Drive>*:**`\Program Files\SASHome\SASMarketingAutomationLauncher\6.5`**

Sample Sasmalauncher.ini File

Here is an excerpt from the sasmalauncher.ini file:

```
---
[properties]
MASTERPROP="C:\SASServer\SASHome\sassw.config"

startdir=<LAUNCHERDIR>
applogloc=
launchercmd=<JREHOME>
mode=console
JavaArgs_1=-Denv.definition.location="<SASENVIRONMENTSURL>"
JavaArgs_2=-Dma.launcher.logdir="ApplicationData\SAS\SASMarketingAutomationLauncher"
JavaArgs_3=-Djava.system.class.loader=com.sas.app.AppClassLoader
JavaArgs_4=-Dsas.ext.config="<SASHOME>\sas.java.ext.config"
JavaArgs_5=-Dsas.app.launch.config=picklist
JavaArgs_6=-Dsas.app.repository.path="<VJRHOME>\eclipse"
JavaArgs_7=-Dsas.app.class.path=.;"C:\SASServer\SASHome\SASMarketingAutomationLauncher\6.5\
JavaArgs_8=-Dsas.app.launch.picklist=picklist;"help\primary.picklist"
JavaArgs_9=
Classpath=-cp "<VJRHOME>/eclipse/plugins/sas.launcher.jar"
MainClass=com.sas.analytics.crm.ma.launcher.MALauncher
```

Add Command-Line Arguments (sasmalauncher.ini)

Use these Java Runtime Environment (JRE) commands within sasmalauncher.ini to perform the corresponding task. No spaces are allowed in the Java argument unless the path string is enclosed in quotation marks.

To direct general logging information to a log file without specifying a filename:
`-Dma.launcher.log` (for example, `JavaArgs_2=-Dma.launcher.log`)

A separate log file is created in the *`<SASHome>`*`\SASMarketingAutomationLauncher\6.5` directory for each Launcher call that is made. The filename is assigned as the time expressed in milliseconds (for example, 1268841617289.log).

In addition to the sassrv user, who must have Write permission, the user who might manually start the Launcher must also have Write permission to the directory.

To create a general log file with a specified filename:

`-Dma.launcher.log=<filename>` (for example, `JavaArgs_2=-Dma.launcher.log=february.log`)

If the file does not exist, then it is created. You must manually specify the .log extension in the filename if you want the extension to be used.

To append the general log output to an existing file:

`-Dma.launcher.log=<filename>` (for example, `JavaArgs_2=-Dma.launcher.log=january.log`)

You must manually specify the .log extension in the filename when you create the file if you want the file extension to be used in the filename.

To direct all logging information to a directory within the home directory of the Launcher application:

`-Dma.launcher.logdir="<dirname>"` (for example, `JavaArgs_2=-Dma.launcher.logdir="logs"`)

The directory must exist. Enclose the path in double quotation marks.

To direct all logging information to an absolute directory:

`-Dma.launcher.abslogdir="<AbsDirectory>"` (for example, `JavaArgs_2=-Dma.launcher.abslogdir="\\srshq\root\logs\launcher"`)

Another example is: `JavaArgs_2=-Dma.launcher.abslogdir="C:\deptZed\logs\launcher"`

The directory must exist. In addition to the sassrv user who must have Write permission, the user who might manually start the Launcher must also have Write permission to the directory.

To enable command logging in Append mode:

`-Dma.launcher.commandlogmode=` (for example, `JavaArgs_9=-Dma.launcher.commandlogmode=append`)

To enable command logging in Replace mode:

`-Dma.launcher.commandlogmode=replace` (for example, `JavaArgs_9=-Dma.launcher.commandlogmode=replace`)

To enable the decoding of U+ Unicode strings

`JavaArgs_xx=-Dma.unicode.marker=U+`

`xx` is the next number in the list of Java arguments.

Troubleshooting the Most Common Launcher Errors

These are the three most common error codes that might be generated by the Launcher:

- Unexpected Error
- Execution Failed
- Unable to Log onto Application Server

Unexpected Error, **Execution Failed**

First, check for errors in the logs of the SAS Marketing Automation engine on the application server.

Next, start the Launcher from a command prompt on the computer where the Launcher is installed, and determine whether the same error or a different error occurs.

Unable to Log onto Application Server

When the Launcher cannot log on to the application server, verify that the application server is running.

List of Error Codes

These are the error codes that can be returned by the Launcher. If a remedy or strategy for handling an error is available, then it is included in the error description.

01 Unexpected error

10 Invalid command line

Compare the generated Launcher command with the valid Launcher command-line arguments.

11 Invalid directive

The value for the directive in the **-d** option is not allowed.

12 Invalid argument scheme

The list of arguments either does not correspond to the directive or is not valid for the directive that is specified by the `-d` option.

13 Ambiguous argument scheme

The list of arguments is ambiguous for the selected directive that is specified with the `-d` option. This error occurs when a conflict exists because either the `-n` option is used to name arguments that are not provided in the correct format, or the named arguments do not correspond to the selected directive.

20 Unable to log on to application server

See "Unable to Log onto Application Server" on page 201.

21 Unable to acquire execution service

The Launcher cannot connect to the execution service within the application server. The Launcher uses an EJB (Enterprise JavaBeans) execution within the application server.

22 Invalid campaign group execution mode

23 Unable to switch to context

24 Unable to acquire Security Manager

30 Execution failed

The Launcher command failed to execute. In UNIX operating environments, this error indicates that the user does not have permission to create and update information map metadata tables.

31 Not executable

This error occurs only when SAS Marketing Automation executes campaigns. A campaign fails if a communication definition has not been assigned to one or more communications in the campaign. Open the related campaign in SAS Customer Intelligence Studio, and validate the campaign to determine what is required to make the campaign ready for execution. For more information, see *SAS Marketing Automation: User's Guide*.

32 Object to execute is locked by another user
The object to be executed is already opened by another client. The Locks category in the Administration workspace of SAS Customer Intelligence enables you to view and release objects that are locked, such as campaigns.

33 User does not have Execute permission on the object
The user ID that is specified in the Launcher command-line arguments does not have permission to edit the object to be executed. Schedule the job using a user ID that has Edit permissions, or assign to the user proper permissions to execute the job. See also "Unable to Log onto Application Server" on page 201, and *SAS Marketing Automation User's Guide*.

34 Identifier resolve failure
It was not possible to resolve the specified identifier. Either the identifier no longer exists, or the specified identifier is incorrect. In extreme circumstances, this error is displayed if the specified campaign name has been incorrectly resolved from corrupted metadata.

35 Optimization failed for a campaign in the group

37 Campaign group cannot be optimized

38 Campaign group does not have an optimization step

45 Metadata server out of memory
Either the metadata server is out of memory, or the application server is refusing the attempt to execute a non-recurring campaign more than one time.

46 Campaign is locked
In some circumstances, you might not be able to edit an object such as a campaign or treatment, even if you have Edit permission. To unlock an object so that you can edit it, select the Locks category in the Administration workspace in SAS Customer Intelligence. Contact the holder of the lock before you release the object.

50 Campaign name not found
The specified campaign name and path could not be found. Either the campaign no longer exists, the specified name is incorrect, or the path is incorrect.

52 Communication name not found
The specified communication name and path could not be found. Either the communication no longer exists, the specified name is incorrect, or the path is incorrect.

55 Campaign group path not found

60 File does not exist
The specified file does not exist.

61 File cannot be deleted
The specified file cannot be deleted.

62 File cannot be written
The specified file cannot be written.

70 Campaign cannot be marked for deployment
The campaign cannot be marked for deployment because it has not been approved.

90 Problem while logging off the application server

This error occurs after the object is successfully executed by the application server, but before the Launcher is disconnected. Look for error messages in the log file of the application server.

91 Problem releasing execution service

The Launcher cannot release the execution service within the application server. The Launcher uses an execution EJB (Enterprise JavaBeans) within the application server.

12

Backing Up and Restoring Data

The Difference between Backing Up Data and Archiving Data

The primary difference between the tasks of backing up data and archiving data is in their objectives. Backups are necessary for the operational recovery of IT business systems during a disaster (for example, a flood, lightning strike, or hard drive crash). Backups are typically stored for several weeks. A daily backup cycle can protect all types of business data, including structured and unstructured data.

Archives (archived data) are created in order to comply with legislation and good corporate compliance practices. Archived data is typically kept for a number of years. To create an archive, manual processes are created to take a backup copy from its normal backup cycle and store it in a fireproof safe for long-term storage. The backup media represents a snapshot of the business at that point in time. An organization might use archived data to protect structured business data (for example, business transactions or legal contracts).

General Strategy for Backing Up Data

Steps for Backing Up Data

As part of the maintenance of your SAS Business Intelligence Platform, you should perform the following steps:

- Back up your SAS metadata repository either nightly or weekly. SAS Customer Intelligence makes extensive use of metadata that is stored on the SAS Metadata Server and physical content that is stored on the SAS Content Server. Therefore, it is critical to perform regular backups of these servers.
- Back up the physical tables in the MATables and MAMisc libraries either nightly or weekly.
- Keep separate XML backup files for each of the information maps and campaign . Store the XML backup files on a separate file system in order to recover data that might be lost because of a hardware malfunction such as a hard drive crash.

  If one or more objects become corrupted, then you can re-import the XML backup file for a specific information map or campaign . In this way, you avoid restoring the entire SAS Metadata Server and SAS Content Server.

Note: All of the data from the SAS Metadata Repository, the SAS Content Server, the SAS Marketing Optimization data store, and the MAMisc libraries must be backed up at the same time.

Synchronizing the Backups for Campaign Environment

When you perform backups of the SAS Metadata Server, the SAS Content Server, and the MATables and MAMisc libraries, the backups must be synchronized in order to correctly restore the Campaign environment.

In order for the backups to be synchronized, the metadata server must be paused during all of these backups. Pausing the server (and setting it to an Offline state) ensures that no Campaign activity can occur between the creation of the backups.

Perform the backups in the following order:

1. Pause the SAS Metadata Server, and set it to an Offline state.
2. Back up either the SAS Content Server or the SAS Metadata Server.
3. Back up the remaining server (either the SAS Metadata Server or the SAS Content Server).
4. Back up the physical tables in the MATables and MAMisc libraries.
5. Resume the SAS Metadata Server.

For details about synchronizing your backups, as well as additional guidance for making backups, see *SAS Intelligence Platform: System Administration Guide*.

The guide is available at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Backing Up the SAS Metadata Server

The SAS Metadata Server includes a server-based facility that performs metadata server backups automatically, on a scheduled basis, without the need for administrator intervention.

For more information, see *SAS Intelligence Platform: System Administration Guide*, which is available at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Backing Up the SAS Content Server

To back up the SAS Content Server, follow these steps:

1. Make sure that the SAS Metadata Server has been paused and set to an Offline state.

2. Stop either the web application server or the SAS Content Server application.

3. Use operating system commands or third-party tools to copy all of the files and subdirectories from the following path:

   `SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository`

Backing Up the MATables and MAMisc Libraries

Be sure to back up the physical tables that are stored in the MATables and MAMisc libraries. To make the backups, use operating system commands or third-party tools.

This is the default location of the MATables library:<*config*>**`Lev1/Applications/SASCustomerIntelligence/CampaignManagement/data/MATables`**.

This is the location of the MAMisc library: **`<config>/Lev1/Applications/SASCustomerIntelligence/CustomerIntelligenceCommon/Data/MAMisc`**.

Backing Up Files Before Installing a New Version

Before you install a new version of SAS Customer Intelligence, back up the following files in the **`SASHome\SASFoundation\9.4\ma\sasmacro`** directory:

- mausrexp.sas
- mausrupl.sas

The headers in both file provide instructions for modifying the file. For information about the mausrexp.sas file, see "The mausrexp.sas Macro" on page 30. The mausrupl.sas file provides for additional actions to be performed on a table after the table has been uploaded. For example, you might want to generate statistics on Oracle tables that are a specified size.

Restore these files after installing the new version.

Restoring Data

All of the data from the SAS Metadata Repository, the SAS Content Server, the SAS Marketing Optimization data store, and the MAMisc libraries must be restored at the same time. For SAS Marketing Optimization, metadata must be restored as the same time as the data store.

Using SAS Integration Utilities to Back Up and Restore Campaigns

To create an XML backup file for a campaign, use the SAS Customer Intelligence Integration Utilities that are provided with SAS Customer Intelligence. For more information about the SAS Customer Intelligence Integration Utilities, see *SAS Customer Intelligence Integration Utilities* at http://support.sas.com/documentation/solutions/ci/index.html.

Although the installation of the SAS Customer Intelligence Integration Utilities is optional, they are typically installed with SAS Customer Intelligence. If the utilities have not been installed at your site, then they can be installed from the media that you received when you licensed SAS Customer Intelligence. Contact your on-site SAS support personnel for details.

Exporting SAS Packages

You can also back up files by exporting SAS packages. For more information, see “Importing and Exporting SAS Packages” on page 68.

13

Improving Performance

Activating the Bulk Load Facility

How the Bulk Load Facility Improves Performance

One of the fastest ways to insert large volumes of data into a relational database is to use the bulk-load capabilities of the database. When you use SAS/ACCESS, you specify the BULKLOAD data set option to use the bulk-load capabilities of database management systems (DBMS) such as Oracle, Teradata, and DB2. The bulk-loading method significantly enhances performance, especially when database tables are indexed.

When the BULKLOAD= data set option is enabled, SAS uses the BULKLOAD= option or calls the appropriate bulk-load facility for each DBMS.

Note: If you receive an

```
invalid database upload options
```

error message when you activate the bulk-load facility, check the log for information about the error. The information appears in the log file only when the trace level for logging is set. For more information, see "Summary of Logs for Troubleshooting" on page 230.

- For DB2: CLI LOAD bulk-load facility
- For Greenplum: A separate installed program named the Greenplum Loader (GPLOAD) executes program `gpfdist`. Specify these options:

  ```
  bulkload=yes   bl_format='csv'  bl_host='hostserver.mycompany.com'
  bl_port='port number' bl_protocol='gpfdist'  bl_delete_datafile=yes
  bl_null = '\n'
  ```

  The server that is specified by `bl_host` points to the SAS tier where `gpfdist` executes.
- For Netezza: SAS/ACCESS to Netezza uses the BULKLOAD= data set option.
- For Oracle: SQL*Loader bulk-load facility
- For PostgreSQL: Executes program `psql` through an upload parameter
- For SAP HANA: Use the data set option INSERTBUFF=32000 table_type=column. Do not use the BULKLOAD= data set option.
- For SQL Server: SAS/ACCESS to OLE DB and ODBC use the BULKLOAD= data set option.

  SAS/ACCESS to SQL Server does not support bulk-load. Specify INSERTBUFF=32676 in the BC Database Upload/Data Set options. Specify DATABASE UPLOAD/APPEND in the table options. Specify the LIBNAME option as CONNECTION=UNIQUE.
- For Teradata: FastLoad bulk-load facility

The bulk-load capability improves performance in the following areas:

- When you store contact history tables in a DBMS, performance might be reduced when you update a contact history table. The bulk-load facility improves performance by obtaining the list of IDs from SAS. The facility passes the IDs to temporary tables in the database. The facility then runs a query or updates the contact history tables in that database.
- The bulk-loading method is used during the export process within a Communication or Export node of a campaign. The bulk-loading method inserts the data into a temporary table within the DBMS so that the selection query is applied entirely within the DBMS.

The performance of the bulk-load facility can be affected by the connectivity between SAS Customer Intelligence and the database server, and by the size of the cell that is being loaded. For complex campaigns, the disk configuration of the database server might reduce performance. However, if only a single bulk-loading process is started for one campaign , then the disk configuration should not significantly affect performance.

For details about SAS/ACCESS, see the *SAS Intelligence Platform: Data Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Enable the Bulk Load Facility

Specify the following options in the Options section of the **Settings** tab in the Business Context Properties window. Example entries for each database are provided in "Example: Enabling the Bulk Load Facility" on page 211.

1. **Data set options** specifies options that are used in the SAS DATA step to create the temporary table. To enable the bulk-load capabilities of the database, specify *BULKLOAD=YES*. Additional options might also be appropriate, depending on which database you are using. For recommended options, see"Example: Enabling the Bulk Load Facility" on page 211. If a bulk-load facility is not available for your database, or if you choose not to use the bulk-load facility, see "Setting Data Set Options When a Bulk Load Facility Is Not Available" on page 213.

2. **Schema** specifies the location for the temporary tables. The following options are valid.

   - Specify your CICOMMON schema if you want the temporary tables to be created in the same schema that stores contact history.
   - Specify another schema to use for the temporary tables.
   - Leave the field blank if you want the temporary tables to be created in the user's default schema.
   - Users must have both Read and Write access to the specified schema, including the following:
     - the ability to create and drop tables and indexes
     - the ability to delete, insert, and update records

3. **Use temporary table capability of database** specifies whether the database's native temporary table capability is to be used to create temporary tables. The check box is deselected by default. If you are enabling the bulk-load facility of your database, do not select this check box. If you specify both *BULKLOAD=YES* and **Use temporary table capability of database**, processing fails.

Example: Enabling the Bulk Load Facility

Overview

The following examples provide recommended values for enabling the BULKLOAD option for your database. Enter the data set options, schema, and **Use temporary table capability of database** options on the **Options** tab of the Business Context Properties window, as described in "Enable the Bulk Load Facility" on page 211. For information about enabling the bulk-load facility of other databases, visit http://support.sas.com/documentation/index.html and search for the keywords "bulk-load facility."

To improve performance for all databases, add the following options to the marketingautomation_autoexec_usermods.sas file.

```
options dbidirectexec;
%let SYS_SQL_IP_SPEEDO=YES;
```

Example for DB2

The bulk-load facility for DB2 is called CLI LOAD.

- Data set options: BULKLOAD=YES BL_METHOD=CLILOAD
- Schema: <*schema-name*>
- Use temporary table capability of database: Not selected

Example for Netezza

- Data set options: BULKLOAD=YES BL_OPTIONS="’LOGDIR’C:\NETEZZA_LOGS\’"
- Schema: (None)
- Use temporary table capability of database: Not selected

Example for Oracle

The bulk-load facility for Oracle is called SQL*Loader. Temporary tables are stored in the specified schema.

- Data set options: BL_DIRECT_PATH=YES BULKLOAD=YES BL_DEFAULT_DIR="C:\ORACLE_LOGS\"
- Schema: *schema-name*
- Use temporary table capability of database: Not selected

Example for SQL Server - OLE DB Connection

- Data set options: BULKLOAD=YES
- Schema: (optional)
- Use temporary table capability of database: Not selected

Example for SQL Server - ODBC Connection

- Data set options: BULKLOAD=YES
- Schema: (optional)
- Use temporary table capability of database: Not selected

Examples for Teradata

The bulk-load facility for Teradata is called FastLoad. Temporary tables are stored in the specified schema.

- Data set options: BULKLOAD=YES SLEEP=1 TENACITY=1
- Schema: (None)
- Use temporary table capability of database: Not selected

If you are storing MATABLES in a separate database, add the schema name. Do not select **Use temporary table capability of database**.

You can also use the Teradata Parallel Transporter (TPT) to implement FastLoad.

To start FastLoad in the SAS/ACCESS interface using the TPT API, specify the TPT=YES data set option in a processing step that populates an empty Teradata table. For more information, see *SAS/ACCESS for Relational Databases: Reference* at http://support.sas.com/documentation/onlinedoc/access/index.html.

Setting Data Set Options When a Bulk Load Facility Is Not Available

For best performance, use the bulk-load facility. If a bulk-load facility is not available for your database, or if you choose not to use the bulk-load facility, then use the following instructions to specify data set options.

Specify Data Set Options without Bulk Loading

If a bulk-load facility is not available for your database, or if you choose not to use the bulk-load facility, specify the following options in the Options section of the **Settings** tab in the Business Context Properties window. Example entries are provided in "Examples: Data Set Options without Bulk Loading" on page 214.

1. **Data set options** specifies options that are used in the SAS DATA step to create the temporary table. To improve performance, specify the INSERTBUFF option.

2. **Schema** specifies the location for the temporary tables. The following options are valid.

   - If you want the temporary tables to be created in the user's default schema, leave the field blank.
   - If you are selecting the **Use temporary table capability of database** option, which uses the native temporary table capability of the database to create temporary tables, leave the **Schema** field blank. The **Schema** field is ignored if **Use temporary table capability of database** is selected.
   - Specify your CICOMMON schema if you want the temporary tables to be created in the same schema that stores contact history.
   - Specify another schema to use for the temporary tables.

   Users must have both Read and Write access to the specified schema, including the following:

   - the ability to create and drop tables and indexes
   - the ability to delete, insert, and update records

3. **Use temporary table capability of database** specifies whether the database's native temporary table capability is to be used to create temporary tables. The check box is deselected by default.

You should select this check box only if your site does not authorize users to create any additional temporary tables. When you select the check box, remember the following.

- The value in **Schema** is ignored, and the location of temporary tables is determined by the database.
- Do not specify `BULKLOAD=YES` in the **Data set options** field. If you specify both `BULKLOAD=YES` and **Use temporary table capability of database**, processing fails.

Examples: Data Set Options without Bulk Loading

Overview

The following examples show data set options that are recommended when a bulk-load facility is not available. Enter these values on the **Settings** tab of the Business Context Properties window, as described in "Setting Data Set Options When a Bulk Load Facility Is Not Available" on page 213.

Example for a Relational Database for Which a Bulk Load Facility Is Not Available

Because no schema is specified in this example, the user's default schema is used.

- Data set options: INSERTBUFF=1000
- Schema: (None)
- Use temporary table capability of database: Not selected

Example for a Relational Database When Users Are Not Authorized to Create Additional Tables

- Data set options: INSERTBUFF=1000
- Schema: (None)
- Use temporary table capability of database: Selected

Example for Amazon Redshift

- Data set options: INSERTBUFF=32676 DBCOMMIT=0
- Schema: (None)
- Use temporary table capability of database: Selected

SAS Grid Computing

In a SAS grid computing environment, SAS computing tasks are distributed among multiple computers on a network. All of the computers are under the control of SAS Grid Manager. In order for SAS Marketing Automation to take advantage of SAS grid processing, the MATables and MAMisc libraries must be in a shared location that is accessible by all of the computers in the grid cluster. The SAS Marketing Optimization libraries must also be in a shared location.

By default, SAS Marketing Automation and SAS Marketing Optimization libraries and the sasmalauncher executable file are installed in a shared location in the SASHome directory. If the SAS configuration directory is in a shared location and the MATables, MAMisc, and SAS Marketing Optimization libraries are installed in their default locations, the libraries are shared by all of the computers in the grid cluster. If the libraries are installed in another location, all of the files must be in a shared location that is accessible by the same path from all computers in the grid cluster. If SASHome is placed in a shared location during installation, then the sasmalauncher executable file is accessible by all of the computers in the grid cluster. If SASHome is not installed in a shared location, you must install the sasmalauncher executable file on every computer in the grid cluster.

For more information, see the SAS Grid Manager documentation at http://support.sas.com/documentation/onlinedoc/gridmgr/index.html.

Using Multiple Database Engines and Database Servers

Overview

It is possible to use multiple database engines or multiple database servers within a single business context. For example, the data that you use for selecting marketing targets can reside in both an Oracle database and a Teradata database.

Configuration

The use of multiple database servers or database engines within a single business context can affect performance. To prevent performance from deteriorating, keep in mind the following recommended configurations:

Shared database engine, multiple database servers

The same database BULKLOAD schema must be used for all database servers.

The same BULKLOAD data set options must be used for all database servers.

Multiple database engines

Configure the BULKLOAD options in the information map by using extended attributes that correspond to business context settings. You can use the following extended attributes:

- BulkloadOptions_*ENGINE* = *options*
- BulkloadSchema_*ENGINE* =*schema*
- BulkloadUseTemp_*ENGINE* =True/*other values*

BULKLOAD options that are set in the information map take precedence over conflicting settings on the **Options** tab of the Business Context Properties window.

If you specify an invalid engine, the engines that are referenced by the information map are listed in the SAS core log.

Using SAS Information Map Studio to Configure Context Tables

The server context tables in SAS Information Map Studio are named MAFS*database*.CUSTOMER. For example, the Oracle server context table is MAFSORA.CUSTOMER, and the Teradata server context table is MAFSTDAT.CUSTOMER. If you are using more than one database server, consider the following information when configuring the server context tables:

- Do not define joins between different server context tables. Joins can be created between tables that are on the same database server.
- Server context tables have their own subject definitions that are represented by unique Subject_ID_*x* extended attributes.

Use mapping to cross contexts. The value for the From_Subject_ID_ *subjectname1*_ To_Subject_ID_ *subjectname2* extended attribute is **equivalent**. This corresponds to value **OneToOne**, with optimization for multiple context support.

Equivalent subjects must have the following:

- the same column name, either the actual column name, or a column name that is specified with the OutputColumnName extended attribute.
- the same format. The DATE and DATETIME formats cannot be combined.

Exporting

You can export to a table in the same database engine as the subject for the Subject_Default extended attribute.

Publishing to the Common Data Model

In order to publish data from multiple databases, there must be a single master common data model that exists in only one schema.

Target common data model tables can be shared for equivalent subjects. For example, there might be two communications. One communication uses data from a Teradata database. The other communication uses data from an Oracle database. The contact history for the communications shares the same common data model table. You can separate the data by creating multiple data items with the required extended attributes to identify the common data model table as a target for multiple subjects.

Specifying JVM Memory Size

You must increase the size of JVM memory when you use large SAS Information Maps. Information maps are user-friendly metadata definitions of physical data sources that enable your business users to query a data warehouse in order to meet specific business needs. Large information maps can contain more than 25 million rows of data, which might require significant memory for loading and processing. You also must increase the size of JVM memory if you are using SAS Marketing Automation with SAS Real-Time Decision Manager.

Set JVM options in **Config\Lev1\Web\WebAppServer\SASServer6_*node_number*\bin\setenv.sh**. You must edit the setenv.sh file for every server in a cluster.

The recommended minimum JVM memory size is 4 GB for SASServer7 and 8 GB for SASServer6. If there are more than 200 campaigns and treatment campaigns, the recommended JVM memory size for SASServer7 is 8 GB.

For more information about JVM tuning options, see SAS Web Applications: Tuning for Performance and Scalability or contact the Enterprise Excellence Center (EEC).

Prevent Deadlock in DB2 Databases

When you execute multiple concurrent campaigns or single campaigns that contain multiple communications, it is possible for a DB2 database to deadlock. Deadlocks typically occur when the campaigns or communications publish data to the common data model or when you update contact history.

A deadlock has occurred if there are error messages in the MACore or stored process server logs that contain the word “deadlock.”

To prevent deadlock, specify the following DB2 database settings:

```
lock list = 20000
max locks = 60
```

Changing locks settings affects shared memory. The settings might need to be different, depending on available system resources.

Designing Campaigns and Tuning Databases

You can design campaigns for faster execution by simplifying the diagrams and reducing the number of nodes. For example, a campaign might have four identical Map nodes. Each Map node is linked to a different Communication node. If you replace the four Map nodes with a single Map node that is linked to the Communication nodes, execution time can be reduced by half.

You can also tune the database for improved performances. For example, you can speed execution by increasing the degree of parallelism in an Oracle database. Tuning options vary between databases. Performance depends on the workload. The database administrator should monitor the database management system during campaign execution and tune the database accordingly.

You can use the DataSetOptions custom property to apply parallelism to Oracle tables, see “Custom Properties (Map Level)” on page 87.

Treatment-Level Optimization

Treatment-level optimization in campaigns requires much more data storage than optimization at the cell level.

For information about system requirements and treatment-level optimization, see http://support.sas.com/resources/sysreq/index.html.

Monitoring SAP HANA Memory Usage

During the execution of SAS Marketing Automation campaigns, temporary tables are created in SAP HANA index server memory. These tables are deleted at the end of campaign execution. However, during campaign execution, the SAP HANA server might run out of memory if there is too much customer data to allow for the creation of the temporary tables.

To allow enough memory for the creation of temporary tables during campaign execution, limit the size of customer data in memory to one-third of the total memory that is available on the system.

The accumulation of translation tables in the SAP HANA heap allocator Pool/JoinEvaluator/TranslationTable can also cause the SAP HANA server to run out of memory. The default maximum number of translation tables is 2000. It is important to monitor SAP HANA memory usage. You can use the ALTER SYSTEM ALTER CONFIGURATION command to reduce the maximum number of translation tables that SAP HANA retains.

The following example reduces the maximum number of translation tables to 100.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET ('joins',
'translator_cache_size') = '100' WITH RECONFIGURE
```

For more information, see ALTER SYSTEM ALTER CONFIGURATION and the following SAP HANA notes:

http://service.sap.com/sap/support/notes/1998599
“SAP HANA: High Memory Consumption due to Translation”

http://service.sap.com/sap/support/notes/1969700
“SQL statement collection for SAP HANA - for further analysis of the objects in the Pool/JoinEvaluator/TranslationTable”

http://service.sap.com/sap/support/notes/1984422
“SAP HANA: Analysis of Out-of-memory (OOM) Dumps”

An SAP HANA user ID and password are required to access these notes.

Prevent Overflow Errors during Histogram Generation

If you use PROC SQL to generate histogram metadata from a Teradata or SAP HANA database, specify `UnivariateMethod=Summary`at the Map, Data, or Folder level of the information map. For more information, see Table 5.6 on page 96.

Troubleshooting JVM Out-of-Memory Error Messages

You might receive the following message: **Exception java.lang.OutOfMemoryError: requested** *size* **bytes**

JVM cannot expand its heap size if memory is completely allocated, and if swap space is not available.

- Increase the available swap space by allocating more of the disk for virtual memory. Set the values for maximum heap sizes by adding the following code to the settings for SASServer6 and SASServer7.

  In **`Config/Lev1/Web/WebAppServer/SASServer6_`*`node_number`*`/bin/setenv.sh`**, add the following command: `-Xmx8092`

  In **`Lev1/Web/WebAppServer/SASServer7_1/bin/setenv.sh`**, add the following command: `-Xmx4096`.

  Note: The suggested values in the maximum heap size commands might need to be modified for your environment.

- Make sure that SASServer6 and SASServer7 are each allocated 4 gigabytes of memory during installation.

14

Troubleshooting

Overview of Troubleshooting

This chapter presents a systematic approach to resolving problems in SAS Customer Intelligence.

Types of Problems

The problems that an administrator might need to troubleshoot fall into three general categories:

- failure to execute a scheduled campaign
- an error message in SAS Customer Intelligence Studio
- unexpected behavior while creating or executing a scheduled campaign

Regardless of the type of problem, begin by collecting basic information. A clear definition of the problem often makes it easier to find a solution.

Problem Definition

A good problem definition includes the following information:

1. What is the exact sequence of steps that produced the problem?
2. What are you seeing (or not seeing) in the logs or in the results that demonstrates the problem?
3. Can the problem be reliably reproduced, or is it intermittent?
4. When did this problem first occur?
5. Has the same sequence of steps been followed before without producing the problem? If so, what might have changed?
   a. information map
   b. data format, volume, and refresh period
   c. inclusion of nulls or spaces in the data
   d. time zone change
   e. amount of space or memory available in work or export directories
   f. permissions and authorizations within SAS products, within the database, or on an operating system directory
   g. library definitions and options within the SAS Management Console Data Library Manager plug-in
   h. operating system environment variables or settings, such as locale

6 Does this problem occur for all users? If not, can this user's problem be replicated on another user's computer?

7 What is the environment in which the problem occurs?

   a operating system of the compute tier, middle tier, and client tier

   b database type and version for the underlying data warehouse (for example, Oracle, SQL Server)

   c version and hot fix level of the SAS Customer Intelligence products

8 Is the problem occurring in a development, test, or production environment?

9 How many transactions per second are being sent at the time of the performance problem?

10 What is the expected response time under this load?

11 What is the range of response times that are you seeing under this load?

12 What steps, if any, have you taken in order to isolate the problem to a particular component or flow? Examples are particular activities, treatments, flows, and campaigns, or particular queries to the database.

When you have collected this information, you are ready to take the following troubleshooting steps.

Failure to Execute a Scheduled Campaign

Pinpoint the Source of the Scheduling Problem

Troubleshooting a failure to execute a scheduled campaign requires isolating the source of the problem. There are three primary sources for this sort of problem: the scheduled campaign itself, the command that is used to run the campaign, or the scheduling software.

If the following suggestions do not provide a solution, see "Common Sources of Error" on page 226.

Is the Campaign Causing the Problem?

To test whether the campaign is causing the problem:

1 Ensure that no other users have the campaign open in SAS Customer Intelligence Studio.

2 Open the campaign in SAS Customer Intelligence Studio and select **Execute Now**.

3 Verify that any communications, exports, and code nodes ran correctly.

4 Check the SASCustIntelCore6.5.log for errors.

5 If the campaign does not execute properly, follow the steps outlined in "Common Sources of Error" on page 226.

6. If the campaign executes without error, then it is not the source of the problem. In this case, proceed to the next step, in which you test the command that is used to execute the campaign.

Is the Command Causing the Problem?

To test whether the command that is used to execute the campaign is the source of the problem:

1. Log on to SAS Management Console as an administrator and open the Schedule Manager plug-in.
2. Identify the flow that is causing the problem that you are troubleshooting.
3. View the properties for the job within that flow.
4. Go to the **Scheduling Details** tab.
5. Copy the command to launch the campaign. Here is an example of such a command:

```
D:\<SASConfig>\Lev1\SASApp\MaLauncher\mabatch.bat  -u "sasdemo" -p {base64}T4Jbb24xMjM=
-x "My Business Context" -d camp -e -v -- BICG2BAFRYLMWAAA
```

6. Log on to the operating system as a user who has permissions to execute this command. Execute the command at a command line.
7. Check for errors. For a list of logs, see “Summary of Logs for Troubleshooting” on page 230. Verify that any exports completed successfully. If they did complete successfully without error, then the problem is with scheduling, rather than with execution in SAS Marketing Automation.
8. If the execution did not complete successfully, check for the following common problems:
   - Insufficient permissions on the operating system. The user ID that is charged with executing MALauncher does not have permissions in the operating system to use the cmd.exe executable file.
   - Insufficient permissions within LSF. The SAS ID that is used is unable to execute a scheduled flow. Remedy this problem by using the `lspasswd` executable command. Instructions for using this command are in the Platform LSF documentation.
   - Improperly installed MALauncher. Check to see that sasmalauncher.exe and sasmalauncher.ini files exist.
   - The LSF administrator user ID might not be defined in the SAS Management Console User Manager plug-in.
   - Errors in the log. Check the *.log files in your home directory for errors. If you do not see logs, you can get an MALauncher log file by adding the following line to the malauncher.ini file:

     ```
      - JavaArgs_8=-Dma.launcher.log
     ```

     You must rerun your test after adding this line in order to get a log.

- If jobs appear in a running state only, and there is no other activity, one job might be delaying other jobs. To resolve this issue by using the lsfadmin user, enter the following Platform LSF command on the command line:

  ```
  bhist -l All > <file.name>
  ```

  When you check the file, it contains the list of running jobs. You can use the `bkill` *<jobid>* Platform LSF command to kill the queued jobs and then reschedule your job if it does not start via this process.

Is the Scheduling Software Causing the Problem?

To test the scheduling software, schedule a job outside SAS Customer Intelligence Studio, such as a SAS Data Integration Studio job. You can increase LSF logging by setting `LSF_LOG_MASK=LOG_DEBUG4` in the lsf.conf file in the \ **`LSF_SHARE`** directory.

Next Steps

When the problem is identified and you have followed the steps above, it still might not be clear what is causing the problem. In this case, try searching SAS Notes at http://support.sas.com/notes/index.html, using different combinations of keywords or error messages that you have encountered.

If the solution to the problem remains unclear, use the problem definition guideline in "Problem Definition" on page 222 to send a description of the problem to SAS Technical Support and including the following:

1. tests that you have run, the time at which you ran them, and their results
2. logs that you generated as you tested
3. a copy of the SASCustIntelCore6.5.log (MACore log) from the time period in which you were running the tests

Error Dialog Box

Troubleshoot the Error Dialog Box

The next category of problem to troubleshoot is the Error dialog box. When you encounter an error, take the following steps:

1. Click **Details** in the Error dialog box to view the full error message.
2. Copy the full error message by clicking on the dialog box, and then select Ctrl+A and Ctrl+C.
3. Open a text editor and paste this error message onto the page. This enables you to view the full text of the error. It also enables you to send the full text of the error to SAS Technical Support, if necessary.

4. Record the time of the error so that you can identify any associated error messages in the logs.

5. Use the error message and the problem description to search for known issues in the SAS Notes at http://support.sas.com/notes/index.html. You might need to try different combinations of keywords or different sections of the error message.

6. If the error message in the dialog box does not help you find the source of the problem, SASCustIntelCore6.*x*.log is the next place to look. By default, this log is written to <*SASPlanName*>`Lev1/Web/Logs/SASServer6_1` on the middle tier machine.

   a. Open this log and search for the word ERROR. The timestamps for the log messages help identify which messages might be associated with the error dialog box.

   b. If you do not find any relevant error messages in the log, then search for the word WARN.

7. If SASCustIntelCore6.*x*.log does not provide the information that you need in order to identify and correct the error, there are more logs and logging options that are described in "Summary of Logs for Troubleshooting" on page 230. See also "Common Sources of Error" on page 226.

Next Steps

After following these steps, it still might not be clear what is causing the problem. The next step is to search for an applicable SAS Note at http://support.sas.com/notes/index.html, using different combinations of keywords or error messages that you have encountered.

If the solution to the problem remains unclear, use the problem definition guideline in "Problem Definition" on page 222 to send a description of the problem to SAS Technical Support. Include information such as the following:

- The time at which the logging levels were increased and the problem was re-created
- The steps used to re-create the problem
- The text of the error in the error dialog box
- The most recent SASCustIntelCore6.*x*.log
- The Stored Process Server logs
- The web application server logs
- The SAS Content Server log

Common Sources of Error

Often, the error message in SASCustIntelCore6.*x*.log gives you enough information to find the source of the problem and correct it. Here are some of the most common errors that you might find as well as information about how to address them.

Locked Asset

To address this error, release the locked object in the Locks category in the Administration workspace of SAS Customer Intelligence. For more information, see *SAS Marketing Automation: User's Guide*.

Other Locked Objects

When more than one process attempts to use a particular database table, SAS data set, or other resource, the contention can cause a variety of problems. One of the most common problems is a process or campaign that does not appear to ever complete. This is most common when you use Process nodes or Custom nodes. The short-term solution is to stop all intentional SAS Customer Intelligence and SAS Business Intelligence processes, and then check to see whether SAS processes are still running. If there appear to be processes that are not responding, kill them. Restart the compute tier. The long-term solution is to ensure that campaigns that contain Process nodes are run in a way that avoids contention.

Library Reference That Was Not Correctly Defined

To address this error:

1. Log on to SAS Management Console as an administrator.
2. Correct the reference by going to the Data Library Manager plug-in, right-clicking on the library in question, and selecting **Properties**.

The library definitions that are used by SAS Customer Intelligence can be viewed in SASCustIntelCore6.*x*.log. Often a careful examination of the LIBNAME statements in this log can help you pinpoint problems with reading or writing data.

Missing Table in MATABLES

To address this error, make sure that all users have closed the problematic campaign.

If the error continues to occur, see "Upstream Linked Diagrams That Are Open, Incomplete, or Corrupt" on page 227.

Upstream Linked Diagrams That Are Open, Incomplete, or Corrupt

A campaign can fail if any of its upstream linked diagrams are open or inaccessible. SASCustIntelCore6.*x*.log can contain a number of different errors that are associated with this problem, including a missing table in the MATABLES library. Problems with contention for a particular table in MATABLES can also exist if multiple campaigns attempt to use the same linked upstream diagram at precisely the same time. If you suspect that this might be causing your problem, take the following steps:

1 Clear the counts for the problem campaign and for all campaigns that are upstream from it.

2 Check that the upstream campaign can be executed successfully.

3 Stagger execution of all downstream campaigns so that their use of the upstream diagram does not cause contention.

Campaign Designers Close SAS Customer Intelligence in the Middle of a Process That They Have Initiated

This error can cause processes not to respond on the middle tier or the compute tier. These orphaned processes might continue to consume resources. This can slow performance or even cause out-of-memory type errors. To remedy this error:

1 Close all open instances of SAS Customer Intelligence.

2 Close all SAS Business Intelligence applications.

3 Shut down the compute tier (object spawner, workspace servers, stored process servers).

4 Check for remaining SAS processes that should not be running. Kill any processes that are not responding.

5 Restart the compute tier.

6 Restart the middle tier.

Slow Performance When Multiple Campaign Designers Are Working in SAS Customer Intelligence Studio or When Multiple Campaigns Are Executed

To remedy this:

1 Review the environment settings in the Environment Variables category in the Setup workspace in SAS Customer Intelligence Studio. This setting controls how many concurrent tasks (for example, SQL statements and stored processes) are allowed to be running in a single campaign execution. This setting has an effect only when campaigns are complex enough to use multiple concurrent processes when they execute. If this number is too high, then when multiple campaigns execute concurrently, each campaign can consume too many resources. The correct number depends on the complexity of the campaigns at your site as well as on the computing resources available. The maximum number of concurrent processes might need to be adjusted up or down.

2 Check database performance, including regeneration of indexes.

3 Check the number of Multibridge stored process instances in the SASApp folder in the Server Manager plug-in in SAS Management Console. The default is 3, but you might need up to 30 in a heavily used environment. This

number depends on the amount of memory that is available on the compute tier as well as the number of concurrent campaigns and SAS Customer Intelligence Studio sessions.

4 Check the maximum number of pooled workspace server sessions that are allowed in the SASApp folder in the Server Manager plug-in in SAS Management Console. The default is 10, but you might need up to 30, again depending on hardware resources and the number of concurrent campaigns and SAS Customer Intelligence Studio sessions.

For more information about improving slow performance, see Troubleshooting and tuning SAS Stored Process Server and SAS Pooled Workspace Server load-balancing for SAS Marketing Automation.

Stored Process Server Error in SASCustIntelCore6.*x*.log

Look for additional details in the Stored Process Server logs. These files are located on your SAS Stored Process Server machine at <*SASPlanName*>`Lev1/Web/Logs/SASServer6_x`. Check the three most recent SAS Stored Process Server log files. There is generally one log for each SAS Stored Process Server process.

Error Codes

SAS Marketing Automation components generate error codes in the following ranges.

Table 14.1 Error Code Ranges

Component	Error Code Range
Metadata Generation	000 - 299
Export Node	300 - 399
Cluster Node	400 - 401
Update History in Common Data Model	700 - 799
Publish to Common Data Model	800 - 899
Statistical Size Estimator	900 - 999
Create Tables, Run Stored Processes, Update Counts	1000 - 1199

Summary of Logs for Troubleshooting

Log Types

You can often gather more detailed error information by increasing the logging level, replicating the problem, and then reviewing the recent additions to the logs. SASCustIntelCore6.*x*.log is often the first log for which you should try increasing the logging level. If the SASCustIntelCore6.*x*.log messages indicate that the problem is actually with a stored process, turn on additional logging for the SAS Stored Process Server instead.

SASCustIntelStudio.log is the log for the SAS Customer Intelligence web client.

When you run SAS Customer Intelligence in a production environment, logging is generally kept to a minimum in order to maximize performance. Increase the logging level only when you are troubleshooting. When you finish troubleshooting, return the server environment to the previous level of logging to avoid degrading performance. Here are the tables that display information about the various types of logs.

Table 14.2 Core Log

Filename	SASCustIntelCore6.*x*.log
Location	<*SASPlanName*> `Lev1/Web/Logs/SASServer6_1`
Log Contents	Java errors, some stored process information, LIBNAME statements that were executed. This is the primary log for SAS Customer Intelligence.
How to Increase the Logging Level	Set the logging level on the Logging page in SAS Customer Intelligence Studio. For more information, see “Set Logging Level” on page 45.

Table 14.3 SAS Stored Process Server

Filename	SASStoredProcessServer_yyyy-mm-dd_pid.log
Location	<*SASPlanName*> `/Lev1/Web/Logs` or <*SASPlanName*>`/Lev1/StoredProcessServer/Logs`
Log Contents	Code and results from the nodes that are executed by the SAS Stored Process Server.
How to Increase the Logging Level	See SAS Note 34114 at http://support.sas.com/kb/34/114.html.

Table 14.4 *Object Spawner*

Filename	ObjectSpawner_yyyy-mm-dd_pid.log
Location	*<SASPlanName>* `\Lev1\ObjectSpawner\logs` or *<SASPlanName>*`\Lev1\Logs`
Log Contents	Start-up information for the stored process servers and pooled workspace servers. Execution permission errors.
How to Increase the Logging Level	See SAS Note 34114 at http://support.sas.com/kb/34/114.html

Table 14.5 *SAS Workspace Server*

Filename	WorkspaceServer_yyyy-mm-dd_pid.log
Location	*<SASPlanName>* `\Lev1\SASApp\PooledWorkspaceServer\Logs` or *<SASPlanName>*`\Lev1\Logs`
Log Contents	Queries that are created by nodes. Because the size of this log increases quickly, enable debug logging for this log only when troubleshooting a specific problem.
How to Increase the Logging Level	See SAS Note 34567 at http://support.sas.com/kb/34/567.html.

Table 14.6 *Platform LSF*

Filename	Several filenames exist as specified during the installation of Platform LSF.
Location	*<LSF-Install-Dir>* `\log`
Log Contents	Execution steps of the Platform LSF Process Manager.
How to Increase the Logging Level	See SAS Note 24428 at http://support.sas.com/kb/24/428.html.

Table 14.7 *SAS Content Server*

Filename	SASContentServer.log
Location	*<SAS-configuration-dir>* `/Lev1/Web/Logs`
Log Contents	Login issues; permission issues.

How to Increase the Logging Level	Not applicable.

Other Logs and Files

If you find that you need more information than is available in the logs above, the following might provide more details:

- SAS Metadata Server log
- event log from the Windows operating system
- database logs
- installation and configuration logs. See SAS Note 49799 at http://support.sas.com/kb/49/799.html.
- the hosts file. The hosts file on Windows can be found in either `C:\WINNT\system32\drivers\etc` or `C:\Windows\system32\drivers\etc`

Troubleshooting in a Clustered Environment

Logs for SAS Intelligence Platform services such as SAS Visual Analytics might not be stored on the same middle-tier node where the user session occurred. These logs are not assigned to a particular node, and can be on any node in the server cluster.

Trace Level for Logging

Configuring in SAS Customer Intelligence

The trace level for logging is set on the Environment Variables page in SAS Customer Intelligence.

For more information, see “Set Logging Level” on page 45.

Gathering Logs

It is helpful to apply the following best practices when gathering logs:

- All logs that you gather must be from the same time period, and that time period must include a time during which the problem occurred.
- Include a timestamp for when the problem occurred. Ensure that this timestamp is included in the period covered by all of the logs.
- If you have it, include a timestamp for when the problem first happened. An approximate time is acceptable.
- Include a brief description of the components that are installed on each machine.
- Label all of the logs such that technical support will know which machine each log came from.

For problems with SAS Customer Intelligence Studio, gather the following logs:

1. from `.../Lev1/web/logs/` of the middle-tier machine, where SAS Customer Intelligence applications are installed (for example, SASServer6):
   - SAS Marketing Automation Core log (for example, SASMarketingAutomationCore`version-number`.log), also called the SAS Customer Intelligence Core log in later versions
2. from `.../Lev1/web/logs/` of the middle-tier machine or machines, where SAS Decision Services applications are installed (for example, SASServer7):
   - SAS Decision Services Design Server log (for example, SASDecisionServicesDesignServer`version-number`.log)
3. web application server log (for example, SystemOut.log) for SASServer6
4. web application server log (for example, SystemOut.log) for SASServer7

For problems with SAS Decision Services, gather the following logs:

1. from `.../Lev1/web/logs/` of the middle-tier machine or machines, where SAS Decision Services applications are installed (for example, SASServer7):
   - SAS Decision Services Design Server log (for example, SASDecisionServicesDesignServer`version-number`.log)
   - SAS Decision Services Engine Server log or logs (for example, SASDecisionServicesEngineServer`version-number`.log)
2. web application server log or logs (for example, SystemOut.log) for SASServer7
3. from the machine or machines where SAS Federation Server is installed:
   - all files in `SASHOME\SASDataFluxFederationServer\var\log`
   - all files in `SASHOME\SASDataFluxAuthenticationServer\var\log`

For DS2 Activities and Model Score results problems, send the logs listed above for SAS Decision Services problems and the logs from the DS2 code when run in Base SAS, if applicable.

If you are unable to gather the logs below, send all files under the following directories for each machine in the implementation. Include the timestamps for when you encountered the problem and the labels that indicate which machine each log came from.

- `/Lev1/logs`
- `/Lev1/web/logs`
- `SASHOME\SASDataFluxFederationServer\var\log`
- `SASHOME\SASDataFluxAuthenticationServer\var\log`
- and the web application server logs directories

Including the following information in your initial helps expedite the solution to your problem:

1. How much memory is available on each of the machines in this implementation?
2. How many cores (CPUs) are available on each of the machines in this implementation?

3. If you have done any resource monitoring, either per machine or per process, please include information about your observations, labeled by machine and by process, if applicable.

Contacting SAS Technical Support

If you are still unable to find and correct the source of the problem, contact SAS Technical Support. Include the following information:

- Problem description. (see “Problem Definition” on page 222).
- The SASCustIntelCore6.*x*.log that contains messages from the time the problem occurred.
- All SAS Stored Process Server logs that contain messages from the time the problem occurred.
- The web application server log or logs from the time the problem occurred.
- The SAS Content Server log from the time the problem occurred.

When gathering logs, ensure that you follow the steps that are presented in “Gathering Logs” on page 232.

If you are unable to gather specific logs, send all files under the following directories for each machine in the implementation. For each log, include the timestamps and labels that identify which logs came from which machines.

- /Lev1/logs
- /Lev1/web/logs
- web application server logs directories

You can open a technical support track by emailing the problem description and attachments to support@sas.com. Files that are too big to email can be sent by FTP, using the instructions that are available at http://support.sas.com/kb/20/941.html. Alternatively, you can start a technical support track by using the web form at http://support.sas.com/ctx/supportform/createForm.

Appendix 1

%CI2LASR Output Table Data Dictionary

The %CI2LASR macro extracts common data model data to an output table. The data is used for reports. For more information, see “Extract Data” on page 154.

The following table does not include columns that are built from the customized extension (_EXT) tables. Those columns depend on the common data model implementation at your site.

Table A1.1 %CI2LASR Macro Output Table Data Dictionary

Column Name	Type	Length	Format	Informat	Common Data Model Source
CAMPAIGN_MAX_OFFER	NUM	8	16.2	16.2	CI_CAMPAIGN,COLUMN: MAX_BUDGET_OFFER_AMT
COMMUNICATION_MAX_OFFER	NUM	8	16.2	16.2	CI_COMMUNICATION, COLUMN: MAX_BUDGET_OFFER_AMT
CAMPAIGN_CD	CHAR	120	$120.	$120.	CI_CAMPAIGN, COLUMN: CAMPAIGN_CD
RUN_DTTM	NUM	8	7.	7.	CI_CAMPAIGN, COLUMN: RUN_DTTM
CAMPAIGN_VERSION_NO	NUM	8	7.	7.	CI_CAMPAIGN, COLUMN: CAMPAIGN_VERSION_NO
CAMPAIGN_CURRENT_VER_FLG	CHAR	4	$4.	$4.	CI_CAMPAIGN,COLUMN: CURRENT_VERSION_FLG
CAMPAIGN_NM	CHAR	240	$240.	$240.	CI_CAMPAIGN,COLUMN: CAMPAIGN_NM
CAMPAIGN_FOLDER_TXT	CHAR	4096	$4096.	$4096.	CI_CAMPAIGN,COLUMN: CAMPAIGN_FOLDER_TXT
CAMPAIGN_START_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_CAMPAIGN,COLUMN: START_DTTM
CAMPAIGN_END_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_CAMPAIGN,COLUMN:END_DTTM
LAST_MODIFIED_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_CAMPAIGN,COLUMN: LAST_MODIFIED_DTTM
LAST_MODIFIED_BY_USER_NM	CHAR	240	$240.	$240.	CI_CAMPAIGN,COLUMN: LAST_MODIFIED_BY_USER_NM
APPROVAL_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_CAMPAIGN,COLUMN: APPROVAL_DTTM
APPROVAL_GIVEN_BY_NM	CHAR	240	$240	$240	CI_CAMPAIGN,COLUMN: BUSINESS_CONTEXT_NM

BUSINESS_CONTEXT_NM	CHAR	240	$240.	$240.	CI_CAMPAIGN,COLUMN: BUSINESS_CONTEXT_NM
CAMPAIGN_MAX_BUDGET_AMT	NUM	8	16.2	16.2	CI_CAMPAIGN,COLUMN: MAX_BUDGET_AMT
COMMUNICATION_OCCURRENCE_NO	NUM	8	7.	7.	CI_COMMUNICATION,COLUMN: COMMUNICATION_OCCURRENCE_NO
COMMUNICATION_NM	CHAR	240	$240.	$240.	CI_COMMUNICATION,COLUMN: SUBJECT_TYPE_NM
SUBJECT_TYPE_NM	CHAR	240	$240.	$240.	CI_COMMUNICATION,COLUMN: SUBJECT_TYPE_NM
COMMUNICATION_CD	CHAR	120	$120.00	$120.	CI_COMMUNICATION,COLUMN: COMMUNICATION_CD
COMMUNICATION_STATUS_CD	CHAR	12	$12.	$12.	CI_COMMUNICATION,COLUMN: COMMUNICATION_STATUS_CD
COMMUNICATION_RECURR_TYPE_CD	CHAR	12	$12.	$12.	CI_COMMUNICATION,COLUMN: COMMUNICATION_RECURR_TYPE_CD
COMMUNICATION_START_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_COMMUNICATION,COLUMN: START_DTTM
COMMUNICATION_END_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_COMMUNICATION,COLUMN: END_DTTM
EXPORT_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_COMMUNICATION,COLUMN: EXPORT_DTTM
COMMUNICATION_MAX_BUDGET_AMT	NUM	8	16.2	16.2	CI_COMMUNICATION,COLUMN: MAX_BUDGET_AMT
PACKAGE_CD	CHAR	128	$128.	$128.	CI_CELL_PACKAGE,COLUMN: PACKAGE_CD
TOTAL_MARKETING_ CONTACT_CNT	NUM	8	NLNUM16.		The total number of attempted contacts for the package, minus any failed contacts (such as bounced email messages). This value is a count of subjects such as Customer, Household, or Account.

TOTAL_ATTEMPTED_ CONTACT_CNT	NUM	8	NLNUM16.		The total number of attempted contacts that were sent to the channel for the package. This value is a count of subjects such as Customer, Household, or Account.
CHANNEL_CD	CHAR	12	$12.	$12.	CI_CELL_PACKAGE,COLUMN: CHANNEL_CD
CONTROL_GROUP_TYPE_CD	CHAR	12	$12.	$12.	CI_CELL_PACKAGE,COLUMN: CONTROL_GROUP_TYPE_CD
HO_TOTAL_ATTEMPTED_CONTACT_CNT	NUM	8	NLNUM16.	NLNUM16.	The total number of attempted contacts that were sent to the channel for the holdout package. This value is a count of subjects such as Customer, Household, or Account.
HO_TOTAL_MARKETING_CONTACT_CNT	NUM	8	NLNUM16.	NLNUM16.	The total number of attempted contacts for the holdout package minus any failed contacts (such as bounced email messages). This value is a count of subjects such as Customer, Household, or Account.
HO_MARKETING_CELL_CD	CHAR	128	$128.	$128.	CI_MARKETING_CELL,COLUMN: MARKETING_CELL_CD
HO_MARKETING_CELL_NM	CHAR	240	$240.	$240.	CI_MARKETING_CELL,COLUMN: MARKETING_CELL_NM
HO_CONTROL_GROUP_TYPE_DESC	CHAR	1024	$1024.	$1024.	CI_CONTROL_GROUP_TYPE , COLUMN: CONTROL_GROUP_TYPE_DESC
TOTAL_RESPONSE_CNT	NUM	8	NLNUM16.		Calculated from CI_RESPONSE_HISTORY.
TOTAL_RESPONSE_VAL_AMT	NUM	8			Calculated from CI_RESPONSE_HISTORY.
TOT_SUCCESS_ ESPONSE_CNT	NUM	8	NLNUM16.		Calculated from CI_RESPONSE_HISTORY.
TOT_SUCCESS_RESPONSE_VAL_AMT	NUM	8			Calculated from CI_RESPONSE_HISTORY.
HO_TOTAL_RESPONSE_CNT	NUM	8	NLNUM16.		Calculated from CI_RESPONSE_HISTORY.
HO_TOTAL_RESPONSE_VAL_AMT	NUM	8			Calculated from CI_RESPONSE_HISTORY.

HO_TOT_SUCCESS_RESPONSE_CNT	NUM	8	NLNUM16.		Calculated from CI_RESPONSE_HISTORY.
HO_TOT_SUCCESS_ RESPONSE_VAL_AMT	NUM	8			Calculated from CI_RESPONSE_HISTORY.
TREATMENT_CD	CHAR	128	$128.	$128.	CI_TREATMENT,COLUMN: TREATMENT_CD
TREATMENT_NM	CHAR	240	$240.	$240.	CI_TREATMENT,COLUMN: TREATMENT_NM
TREATMENT_REFERENCE_TXT	CHAR	2000	$2000.	$2000.	CI_TREATMENT,COLUMN: TREATMENT_REFERENCE_URL
TREATMENT_REFERENCE_URL	CHAR	2000	$2000.	$2000.	CI_TREATMENT,COLUMN: TREATMENT_REFERENCE_URL
TREATMENT_CURRENT_VER_FLG	CHAR	4	$4.	$4.	CI_TREATMENT,COLUMN: CURRENT_VERSION_FLG
TREATMENT_START_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_TREATMENT,COLUMN: START_DTTM
TREATMENT_END_DTTM	NUM	8	DATETIME20.	DATETIME20.	CI_TREATMENT, COLUMN: END_DTTM
TREATMENT_VERSION_NO	NUM	8	7.	7.	CI_TREATMENT,COLUMN: TREATMENT_VERSION_NO
CAMPAIGN_GROUP_CD	CHAR	120	$120.	$120.	CI_CAMPAIGN_GROUP,COLUMN: CAMPAIGN_GROUP_CD
CAMPAIGN_GROUP_NM	CHAR	240	$240.	$240.	CI_CAMPAIGN_GROUP,COLUMN: CAMPAIGN_GROUP_NM
CAMPAIGN_GROUP_FOLDER_TXT	CHAR	4096	$4096.	$4096.	CI_CAMPAIGN_GROUP,COLUMN: CAMPAIGN_GROUP_FOLDER_TXT
CAMPAIGN_GROUP_TYPE_DESC	CHAR	1024	$1024.	$1024.	CI_CAMPAIGN_GROUP_TYPE , COLUMN: CAMPAIGN_GROUP_TYPE_DESC
CAMPAIGN_STATUS_DESC	CHAR	1024	$1024.	$1024.	CI_CAMPAIGN_STATUS , COLUMN: CAMPAIGN_STATUS_DESC

CAMPAIGN_TYPE_CD	CHAR	12	$12.	$12.	CI_CAMPAIGN,COLUMN: CAMPAIGN_TYPE_CD
CAMPAIGN_TYPE_DESC	CHAR	1024	$1024.	$1024.	CI_CAMPAIGN_TYPE,COLUMN: CAMPAIGN_TYPE_DESC
COMMUNICATION_RECURR_TYPE_DESC	CHAR	1024	$1024.	$1024.	CI_COMMUNICATION_RECURR_TYPE, COLUMN: COMMUNICATION_RECURR_TYPE_DESC
COMMUNICATION_STATUS_DESC	CHAR	1024	$1024.	$1024.	CI_COMMUNICATION_STATUS, COLUMN: COMMUNICATION_STATUS_DESC
CHANNEL_NM	CHAR	240	$240.	$240.	CI_CHANNEL,COLUMN: CHANNEL_NM
MARKETING_CELL_CD	CHAR	128	$128.	$128.	CI_MARKETING_CELL,COLUMN: MARKETING_CELL_CD
MARKETING_CELL_NM	CHAR	240	$240.	$240.	CI_MARKETING_CELL,COLUMN: MARKETING_CELL_NM

Glossary

authentication
See client authentication

bulk load
to load large amounts of data into a database object, using methods that are specific to a particular DBMS. Bulk loading enables you to rapidly and efficiently add multiple rows of data to a table as a single unit.

business context
a designation that identifies the information that a user can access. Data access is restricted to only the information that is required for a specific business need. A user can have access to more than one business context.

campaign
a planned set of one or more communications that are directed at a selected group of customers or potential customers for a commercial goal.

client authentication
the process of verifying the identity of a person or process for security purposes.

communication definition
a template that defines information about a communication, such as its export definition, code, channel, and custom details. Communication definitions are defined in SAS Management Console.

contact history
a record of the groups of individuals or organizations that have been identified to be contacted for a communication.

counts metadata
metadata that SAS Customer Intelligence Studio generates when some types of node diagrams are displayed. Counts metadata differs from the SAS metadata in the SAS Metadata Repository. Counts metadata typically consists of a list of values and a count of how often each value occurs in the database.

custom property
a user-defined property that you can create for information maps, relationships, data items, and filters, and for folders in an information map.

data item
in an information map, an item that represents either data (a table column, an OLAP hierarchy, or an OLAP measure) or a calculation. Data items are used for building queries. Data items are usually customized in order to present the data in a form that is relevant and meaningful to a business user.

data set

See SAS data set

database management system

a software application that enables you to create and manipulate data that is stored in the form of databases.

DBMS

See database management system

encryption

the act or process of converting data to a form that is unintelligible except to the intended recipients.

foreign key

a column or combination of columns in one table that references the corresponding primary key in another table. A foreign key must have the same attributes as the primary key that it references.

format

See SAS format

holding area

a temporary area of memory in which the output from a query of an SPD Server table is stored.

information map

a collection of data items and filters that provides a user-friendly view of a data source. When you use an information map to query data for business needs, you do not have to understand the structure of the underlying data source or know how to program in a query language.

Integrated Object Model

the set of distributed object interfaces that make SAS software features available to client applications when SAS is executed as an object server.

IOM

See Integrated Object Model

Java Naming and Directory Interface

a standard extension to the Java platform that enables developers to create applications that can interact with a number of different naming services and directory services, such as the Domain Name System (DNS) and the Lightweight Directory Access Protocol (LDAP).

Java Virtual Machine

a software application that can execute Java bytecode, on either a client or a server, enabling Java programs to be run on many different hardware and software platforms.

JNDI

See Java Naming and Directory Interface

job

a collection of SAS tasks that can create output.

JVM
See Java Virtual Machine

listen port
in a network, a communications endpoint at which a server listens for requests for service from the client application.

log
See log file

log file
a file in which information about software processing is recorded as the processing occurs. A log file typically includes error messages and warning messages, but it can also include informational messages and statistics such as the number of records that have been processed or the amount of CPU time that a program required.

macro variable
a variable that is part of the SAS macro programming language. The value of a macro variable is a string that remains constant until you change it.

metadata
descriptive data about data that is stored and managed in a database, in order to facilitate access to captured and archived data for further use.

metadata server
a server that provides metadata management services to one or more client applications.

middle tier
in a SAS business intelligence system, the architectural layer in which web applications and related services execute. The middle tier receives user requests, applies business logic and business rules, interacts with processing servers and data servers, and returns information to users.

multi-part key
a primary key that includes multiple columns from multiple tables. The columns uniquely identify rows from the tables.

object spawner
a program that instantiates object servers that are using an IOM bridge connection. The object spawner listens for incoming client requests for IOM services. When the spawner receives a request from a new client, it launches an instance of an IOM server to fulfill the request. Depending on which incoming TCP/IP port the request was made on, the spawner either invokes the administrator interface or processes a request for a UUID (Universal Unique Identifier).

operating environment
a computer, or a logical partition of a computer, and the resources (such as an operating system and other software and hardware) that are available to the computer or partition.

pass-through facility
See SQL pass-through facility

plug-in
a file that modifies, enhances, or extends the capabilities of an application program. The application program must be designed to accept plug-ins, and the plug-ins must meet design criteria specified by the developers of the application program.

primary key
a column or combination of columns that uniquely identifies a row in a table.

record-level locking
locking at the record level in a table or data set. The user who owns the lock has exclusive access to a single record, while other users can access other records in the same table or data set.

response
the reaction that an individual has to a campaign, such as requesting a quote, making an inquiry, opening an email message, or buying the product.

SAS data set
a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views.

SAS format
a type of SAS language element that is used to write or display data values according to the data type: numeric, character, date, time, or timestamp.

SAS Management Console
a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Repository
a container for metadata that is managed by the SAS Metadata Server.

SAS SPD Server
a SAS Scalable Performance Data Server. An SPD Server restructures data in order to enable multiple threads, running in parallel, to read and write massive amounts of data efficiently.

SAS Stored Process
a SAS program that is stored on a server and defined in metadata, and which can be executed by client applications.

SAS system option
a type of SAS language element that is applied to any of a number of operations during a SAS session. System options can control SAS session initialization, SAS interactions with hardware and software, and input and output processing of SAS files.

SAS Workspace Server
a SAS server that provides access to Foundation SAS features such as the SAS programming language and SAS libraries.

SASSPDS
the SAS engine that provides access to the SAS SPD Server.

schema
a map or model of the overall data structure of a database. A schema consists of schema records that are organized in a hierarchical tree structure. Schema records contain schema items.

spawner
See object spawner

SPD Server
See SAS SPD Server

SQL pass-through facility
the technology that enables SQL query code to be passed to a particular DBMS for processing.

stored process
See SAS Stored Process

stored process server
a server that can execute SAS programs as required by one or more client applications.

subject
the hierarchical level to which selection criteria are applied. For example, a record can be selected because of household, customer, or account criteria. All three hierarchical levels are subjects.

symbolic variable
See macro variable

system option
See SAS system option

thread
the smallest unit of processing that can be scheduled by an operating system.

unique key
one or more columns that can be used to uniquely identify a row in a table. A table can have one or more unique keys.

Index