THE POWER TO KNOW®

SAS® Web Analytics 5.4

Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS® Web Analytics 5.4: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Web Analytics 5.4: Administrator's Guide

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

1st electronic book, February 2011
2nd electronic book, August 2012

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

Contents

About This Book

Audience

The *SAS Web Analytics: Administrator's Guide* is recommended for the experienced administrator and is written for users whose responsibilities and interests include the following:

- setting up security at a site
- collecting Clickstream data and setting up data marts
- understanding the SAS Web Analytics ETL (extract, transform, load) process and how jobs are run and customized
- using SAS Business Intelligence tools with SAS Web Analytics
- working with search engine bid data (SEBD) data sources
- integrating SAS Customer Intelligence capabilities with SAS Web Analytics
- setting up and configuring OLAP cubes

What's New in the SAS Web Analytics 5.4 Administrator's Guide

Overview

SAS Web Analytics 5.4 has the following new features and enhancements:

- new jobs for preparing analytic data
- new aggregates that support internal searches and capabilities for creating custom aggregates
- support for data that originates from SAS Real-Time Decision Manager
- support for making reports available in a production environment
- new tables added to the data model that support internal searches, session attribution, and goals
- new capabilities for creating OLAP cubes
- new procedures for setting internal and external domains
- new documentation enhancements for migrating Web marts and for creating Clickstream jobs

New Jobs for Preparing Analytic Data

Several new jobs prepare subsets of data that are used for analysis in various reports: Weba_1600_Active_Funnels, Weba_2200_Create_Path_Data, Weba_2500_Insight_Monitor1, and Weba_2501_Prep_Insight_Monitor.

There are also three new staging jobs that are used to prepare analytic data for loading into the data mart tables:

- The Weba_1200_Warehouse_Staging_Tables job prepares data for loading into the Web data mart.
- The Weba_1220_Goal_Staging_Tables job prepares data for loading goal tables.

- The Weba_1230_Link_Staging_Tables job prepares data for loading link tables.

New Capabilities for Creating Custom Aggregates

You can now configure options in the Aggregate transformation to enhance performance. The aggregate transformation also contains options you can customize for columns to be aggregated and statistics to be derived.

Support for Data from SAS Real-Time Decision Manager

SAS Web Analytics enables you to analyze the success of campaigns that are created and executed through SAS Customer Intelligence products.

Marketing campaigns that are created in SAS Marketing Automation can be delivered to the customer by SAS Real-Time Decision Manager as an offer for single channel and multi-channel campaigns.

SAS Real-Time Decision Manager sends identifiers for customers, campaigns, and Web site offers so that its campaign data can be collected and analyzed. The ETL job that updates the response history data is Weba_2550_MA_Response_History_Update.

Support for Making Reports Available in a Production Environment

If you are moving a Web mart from a test environment into a production environment, you can use a new method to modify an attribute of the Web mart object. Modifying the attribute makes the Web mart available in drop-down lists for SAS Web Report Studio and in SAS Web Analytics reports.

New Tables Added to the Data Model

New tables have been added to the data model to support internal searches, session attribution, and goals. The new tables are Goal_Dim, Goal_X_Session, Session_Attribution, Origination_Dim, and Int_Search_Term_Dim.

General Enhancements

- There are new capabilities for creating OLAP cubes.
- The Aggr_Int_Search aggregate supports tracking searches for content within a site.
- Page tag logs might contain pages that are not part of the Web site (external). Detailed instructions are included for designating a domain as external.

Documentation Enhancements

- ❑ New detailed instructions for migrating your Web marts from SAS Web Analytics 5.3.x to 5.4 enable you to upgrade your installation.
- ❑ A checklist is provided so you can set up the Clickstream job to accurately capture Clickstream data.

CHAPTER 1

Introduction to SAS Web Analytics

Overview of SAS Web Analytics

How Does SAS Web Analytics Work?

SAS Web Analytics software analyzes and reports on Web site browsing behavior. SAS Web Analytics extracts Web log information from SAS Data Surveyor for Clickstream Data, and can be adapted to work with alternate input sources.

SAS Web Analytics performs the following functions for a Web site:

- reads the detail data sets that are generated by the SAS Data Surveyor for Clickstream Data software. The Data Surveyor software generates the data sets from Web logs or server logs for page tagging data collection.
- loads a warehouse of the Web data.
- (Optional) reads data into the warehouse from other sources.
- aggregates and analyzes the historical data contained in the warehouse.

- provides a Web-based interface for dynamically producing reports about Web site visitors and their behavior.

What Is Considered a Web Site?

Fundamentally, SAS Web Analytics considers a Web site a set of pages that are logically grouped and that share a common domain. SAS Web Analytics analyzes and reports on the data that is contained in the warehouse on a site-by-site basis.

At a higher level, a Web site, which is seen by users in a browser, is a collection of multiple Web servers and multiple subdomains. For example, if the domain is example.com, then you could have these subdomains that a visitor would traverse during a visit:

- **`www.example.com`**
- **`support.example.com`**
- **`products.example.com`**
- **`code.example.com`**

From the visitor's perspective, there is only one site, but actually, multiple sites exist, and each site's content is maintained independently. This content can be hosted on one or more Web servers.

Typically, the Web traffic to a Web site is split according to server architecture and scalability needs, and then further split based on subdomain considerations. Ultimately, the various logs, whether they are from the Web server or the clickstream collection server, are processed by SAS Web Analytics and loaded into a warehouse that stores the data for the Web site's analysis and reporting. SAS Web Analytics refers to this warehouse of data as a site. A site is defined by a set of libraries that point to these collections of data:

SAS Web Analytics data mart

stores the historical detail and visit data for Web site visitors, and stores aggregations related to the historical data. Also stores all the data warehouse tables.

SAS Web Analytics analytical mart

stores the analytical aggregations and the data that is used for reporting analysis.

SAS Web Analytics configuration data

stores internal data for the operation of the product.

SAS Web Analytics staging data

provides intermediate data storage for loading the historical data mart.

SAS Web Analytics Architecture

SAS Web Analytics is built on the SAS 9.2 Enterprise BI Server and SAS 9.2 Enterprise Data Integration Server architecture.

Figure 1.1: SAS Web Analytics 5.4 Architecture

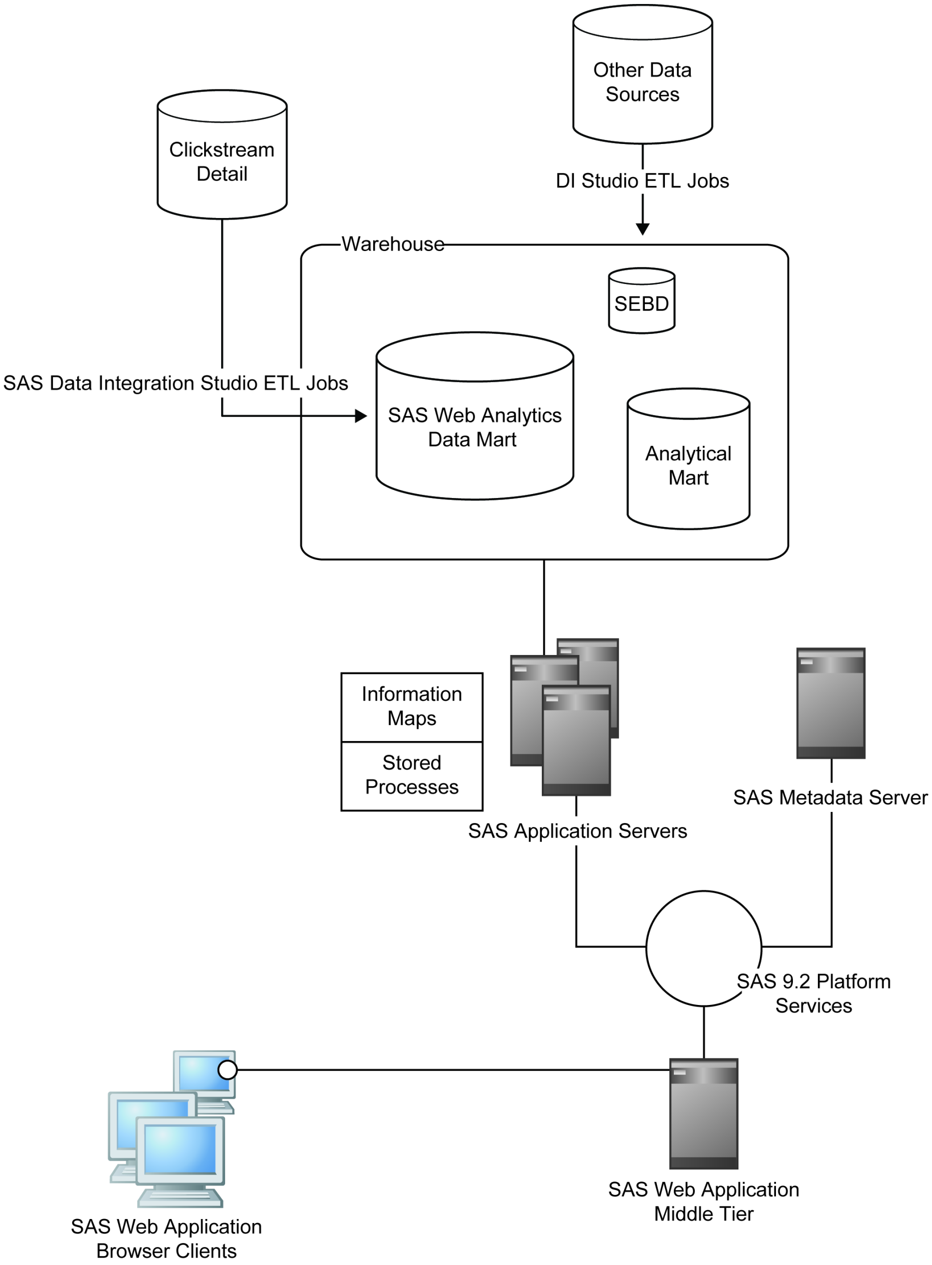
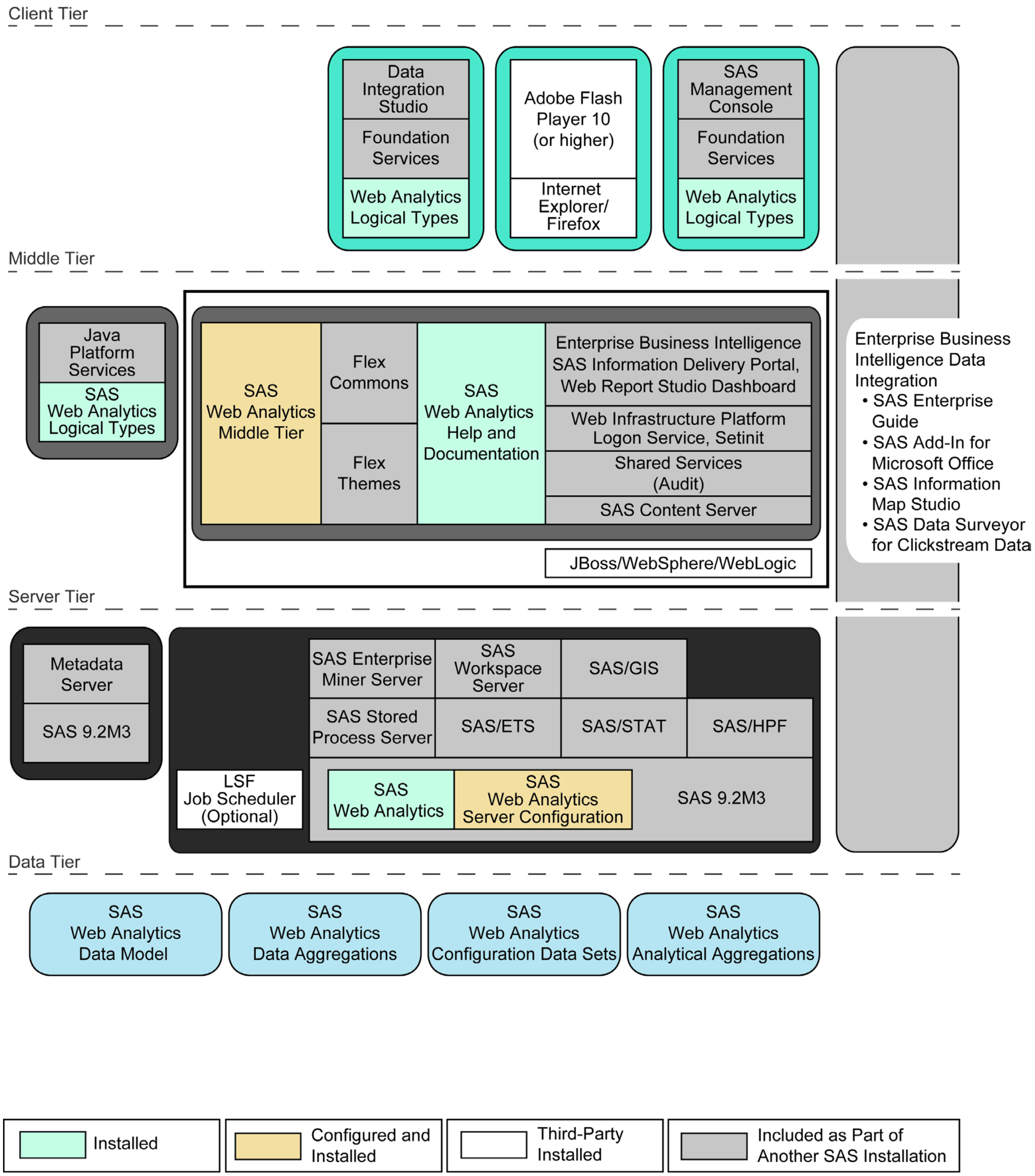


Figure 1.2: SAS Web Analytics 5.4 Topology


Client Tier
Data Integration Studio
Foundation Services
Web Analytics Logical Types
Adobe Flash Player 10 (or higher)
Internet Explorer/ Firefox
SAS Management Console
Foundation Services
Web Analytics Logical Types
Middle Tier
Java Platform Services
SAS Web Analytics Logical Types
SAS Web Analytics Middle Tier
Flex Commons
Flex Themes
SAS Web Analytics Help and Documentation
Enterprise Business Intelligence SAS Information Delivery Portal, Web Report Studio Dashboard
Web Infrastructure Platform Logon Service, Setinit
Shared Services (Audit)
SAS Content Server
JBoss/WebSphere/WebLogic
Enterprise Business Intelligence Data Integration
• SAS Enterprise Guide
• SAS Add-In for Microsoft Office
• SAS Information Map Studio
• SAS Data Surveyor for Clickstream Data
Server Tier
Metadata Server
SAS 9.2M3
SAS Enterprise Miner Server
SAS Workspace Server
SAS/GIS
SAS Stored Process Server
SAS/ETS
SAS/STAT
SAS/HPF
LSF Job Scheduler (Optional)
SAS Web Analytics
SAS Web Analytics Server Configuration
SAS 9.2M3
Data Tier
SAS Web Analytics Data Model
SAS Web Analytics Data Aggregations
SAS Web Analytics Configuration Data Sets
SAS Web Analytics Analytical Aggregations
Installed
Configured and Installed
Third-Party Installed
Included as Part of Another SAS Installation

Accessibility Features of SAS Web Analytics

A complete list of accessibility features is available in the Introduction of the *SAS Web Analytics 5.4 User's Guide*. You can access the guide in the following ways:

- ❑ Go to `http://support.sas.com/documentation/solutions/webanalytics/`
  user ID=**sas**, password=**WAuser123**

- ❑ Select **Contents** from the **Help** menu within the SAS Web Analytics 5.4 software.

Prerequisites for SAS Web Analytics

Before you administer SAS Web Analytics, download and install the latest SAS Middle Tier hot fix, which is available at `http://ftp.sas.com/techsup/download/hotfix/HF2/C80.html`.

Is the Web Site "Analytics-Ready"?

What Is Analytics-Ready?

Analytics-ready means that a Web site has a way to track individual visitors' movements and can identify the page content viewed by each visitor. A Web site needs the following to be analytics-ready:

- ❑ an identifier, such as a visitor ID, assigned to every visitor who enters the Web site
- ❑ an identifier for page content
- ❑ the ability to pass additional information about a visitor or page to the log

Visitor Identification

Visitor Identifiers

A visitor identifier is a field within the Web or tagging log that is used to identify unique visitors to a site. If a visitor identifier is not present, then the Internet Protocol (IP) address and user agent are used. However, the IP address and user agent combination is not unique because the IP address will most likely belong to an Internet Service Provider (ISP) such as AOL. When the IP address belongs to an ISP, it can change with every page a user views within your site.

Typically, the visitor identifier is anonymous and can be a cookie or session ID, depending on the underlying technology of your site. The user logs on to the site with a user ID before accessing the site's content.

For industry-standard definitions for the different types of visitor identifiers, see the World Wide Web Consortium site at `www.w3c.org` or its glossary at `http://www.w3.org/2003/Glossary/`.

Cookies

Cookies are implemented by the Web master, and the site visitor can choose to accept the cookies through Web browser settings. Site visitors can also choose to retain cookies or delete them periodically. If a persistent cookie is enabled by the Web master, then the same cookie is used as the visitor identifier when the visitor returns to the Web site.

JSession IDs

JSession IDs are server-side identifiers set by a JavaServer Pages (JSP) application. Each JSP application has its own server ID. Therefore, programming is necessary to retain the previous application's session ID when the visitor moves to a different JSP application.

Authentication User ID

A Web site is set up so that a user is required to log on to enter the site. Banking and other financial sites that require logging on could use the authentication user ID as the visitor ID for a visit. The user logon can be set up by the Web master for an entire site or by a Web site developer for specific sections of a site.

Page Identification

Many Web sites contain different URLs that resolve to the same viewable page (for example, both **`http://`*`yourdomain`*`/index.html`** and **`http://`*`yourdomain`*`/`** return the same page). As long as the URLs are different, the analytics processes them as different pages.

If you want your analytics to process these two pages as the same page, then modify your data so that it translates all the equivalent URLs to a single URL. Make this modification at any of several places in the ETL process:

- ❑ Before SAS Data Surveyor for Clickstream processing – by modifying the source log directly with a custom program that you create
- ❑ During SAS Data Surveyor for Clickstream processing – by implementing transformation rules that use Clickstream Data Surveyor
- ❑ After SAS Data Surveyor for Clickstream processing – by modifying the clickstream output SAS data set with a custom program that you create

CAUTION:

After you have begun loading data into your warehouse, be very careful about changing the data cleansing procedures you already have in place. Such changes can cause a break in the continuity of your warehouse data. To avoid breaks in data continuity, it is recommended that you implement the change, and then completely rebuild the warehouse.

Passing Information Using Query String Parameters

What Is a Query String?

The query string is part of the Uniform Resource Identifier (URI) that occurs after the question mark (?). The query string consists of name value pairs that are separated by an ampersand (&) and can be used to do the following tasks:

- ❑ to pass information to the application that will create the next page viewed.

- to pass information to the log. This means that additional information about a visitor can be made available for analysis within a log file.

Here is an example of a query string:

```
http://yourdomain/index.html?parm1=1&parm2=2
```

Here are examples of data that can be passed through the query string:

- form fields.
- marketing campaign information. For an example of using a query string to set **`seb_campaignid`**, see Chapter 5, "Working with Search Engine Bid Data."
- a JSession ID from the Web application.

Constructing the Query String

Query strings are collections of name value pairs that occur on the URI after the question mark (?) and are separated by an ampersand (&). Here is an example of a URI with a query string:

```
www.mydomain.com/sales_page.jsp?product+xyz&town=Apex
```

Query strings are created by one of the following methods:

- HTML Forms – If the Web page has a form with a button that sends the visitor to the next page, then the items within the form are used to create a query string.

CAUTION:

HTML forms have a Method option. When the option is set to POST, it prevents the query string from being written within the log.

- Programmatic – A JSP, PHP, or other application that creates the Web page content adds a query string to the URL.
- Manual – The link on a Web page or e-mail includes a query string.

CAUTION:

Personally identifiable information (PII) should not be passed through query strings.

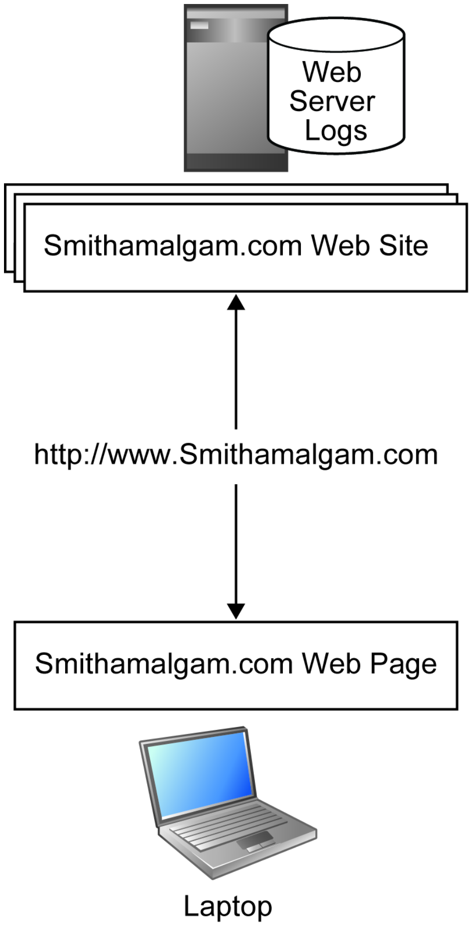
Example: Connect Back-End Data to a Site Visit by Using a Query String

During a visit, you can connect a customer's behavior on your site with other information by adding a name value pair that contains an identifier to a query string. For example, the visitor needs to identify himself when purchasing merchandise. An order ID can be placed in a query string name value pair when the order summary page is displayed for the visitor. After the order number is available within the log, it can be used to link that visit with additional customer and order information either during the Web site ETL or in a post-ETL process.

Introduction to Web Data

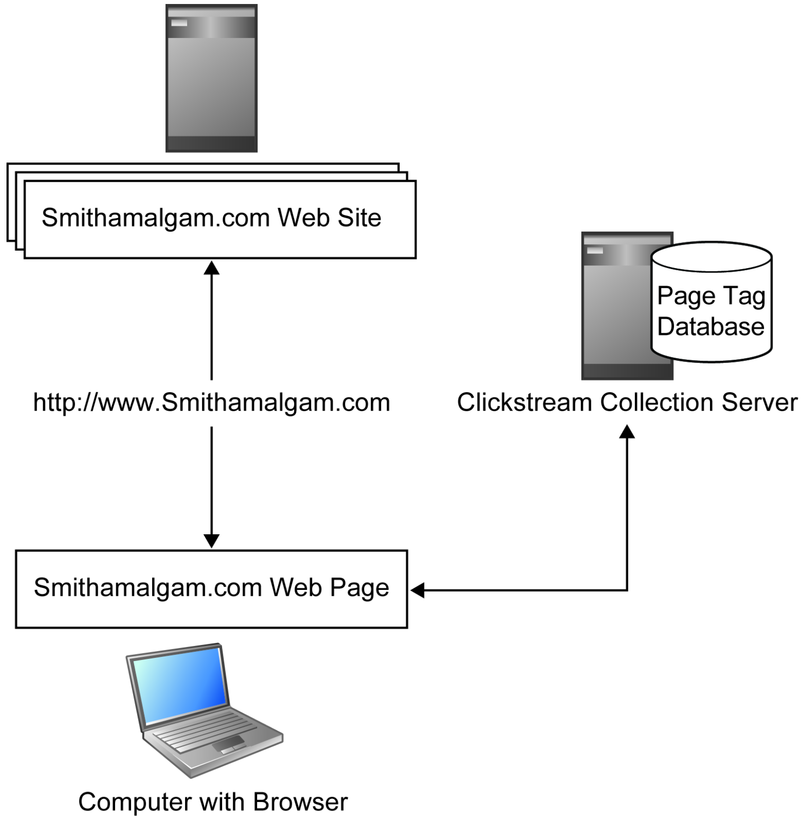
Web data consists of Web site page requests by a Web browser to a Web server. An individual page request can trigger requests within the Web server for the GIF, JPEG, JavaScript, and other files associated with the page request. All of the file requests are recorded in a Web server log, which can be written to a log file at specific intervals by the Web server.

Figure 1.3: Web Logs Capturing Data from a User Browsing a Web Site


Web Server Logs
Smithamalgam.com Web Site
http://www.Smithamalgam.com
Smithamalgam.com Web Page
Laptop

An alternative to writing log files as a method of Web data collection is page tagging. Tags are units of JavaScript code that direct information to a tagging server.

Figure 1.4: SAS Clickstream Server Capturing Web Data from a User Browsing a Web Site


Smithamalgam.com Web Site
Page Tag Database
http://www.Smithamalgam.com
Clickstream Collection Server
Smithamalgam.com Web Page
Computer with Browser

Introduction to SAS Unicode

Why Should You Use SAS Unicode?

SAS Unicode provides an environment for multiple language support, which includes support for the following tasks:

- ❑ handling different character encodings
- ❑ providing messages according to the locale and language of the user
- ❑ enabling a multi-lingual user interface

The SAS Web Analytics Web application provides a multi-lingual user interface that with SAS Unicode server support operates consistently with the locale and language settings of the user's browser.

SAS Unicode server is a particularly good option for Web data because most Web sites have a global presence and could receive request parameters (such as search terms) in multiple languages. Analysis is straightforward for this data if the strings from URL encodings are decoded using the URLDECODE transformation option in SAS Data Surveyor for Clickstream Data. (DECODE is the default transformation option.)

For more information about the URLDECODE option, see the Parse transformation Input Options in the SAS Data Surveyor for Clickstream Data Help.

SAS Unicode server handles storage of these strings from potentially different languages, regardless of their native encoding.

Note: When you use the Clickstream Parse transformation to decode a URI, use the URIDECODE parameter. However, the query string, referrer parameter, and requested file are not decoded. For more information, see the online Help for SAS Data Surveyor for Clickstream 2.2 and the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

Multi-byte Language Support for Reports

SAS Web Analytics supports languages with multi-byte character set (MBCS) encodings. SAS Unicode server provides uniform and multi-language support. Alternatively, languages with MBCS encodings are supported natively by using SAS in MBCS. However, without SAS Unicode server support, the SAS Web Analytics data server only supports data that is consistent with the selected MBCS encoding.

Data Sharing between SAS Web Analytics and SAS Customer Intelligence

SAS Web Analytics and SAS Customer Intelligence (the SAS Marketing Automation product) might have different types of SAS encodings defined. When SAS Web Analytics and SAS Customer Intelligence are installed together, SAS Marketing Automation reads the response history data from SAS Web Analytics. Data integration across potentially different SAS encodings is possible because SAS uses its Cross-Environment Data Access (CEDA) feature to handle data translations. For more information about CEDA, see *SAS 9.2 Language Reference: Concepts*.

Note: SAS Marketing Automation might return a message to SAS Web Analytics using encoding that is different from the encoding of the SAS Web Analytics SAS server, which renders the message unreadable. If this happens, adjust the SAS Marketing Automation installation to use UTF-8 encoding. For more information, see the *SAS Marketing Automation: Administrator's Guide.*

Setting Up the SAS Unicode Server Environment

To use SAS Unicode server successfully, you must ensure that the SAS servers and all sessions that they invoke are SAS Unicode server sessions.

There are multiple ways of setting up the environment to use SAS Unicode server. One method is discussed here.

Note: For the method that is best for your environment, see `http://support.sas.com/resources/papers/92unicodesrvr.pdf`. For information about setting up your SAS BI Server environment to use SAS Unicode server, see Appendix B.

SAS Unicode Server support can be set up within Windows environments by editing the configuration file `!sasroot/SASFoundation/9.2/sasv9.cfg` (where `!sasroot` is the installation location of SAS). Ensure that the file includes the SAS configuration file that is in the stand-alone **SAS 9.2 (Unicode Server)**. From the Windows Start menu, select **Programs** ▸ **SAS** ▸ **SAS 9.2 License Renewal & Utilities**.

For example, in Windows, here are the contents of the file **!sasroot/SASFoundation/9.2/sasv9.cfg**:

```
 -CONFIG "C:\Program Files\SAS\SASFoundation\9.2\nls\u8\SASV9.CFG"
```

SAS Unicode server support can be set up within the UNIX environment by editing the configuration file **!sashome/SASFoundation/9.2/sasv9_local.cfg**, where **!sashome** is the installation location of SAS, to include the following contents:

```
-config !sasroot/nls/u8/sasv9.cfg
```

Because all SAS sessions read this configuration file, all SAS sessions, including those on SAS servers, will be SAS Unicode server sessions after you edit and save the file. You must restart the SAS sessions for this change to take effect.

For more information about switching languages of reports dynamically, see **http://support.sas.com/resources/papers/LocaleSwitching.pdf**.

Introduction to Search Engine Bid Data

Search engine bid data (SEBD) is provided by Google Inc. as part of its Google AdWords program. SEBD consists of paid keyword activity and keyword account information. This information is made available by Google through the Google AdWords application programming interface (API).

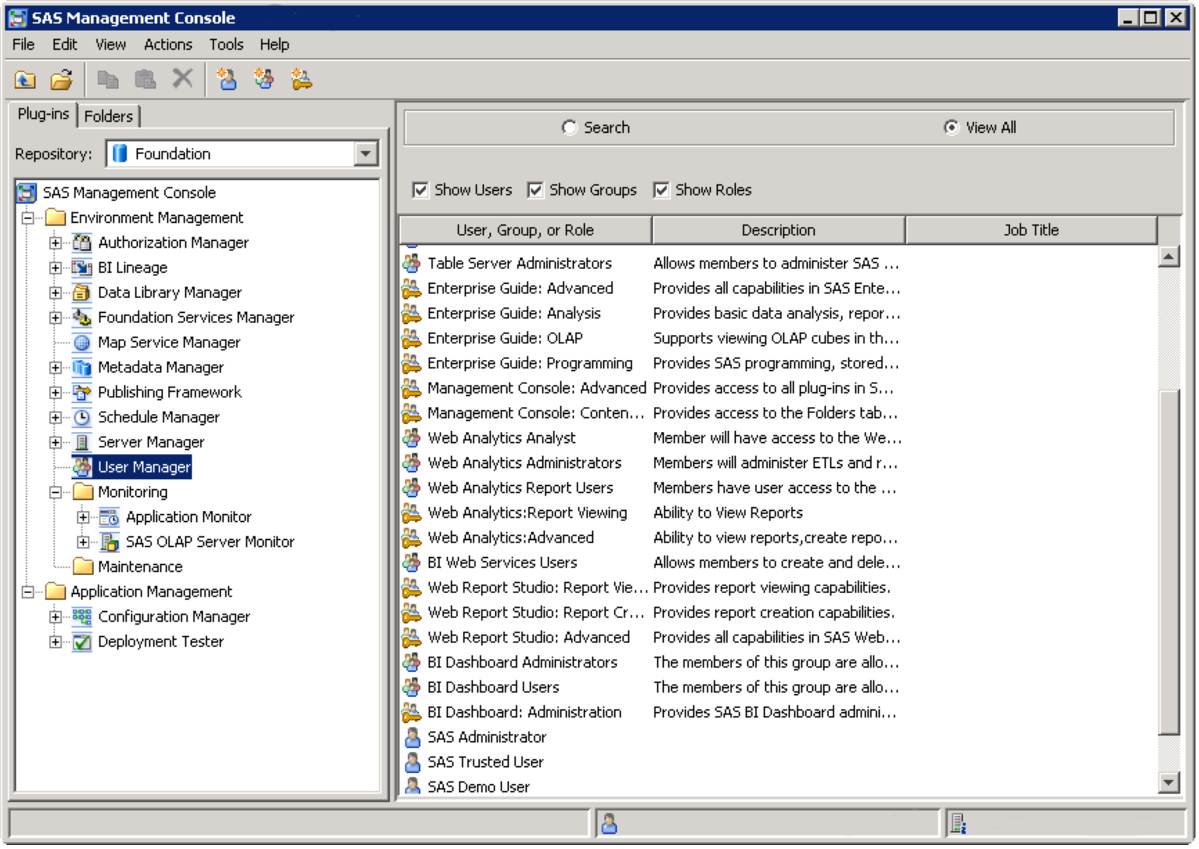
For information about Google AdWords, keywords, or keyword auctions, go to **http://www.google.com/corporate/** and click **Google AdWords**. For information about the Google AdWords API, go to **http://code.google.com/apis/adwords/docs/developer/index.html**.

SAS Web Analytics Security Setup

Users and Groups

Users need to be assigned to the SAS Web Analytics user groups that are provided in SAS Management Console User Manager. User groups for SAS Web Analytics are shown in the User Manager along with users and roles.

Figure 1.5: SAS Management Console



Repository: Foundation

○ Search ● View All

☑ Show Users ☑ Show Groups ☑ Show Roles

User, Group, or Role	Description	Job Title
Table Server Administrators	Allows members to administer SAS ...	
Enterprise Guide: Advanced	Provides all capabilities in SAS Ente...	
Enterprise Guide: Analysis	Provides basic data analysis, repor...	
Enterprise Guide: OLAP	Supports viewing OLAP cubes in th...	
Enterprise Guide: Programming	Provides SAS programming, stored...	
Management Console: Advanced	Provides access to all plug-ins in S...	
Management Console: Conten...	Provides access to the Folders tab...	
Web Analytics Analyst	Member will have access to the We...	
Web Analytics Administrators	Members will administer ETLs and r...	
Web Analytics Report Users	Members have user access to the ...	
Web Analytics:Report Viewing	Ability to View Reports	
Web Analytics:Advanced	Ability to view reports,create repo...	
BI Web Services Users	Allows members to create and dele...	
Web Report Studio: Report Vie...	Provides report viewing capabilities.	
Web Report Studio: Report Cr...	Provides report creation capabilities.	
Web Report Studio: Advanced	Provides all capabilities in SAS Web...	
BI Dashboard Administrators	The members of this group are allo...	
BI Dashboard Users	The members of this group are allo...	
BI Dashboard: Administration	Provides SAS BI Dashboard admini...	
SAS Administrator		
SAS Trusted User		
SAS Demo User		

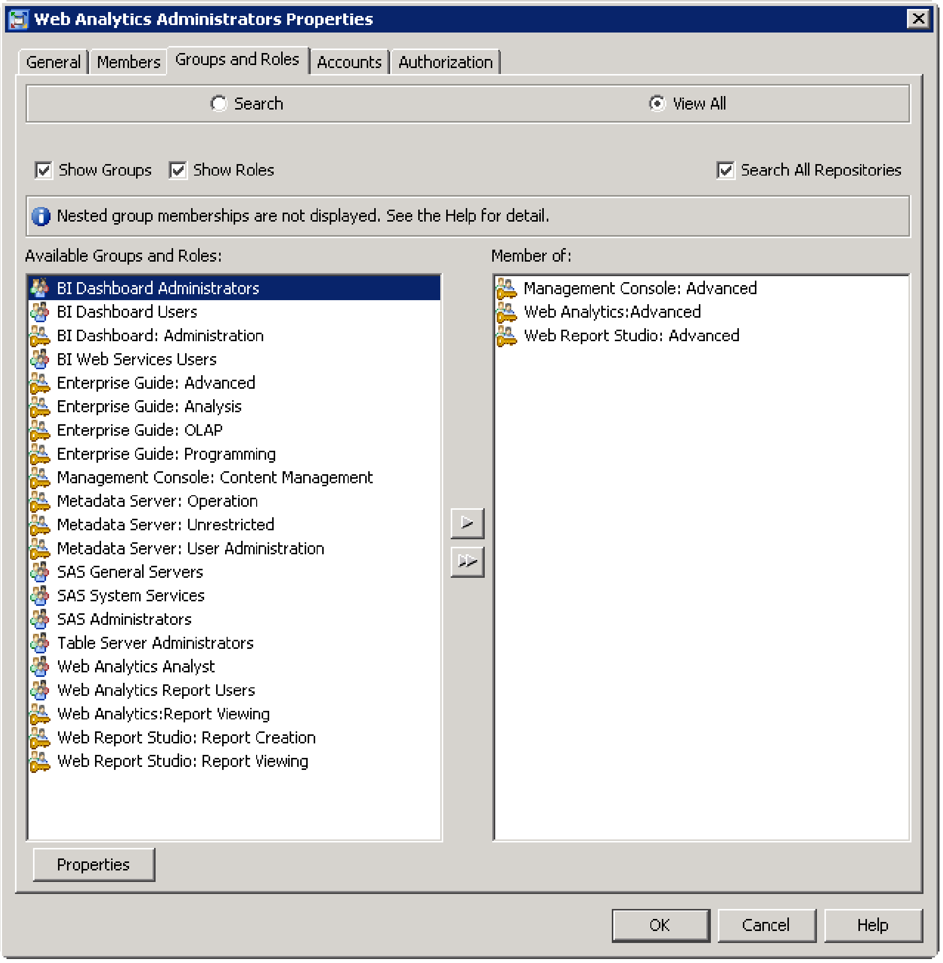
The following table lists the user groups provided for SAS Web Analytics and describes their uses.

Table 1.1: Web Analytics User Groups

Group Name	Description
Web Analytics Administrators	Use this group to administer the ETL processes and reports associated with SAS Web Analytics. The group is initialized automatically with the **Web Analytics: Advanced** role. In addition, the administrative user has the following access: Weblog ETL can create a Web mart site, customize ETL jobs, and schedule jobs. SEBD Google AdWords ETL can access ETL jobs. It is recommended that these roles be added to this group: **Enterprise Guide: Advanced** **Management Console: Advanced** **Web Report Studio: Advanced**
Web Analytics Analyst	Use this group to grant users access to the SAS Web Analytics application and SAS Web Analytics Analyst portal page. The group is initialized automatically with the **Web Analytics: Advanced** role. It is recommended that these roles be added to this group: **Enterprise Guide: Advanced** **Web Report Studio: Advanced**
Web Analytics Report Users	Use this group to grant users access to the SAS Web Analytics Reports portal page.

You can assign users to groups based on the following criteria:

- ❑ If some users are ETL (extract, transform, and load) developers or administrators, then assign them to the **Web Analytics Administrators** group.



○ Search ● View All

☑ Show Groups ☑ Show Roles ☑ Search All Repositories

Nested group memberships are not displayed. See the Help for detail.

Available Groups and Roles:

- BI Dashboard Administrators
- BI Dashboard Users
- BI Dashboard: Administration
- BI Web Services Users
- Enterprise Guide: Advanced
- Enterprise Guide: Analysis
- Enterprise Guide: OLAP
- Enterprise Guide: Programming
- Management Console: Content Management
- Metadata Server: Operation
- Metadata Server: Unrestricted
- Metadata Server: User Administration
- SAS General Servers
- SAS System Services
- SAS Administrators
- Table Server Administrators
- Web Analytics Analyst
- Web Analytics Report Users
- Web Analytics:Report Viewing
- Web Report Studio: Report Creation
- Web Report Studio: Report Viewing

Member of:

- Management Console: Advanced
- Web Analytics:Advanced
- Web Report Studio: Advanced

- If some users are analysts, then assign them to the **Web Analytics Analyst** group.
- If some users need access only to reports that were created within the SAS Web Analytics application, then assign those users to the **Web Analytics Report Users** group.

For more information about the User Manager, see the online Help in SAS Management Console and the *SAS 9.2 Intelligence Platform: Security Administration Guide*. For information about security setup and best practices, see “Security Overview” in *SAS 9.2 Intelligence Platform: Overview*.

Roles

The following roles are provided in SAS Management Console for SAS Web Analytics.

- **Web Analytics: Advanced** – gives a user report creation, report viewing, and administrative capabilities within the SAS Web Analytics application.
- **Web Analytics: Report Viewing** – gives a user report viewing capabilities within the SAS Web Analytics application.

Folder Authorizations

The following table lists recommended authorizations for the SAS Web Analytics folders. The authorizations are automatically set up for the Web Analytics groups during the Web Analytics configuration.

Table 1.2: Recommended Folder Authorizations for SAS Web Analytics Groups

Folder	Group	Authorization
/Products/SAS Web Analytics	Web Analytics Administrators	MetadataRead, MetadataWrite, and administer access to all folders within this folder structure.
/Products/SAS Web Analytics	Web Analytics Analyst	MetadataRead, MetadataWrite, and administer access to all folders within this folder structure.
/Products/SAS Web Analytics	Web Analytics Report Users	MetadataRead-only access.
/Products/SAS Web Analytics/Reports	Web Analytics Analyst	MetadataRead and MetadataWrite access. Read, Write, and administer access to all folders within the folder structure.
/Products/SAS Web Analytics/Reports	Web Analytics Report Users	MetadataRead-only access.
/Products/SAS Web Analytics/SEBD/Data Sources/SEBDMART	Web Analytics Analyst	MetadataRead-only access.
/Products/SAS Web Analytics/SEBD/Data Sources/SEBDMART	Web Analytics Report Users	MetadataRead-only access.
/System/Applications/SAS Web Analytics	Web Analytics Administrators	MetadataRead, MetadataWrite, and administer access to all folders within this folder structure.
/System/Applications/SAS Web Analytics/Web Analytics 5.4/Sites	Web Analytics Analyst	MetadataRead and MetadataWrite access.
/System/Applications/SAS Web Analytics/Web Analytics 5.4/Sites	Web Analytics Report Users	MetadataRead-only access.
/Shared Data/Web Analytics	Web Analytics ETL Administrators	MetadataRead, MetadataWrite, and administer access to all folders within this folder structure. Read access to cubes and tables within this folder structure.
/Shared Data/Web Analytics	Web Analytics Analyst	MetadataRead and MetadataWrite access. Read access to cubes and tables within this folder structure.
/Shared Data/Web Analytics	Web Analytics Report Users	MetadataRead access. Read access to cubes and tables within this folder structure.

CHAPTER 2 Setting Up a Web Mart

Checklist for Setting Up a Web Mart

1. Make any necessary changes to the Web site to ensure that analytic data is collected. For more information, see Chapter 1, “Introduction to SAS Web Analytics.”
2. Arrange for the creation of logs on a regular basis. Typically, logs are created by the Web master so that they contain 24 hours of data. The regularity of the log creation depends on the needs of the site.

   Depending on the structure of the Web servers that host your site, one or more files are created. Coordinate with your Web master to determine how many files are created each time a log is created.
3. Determine where the Web site data repository will be located. For more information, see “Pre-Initialization Setup” on page 19.
4. Initialize the site. For more information, see “Site Initialization” on page 20.
5. Copy the appropriate Clickstream job to the log files for processing. For more information, see “Post-Site Initialization ” on page 27.
6. Set up the Clickstream job. For more information, see “Setting Up the Clickstream Job” on page 31.

Log Creation

General Steps for Collecting Data

The first step for analyzing Web data is to collect the data. If you are using Web logs or application logs, then the Web master needs to set up the format of the logs and schedule file creation. The following fields are required:

- ❑ IP address
- ❑ date and time of the request
- ❑ requested file – this should include the request method, URL, and query string (URI)
- ❑ status code
- ❑ bytes sent
- ❑ user agent
- ❑ referrer

Additionally, these optional fields could be available depending on the Web site setup and the Web server used:

- ❑ cookies
- ❑ domain
- ❑ authenticated login user ID
- ❑ custom fields

Web and Application Server Logs

It is recommended that you use a Web server that is set up to create logs with Extended Log Format (ELF). At a minimum, ELF logs contain these fields:

- ❑ IP address
- ❑ date and time of the request – this can be in GMT format. Check with your Web master about setup.
- ❑ requested file – includes method and complete URI
- ❑ status code
- ❑ bytes sent
- ❑ user agent
- ❑ referrer

Note: Depending on the Web server, additional fields might be available to include within the ELF log.

CAUTION:

If the Common Log Format is used, then the following required information might not be present, depending on the Web Server that is hosting your site:

- ❑ status code
- ❑ user agent
- ❑ referrer

Clickstream Collection Server Logs

The preferred data collection server for Web Analytics is the clickstream collection server. For information about page tagging, collection server setup, and log files, see the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide.*

Initialize the Site

Introduction to Site Initialization

The Site Initialization job creates the following items:

- ❑ all of the metadata associated with an individual site, including the jobs and a Web mart object
- ❑ all of the tables associated with a Web mart, including the data mart tables and configuration tables

Note: If the data mart will be stored in Oracle, there is an option to create or not create the tables. The tables can be created later by the database administrator.

Pre-Initialization Setup

Complete the following steps before you set up a Web mart:

1. Set up SAS Management Console to meet your organization's needs. A best practice is to create an ETL administrator user ID. The ETL administrator user ID should be able to log on to the machine where the ETL will be run and belong to the following groups:

   Management Console: Advanced

   Web Analytics Administrator – this group is used to define authorizations for the `/System/Applications/SAS Web Analytics` and `/Products/SAS Web Analytics` folders.

   Web Report Studio: Advanced

   For more information about setting up SAS Web Analytics user groups, see "SAS Web Analytics Security Setup" in Chapter 1.

2. Update the Java value in the sasv9.cfg file for the JREOPTIONS SAS option from **-Xmx128m** to **-Xmx256m**. For more information, see Appendix 4.

3. (Optional) Set up Oracle. Set up a library within SAS Management Console that links to the Oracle database schema where the data will be stored. See the Oracle database administrator for the correct procedure for creating the tables. For information about creating the data definition language (DDL), see Chapter 6, "Working with Oracle Databases." For information about Oracle databases and the ETL aggregate jobs, see Appendix 5.

   Note: It is recommended that you follow best practices for setting up an Oracle server. Best practices include creating the DBUSER Group, which can be set up to contain logon information for Oracle databases.

   For more information about assigning a library for an Oracle database in SAS Management Console, see the *SAS 9.2 Management Console Guide to Users and Permissions*.

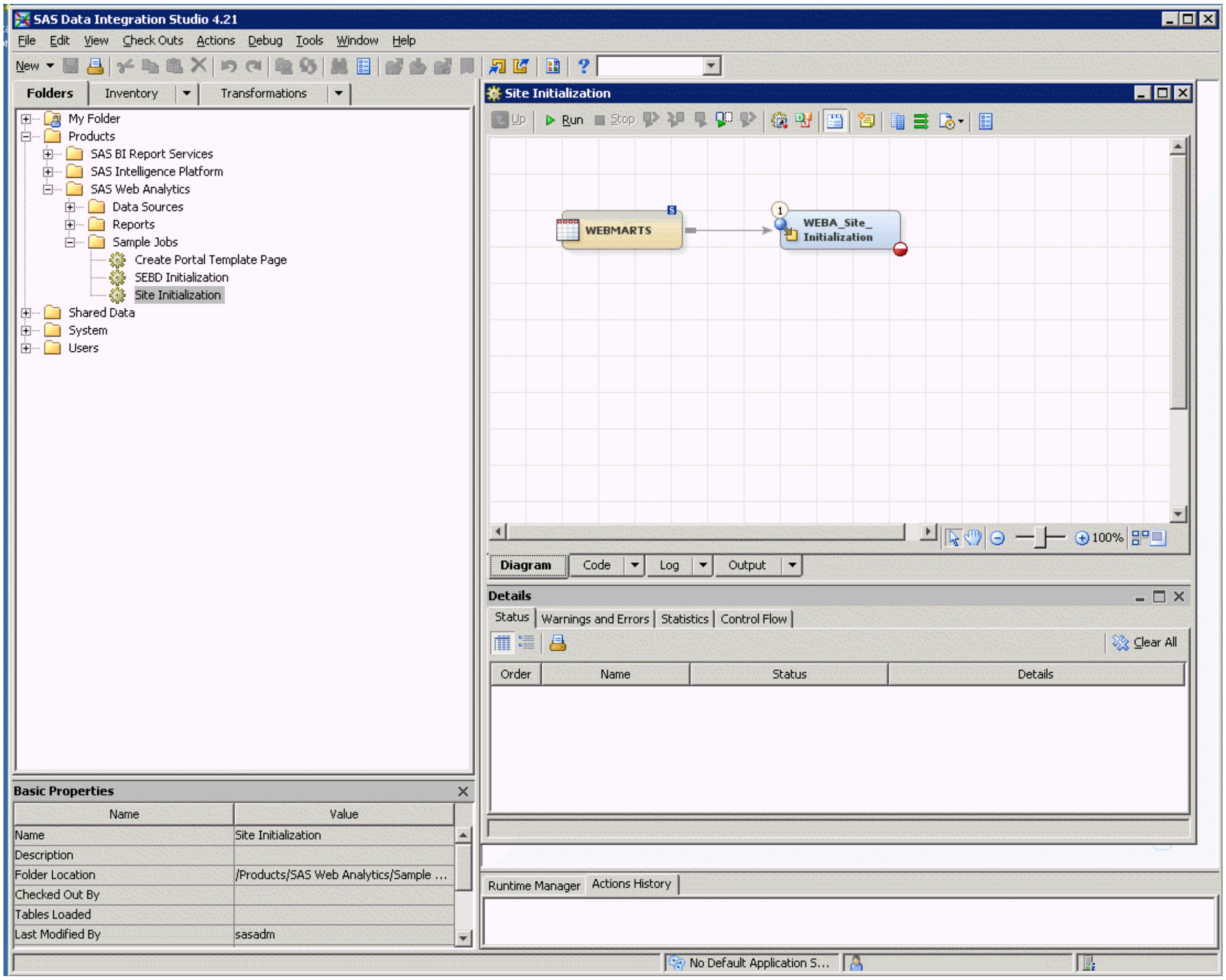
4 If you plan to run the site initialization on a UNIX server, then you must set up for using the UNIX environment. See the Windows and UNIX instructions for running a SAS Web Analytics site initialization program beginning on page 25.

Site Initialization

1 Open SAS Data Integration Studio and navigate to **`/Products/SAS Web Analytics/Sample Jobs`**.

2 Copy the Site Initialization job and use Paste Special to paste the job into another folder. For example, use **`/My Folder`** or **`/Shared Data/My Folder`**.

3 Open the Site Initialization job by double-clicking the job.

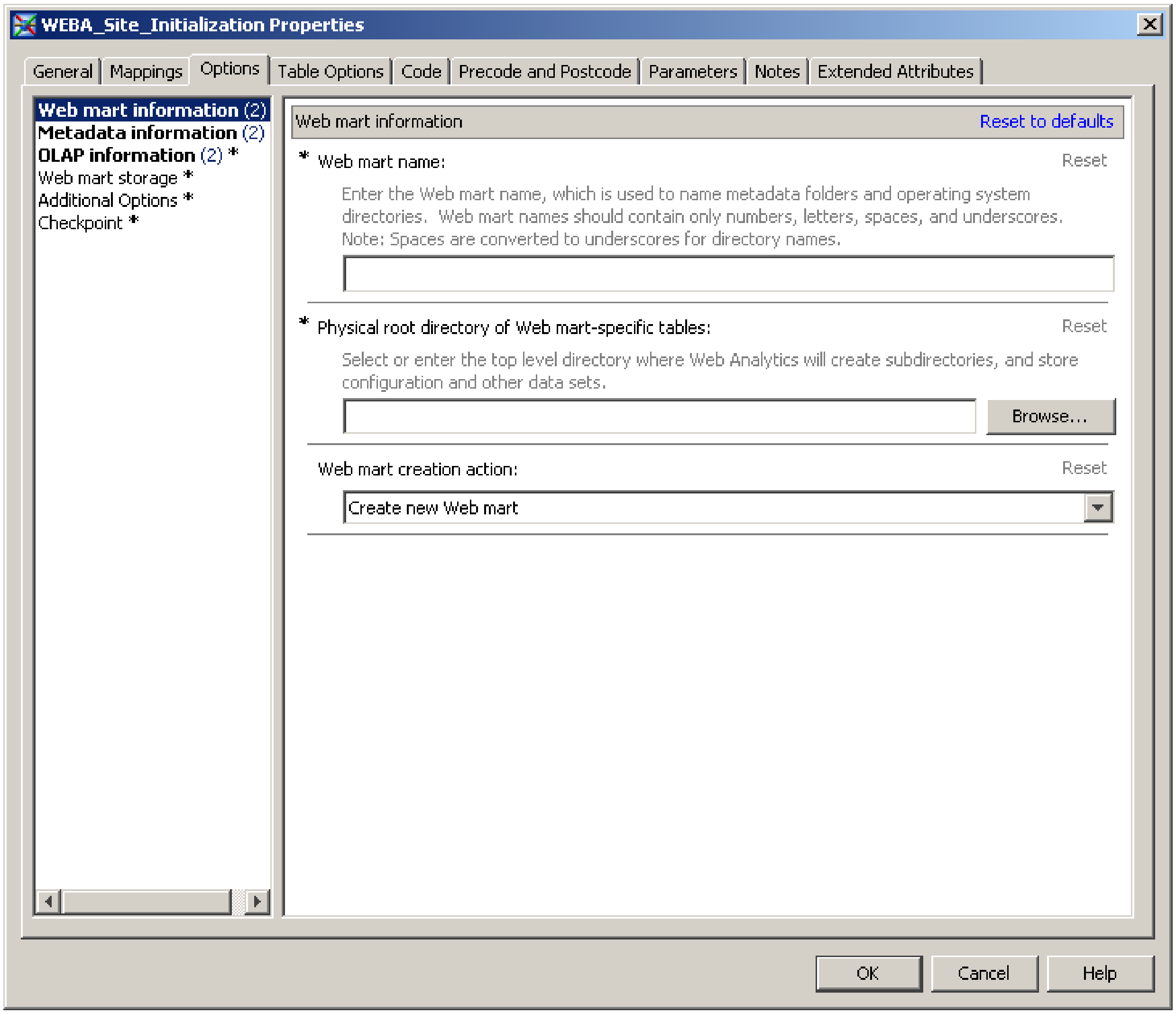


Basic Properties

Name	Value
Name	Site Initialization
Description	
Folder Location	/Products/SAS Web Analytics/Sample ...
Checked Out By	
Tables Loaded	
Last Modified By	sasadm

4 Double-click the **WEBA_Site_Initialization** transformation within the selected job, and then click the **Options** tab in the Properties dialog box. Complete the required fields. In the **Web mart creation action** field you can select a new Web mart.

Here is a sample **Options** tab for initializing a Web mart that contains SAS data sets:



WEBA_Site_Initialization Properties

Web mart information (2)
Metadata information (2)
OLAP information (2) *
Web mart storage *
Additional Options *
Checkpoint *

Web mart information

* Web mart name:

Enter the Web mart name, which is used to name metadata folders and operating system directories. Web mart names should contain only numbers, letters, spaces, and underscores. Note: Spaces are converted to underscores for directory names.

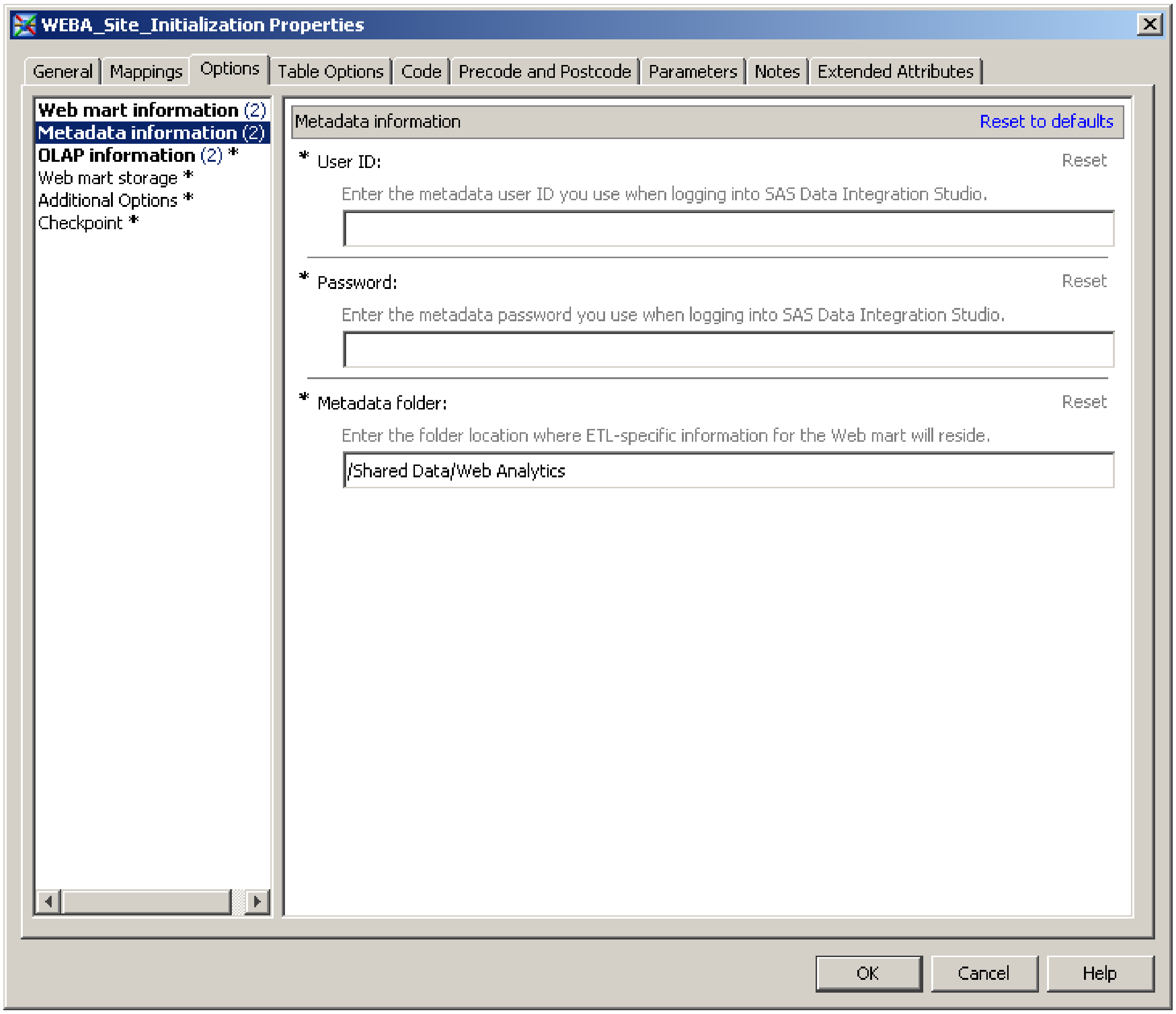
* Physical root directory of Web mart-specific tables:

Select or enter the top level directory where Web Analytics will create subdirectories, and store configuration and other data sets.

Web mart creation action:

Create new Web mart

5 Enter metadata information for the Web mart.



Metadata information

* User ID:

Enter the metadata user ID you use when logging into SAS Data Integration Studio.

* Password:

Enter the metadata password you use when logging into SAS Data Integration Studio.

* Metadata folder:

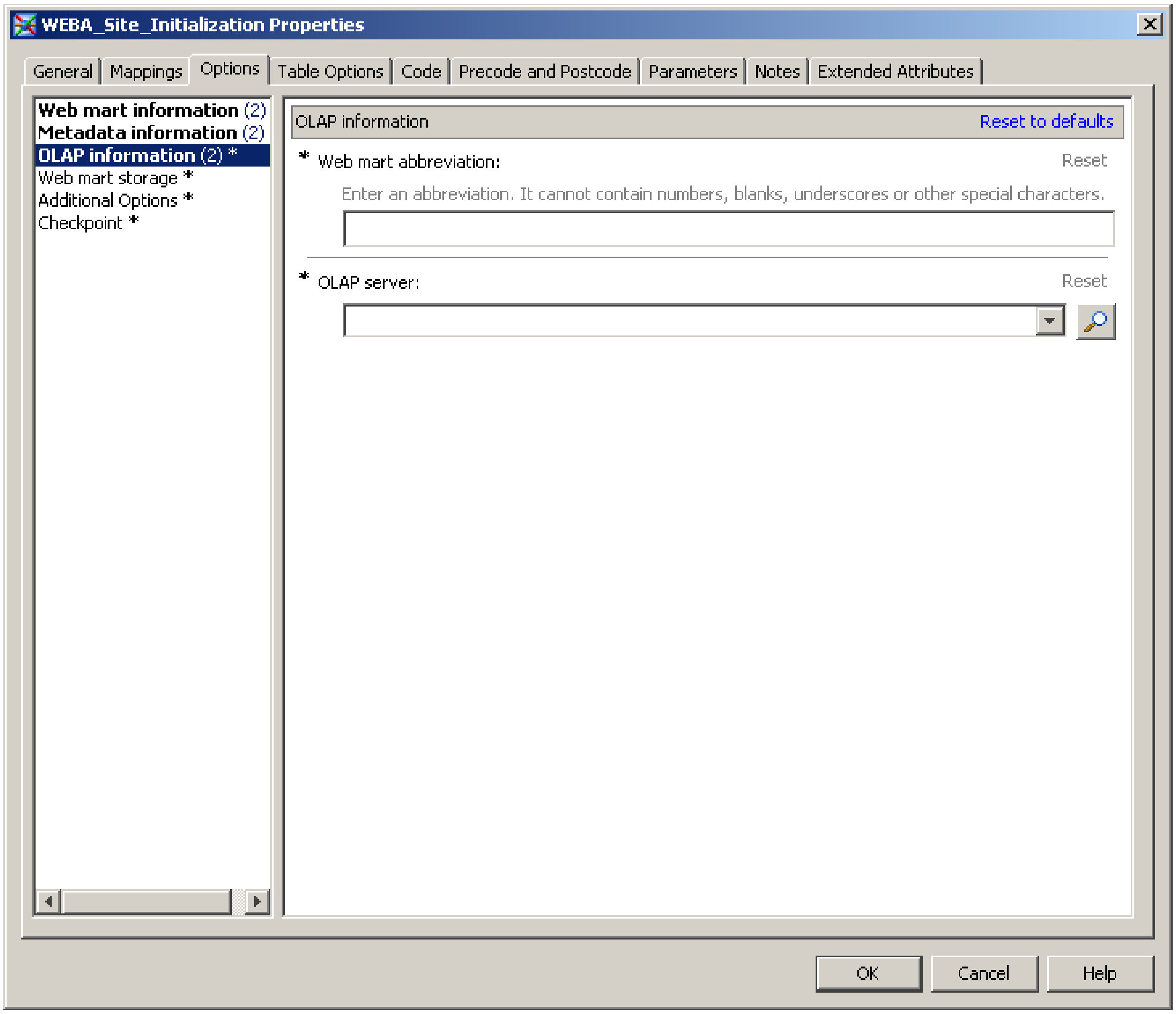
Enter the folder location where ETL-specific information for the Web mart will reside.

/Shared Data/Web Analytics

Note: The User ID should be a member of the Web Analytics Administrator group or have Read and Write access to the metadata folder **`/System/Applications/SAS Web Analytics/Web Analytics 5.4`** and the folder name that was entered in the **Metadata root directory** field.

Click **OK**.

6 Enter OLAP information for the Web mart.



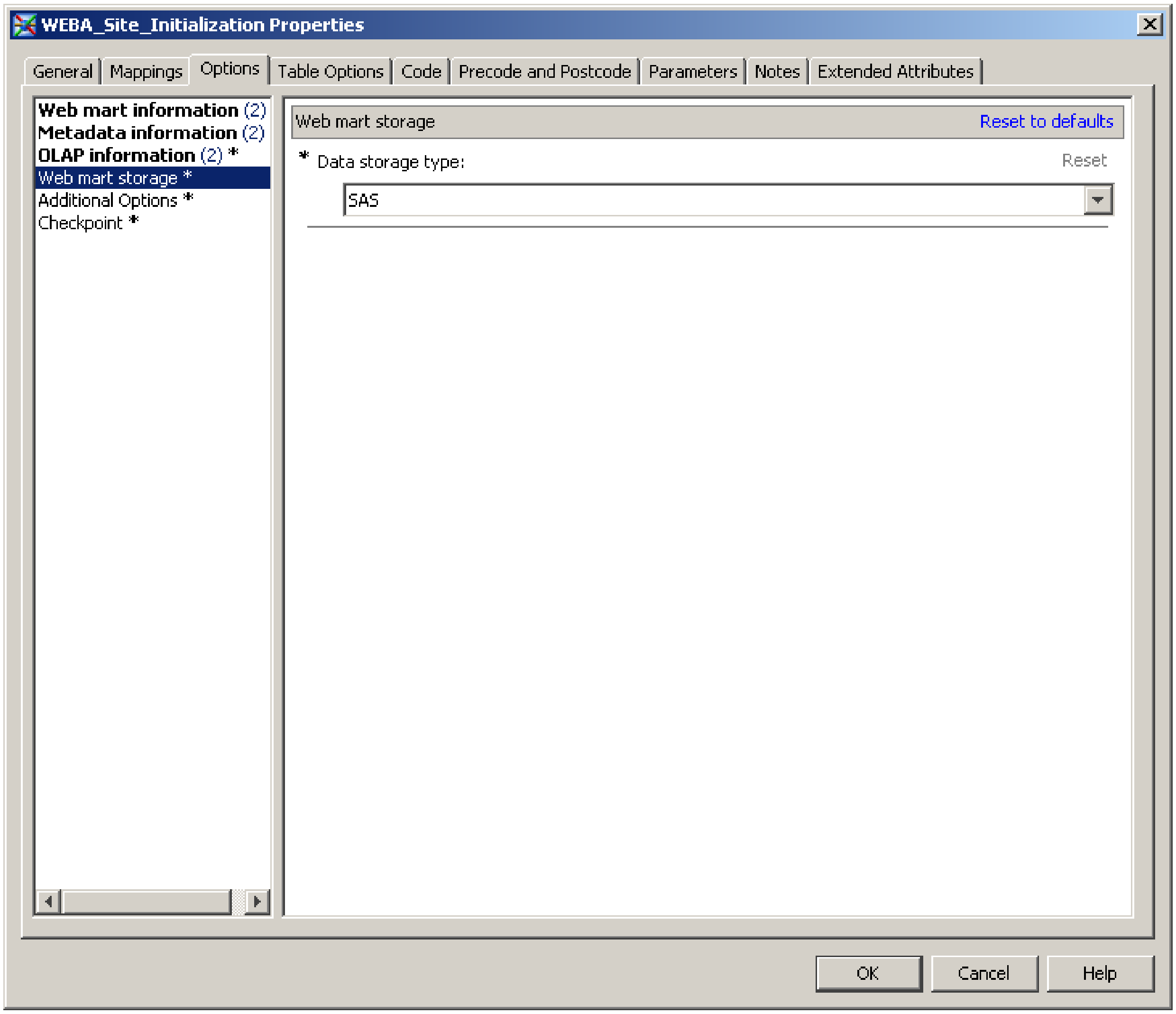
WEBA_Site_Initialization Properties

* Web mart abbreviation: Reset

Enter an abbreviation. It cannot contain numbers, blanks, underscores or other special characters.

* OLAP server: Reset

7 Enter data storage information for the Web mart.



WEBA_Site_Initialization Properties

* Data storage type: SAS

Note: When selecting a data mart table type, you need to determine where the Web mart data will be stored, in SAS or Oracle. (This is a decision that should have been made before starting the site initialization process.)

8 Run the job.

Windows

Click the Run icon in the SAS Data Integration Studio Job pane. If the Site Initialization job runs successfully, then a new folder structure is created under the folder that you specified for the metadata root directory on the **Options** tab. If you are using the default value for the metadata root directory, the new folder structure is created under **`Shared Data/Web Analytics`**.

UNIX

You can run the SAS Web Analytics site initialization program only within an environment where windowing services are available. Typically, a valid display environment means the following:

An X11 server is running (for example, on a UNIX server or a PC (Exceed(tm));

The user environment that is invoking the SAS session must have X11 permissions to connect and use the X11 resources. If needed, permissions are granted through the `xhost` command.

The DISPLAY environment variable is set (and exported) to a valid display on the X11 server (for example, localhost:0.0) .

This display environment is similar to the one users would need to run for an interactive SAS session, xterm, and so on.

In order to run the Site Initialization job from SAS Data Integration Studio, the workspace server has to be set up. This is so that when the SAS session is invoked within the workspace server context, the display environment is set up as described. Set up the workspace server by editing the WorkspaceServer_usermods.sh file and possibly the shell profile of the user that runs the workspace server (for example, sassrv).

Run the Job in UNIX Using an Interactive SAS Session

If you are unable to modify your workspace server's environment as needed, schedule the Site Initialization job, and then deploy and run the job outside of SAS Data Integration Studio. SAS Web Analytics macros need to be available and the display environment needs to be set up. For information about scheduling a job, see the SAS Data Integration Studio online Help.

There are three ways to invoke the site initialization once it is deployed:

Run the program using the `sasbatch` command found in <*your-server-installation-location*>/Config/Lev1/SASApp/BatchServer (for example, <*your-server-installation-location*>**/Config/Lev1/SASApp/BatchServer/sasbatch.sh -sysin** *path-to-program***/Site_Initialization.sas -log** *path-to-program***/Site_Initialization.log**)

Run the program using the SAS command found in the installation directory and specify the SASApp application server configuration file (for example, <*your-installation-dir*>/ **sas -config** <*your-server-installation-location*>/**Config/Lev1/SASApp/sasv9.cfg** *path-to-program* / **Site_Initialization.sas -log** *path-to-program/* **Site_Initialization.log**)

Invoke an interactive SAS session.

Open and submit the Site_Initialization.sas program.

Note: If you change the parameters of the Site Initialization job after the job has run successfully, save and close the job before you rerun it. To avoid memory errors, reopen the job and run it with the new parameters.

Post-Site Initialization Checklist

Setup Checks

❑ After the Site Initialization job has been run, the following components have been created within the metadata:

A folder that is labeled with the Web mart name under the metadata root directory. This folder contains the following folders:

5.4 Jobs	all ETL jobs reside in this folder

Folders that contain the library and table objects associated with the jobs:

EvntSrc
WaAnly
WaCnfg
WaMart
WaSrc
WaWork

Verify that the correct directories were created. The following directories were created under the operating system root directory:

EvntSrc
WaAnly
WaCnfg
WaMart
WaSrc
WaWork

Note: If an Oracle database is specified, the **WaMart** directory is not created.

❑ Verify that the directory **WaCnfg** contains a collection of data sets, including the Webmart table that was mentioned in step 6 of "Site Initialization." Verify that the Webmart table contains a single record and a column named GUID that identifies the Web site.

❑ If the repository is SAS, then the directory **WaMart** contains a collection of data sets, one per warehouse table.

❑ Verify that the directories **EvntSrc**, **WaAnly**, **WaSrc**, and **WaWork** are empty. If the repository is SAS, then the WaMart library should contain the data mart tables that were created by the Site Initialization job.

❑ If the repository is Oracle, then the WaMart library is an Oracle database, not a directory.

❑ If Oracle tables need to be created by your DBA, then complete these steps to create the table definitions that are ready for Oracle:

1 Determine in which directory the SAS Web Analytics umacros are stored. For a Windows installation, the location is typically <*your-SAS-installation-directory*>**\weba\ucmacros**. For a UNIX installation, the location is typically <*your-SAS-installation-directory*>**/ucmacros/weba**.

2 Run the following macro from that directory:

```
%wa_sas_to_oracle_ddl(
       input_dir=<your-ucmacros-installation-directory>,
       output_dir= <your-target-Oracle-DDL-directory>);
```

3. Ask your Oracle DBA to execute the resulting DDL from the file <*your-target-Oracle-DDL-directory*>oracle_ddl.txt to create the Oracle tables and their indexes.

- ❑ A table object named Webmart in the **/WaCnfg** folder for the new site. This object provides the identifier for the Web mart (in the Guid column) and is used to enable ETL and report processing.

CAUTION:

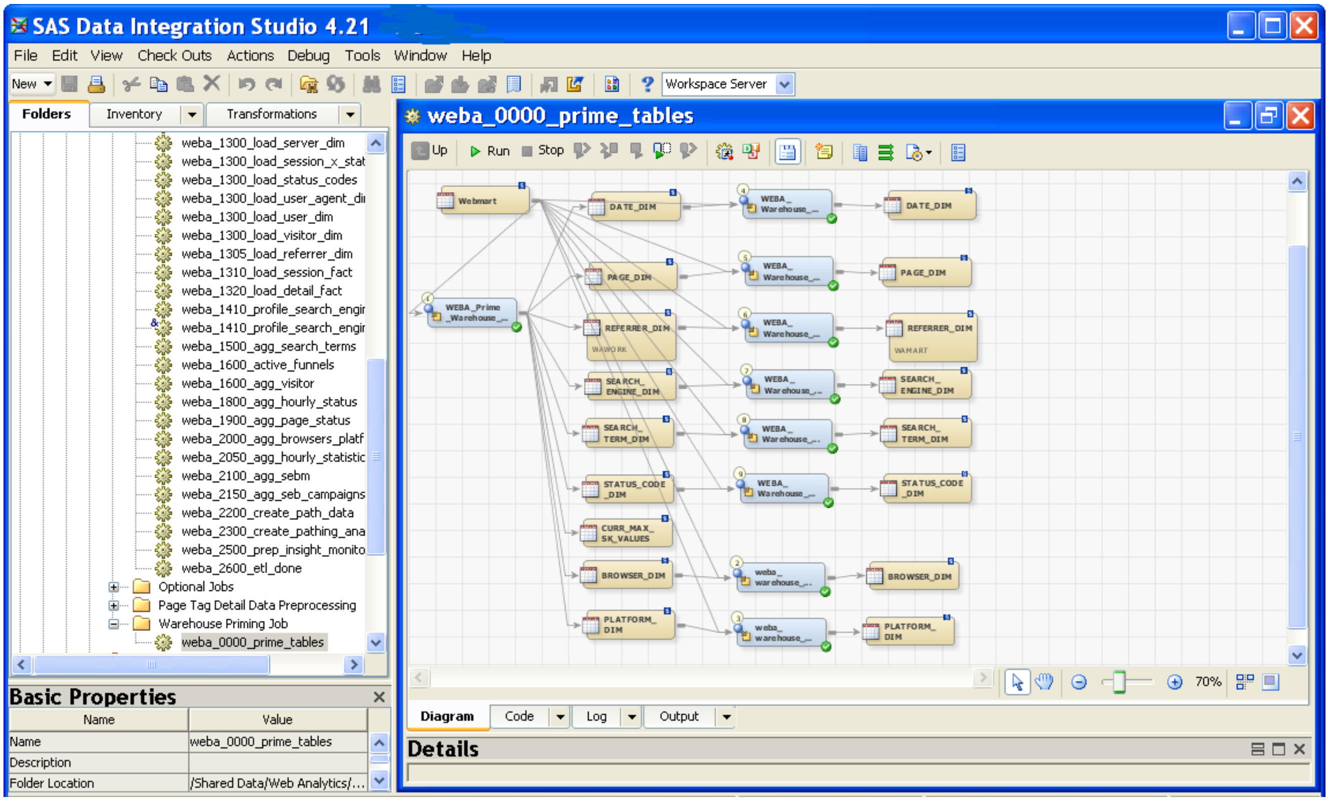
Do not delete the Webmart object from the WaCnfg directory because deleting the object completely disables the Web mart.

Prime the Web Mart Tables

After the data mart tables have been created and the indexes have been applied, you are ready to prefill the dimension tables.

1. In SAS Data Integration Studio, navigate to /<*Web mart folder*>/<*Web mart name*>/`5.4 Jobs/Warehouse Priming Job`. The <*Web mart folder*> is the specified folder within the Site Initialization job. The default folder location is `/Shared Data/Web Analytics.`

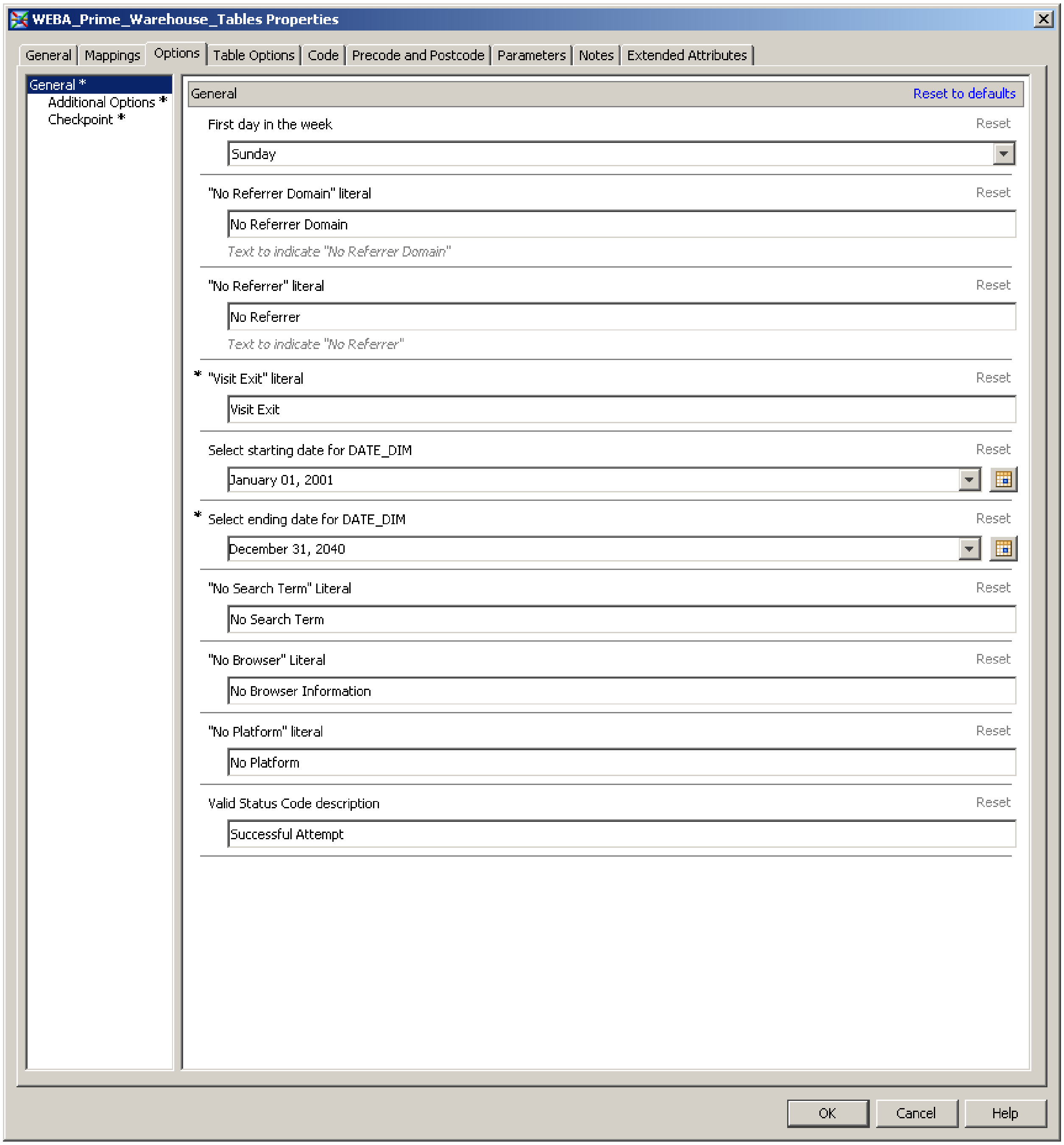
2. Double-click the **weba_0000_prime_tables** job.


SAS Data Integration Studio 4.21
File Edit View Check Outs Actions Debug Tools Window Help
New
Workspace Server
Folders
Inventory
Transformations
weba_1300_load_server_dim
weba_1300_load_session_x_stat
weba_1300_load_status_codes
weba_1300_load_user_agent_di
weba_1300_load_user_dim
weba_1300_load_visitor_dim
weba_1305_load_referrer_dim
weba_1310_load_session_fact
weba_1320_load_detail_fact
weba_1410_profile_search_engi
weba_1410_profile_search_engi
weba_1500_agg_search_terms
weba_1600_active_funnels
weba_1600_agg_visitor
weba_1800_agg_hourly_status
weba_1900_agg_page_status
weba_2000_agg_browsers_platf
weba_2050_agg_hourly_statistic
weba_2100_agg_sebm
weba_2150_agg_seb_campaigns
weba_2200_create_path_data
weba_2300_create_pathing_ana
weba_2500_prep_insight_monito
weba_2600_etl_done
Optional Jobs
Page Tag Detail Data Preprocessing
Warehouse Priming Job
weba_0000_prime_tables
weba_0000_prime_tables
Up
Run
Stop
Webmart
WEBA_Prime_Warehouse
DATE_DIM
PAGE_DIM
REFERRER_DIM
WAWORK
SEARCH_ENGINE_DIM
SEARCH_TERM_DIM
STATUS_CODE_DIM
CURR_MAX_SK_VALUES
BROWSER_DIM
PLATFORM_DIM
WEBA_Warehouse_...
weba_warehouse_...
DATE_DIM
PAGE_DIM
REFERRER_DIM
WAMART
SEARCH_ENGINE_DIM
SEARCH_TERM_DIM
STATUS_CODE_DIM
BROWSER_DIM
PLATFORM_DIM
70%
Diagram
Code
Log
Output
Details

Basic Properties

Name	Value
Name	weba_0000_prime_tables
Description	
Folder Location	/Shared Data/Web Analytics/...

3. Change the default values that are used to populate the dimensional tables by right-clicking the **weba_prime_warehouse_tables** transformation (node 1), and then clicking the **Options** tab.


WEBA_Prime_Warehouse_Tables Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
General *
Additional Options *
Checkpoint *
General
Reset to defaults
Reset
OK
Cancel
Help

First day in the week: Sunday

"No Referrer Domain" literal: No Referrer Domain

Text to indicate "No Referrer Domain"

"No Referrer" literal: No Referrer

Text to indicate "No Referrer"

* "Visit Exit" literal: Visit Exit

Select starting date for DATE_DIM: January 01, 2001

* Select ending date for DATE_DIM: December 31, 2040

"No Search Term" Literal: No Search Term

"No Browser" Literal: No Browser Information

"No Platform" literal: No Platform

Valid Status Code description: Successful Attempt

Here are the tables and their associated options:

Date_Dim table contains calendar information.

First day in the week.

Specifies any day as the first day of the week. The default value is Sunday.

Select starting date for DATE_DIM.

The starting date is used as the first day within the Date_Dim table. The default value is January 1, 2001.

Select ending date for DATE_DIM.

The ending date is used as the ending date during the Date_Dim table population. The default value is December 31, 2040.

Referrer_Dim table is primed with a record when the referrer value is blank within the Web server or page tag server log record.

No Referrer Domain literal.

Is used to load the Referrer_Domain_Nm field. The default is No Referrer Domain.

No Referrer literal.

Is used to load the Referrer_Txt field. The default is No Referrer.

Page_Dim table is primed with a record that indicates the visit has ended. The text is used for reporting purposes.

Visit exit literal.

The default value is Visit Exit. The value of Search_Term is used to prime the table with a record when the search term is blank within the Web server or page tag server log record.

No Search Term literal.

The default value is No Search Term.

Browser_Dim table is primed with a record when the browser name is missing in the User Agent value within the Web server or page tag server log record.

No Browser literal.

The default value is No Browser Information.

Platform_ Dim table is primed with a record when the browser name is missing in the User Agent value within the Web server or page tag server log record.

No Platform literal.

The default value is No Browser Information.

Status_Code_Dim table is completely loaded within the job with a set of valid status codes. An additional record is added when the status code is missing within a Web server log record.

Valid status code description.

The default value is Successful Attempt.

4. After any customizations are completed, click **OK**.
5. Run the job.

Your site is now ready to deploy and schedule the ETL. For more information, see Chapter 3, “The SAS Web Analytics ETL Process.”

Setting Up the Clickstream Job

Introduction to the Clickstream Job

In order to accurately capture Clickstream data, you must copy the Clickstream Template job, and then modify it for your site.

To set up the Clickstream job:

1. Set up the correct Clickstream job template and specific fields.

   For more information, see Chapter 9, "Using Template Jobs" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide.*

2. Set up the Clickstream job.

   For more information, see Chapter 10, "Performing Common Tasks" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

3. Set up internal search tracking.

   For more information, see Chapter 10, "Tracking Internal Searches" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

4. Set up the SAS Page tag. Make sure the tag that captures information to feed data to new tables is set up properly.

   For more information, see Chapter 5, "Preparing SAS Page Tag Data" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

The SAS page tag writes three types of records to the page tagging log:

Load	the page loaded in the browser
Click	a visitor clicked on a link in a page
Submit	a visitor clicked on a submit button on a page

For information about visitor clicks in the page tagging log, see Chapter 3, "Data Source Preparation" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

Paid_Search_Flg Field

The Paid_Search_Flg field is a one-byte character field in the Clickstream Data Surveyor detail data set. It is set to **1** if its row pertains to a pay-per-click request. Otherwise, the field is set to **0**. Determining whether a row pertains to a pay-per-click request is based on user-defined rules. To set up the Paid_Search _Flg field, you must add user columns. For more information, see "Maintaining User Columns" in the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide.*

Troubleshooting Site Initialization

Problem	"Insufficient memory" message appears while importing one of the metadata packages.
Solution	An "Insufficient memory" message as a result of a CORBA.NO_MEMORY exception means that the metadata server itself ran out of memory. For more information, see SAS

Note 33554, available at **http://support.sas.com/kb/33/554.html**.

Problem

The newly initialized Web mart is not visible in Weba.Webmarts.

Solution

A Web mart does not show up in Weba.Webmarts until the first ETL has been run successfully. Because Weba.Webmarts is used as the prompt for all of the SAS Web Report Studio reports, users of the Web application need to select a Web mart that already contains data.

Problem

The following error message appears:

```
ERROR: The Batch Jobs deployment directory is not
associated with SAS Application server, xxxxxx.
```

Solution

This error occurred because a deployment directory called Batch Jobs was not created during the configuration of the Data Step Batch Server. For information about creating the Batch Jobs deployment directory, see “Defining a SAS DATA Step Batch Server,” in *Scheduling in SAS 9.2,* available at **http://support.sas.com/documentation/**.

Problem

A user is not able to see SAS Web Analytics reports.

Solution

A user needs to be a member of one of the SAS Web Analytics user groups to see any of the Web Analytics reports. See "SAS Web Analytics Security Setup" in Chapter 1 to determine which user group to assign users to.

Problem

The following error messages appear:

```
ERROR: The selected job options request the
recreation of metadata for an existing Web mart,
but the following Web mart directories do not
exist: WaMart

ERROR: Contact the SAS Web Analytics Administrator
for assistance.
```

Solution

If one or more of the Web mart directories exist in another location, do the following:

1. Create the missing directories in the root directory.
2. After the site metadata has been recreated, edit the path information of the affected libraries.

If the message occurred because one or more Web mart directories do not exist, a new site should be created.

CHAPTER 3

The SAS Web Analytics ETL Process

Overview of the SAS Web Analytics ETL Process

Understanding the Web Data ETL Process

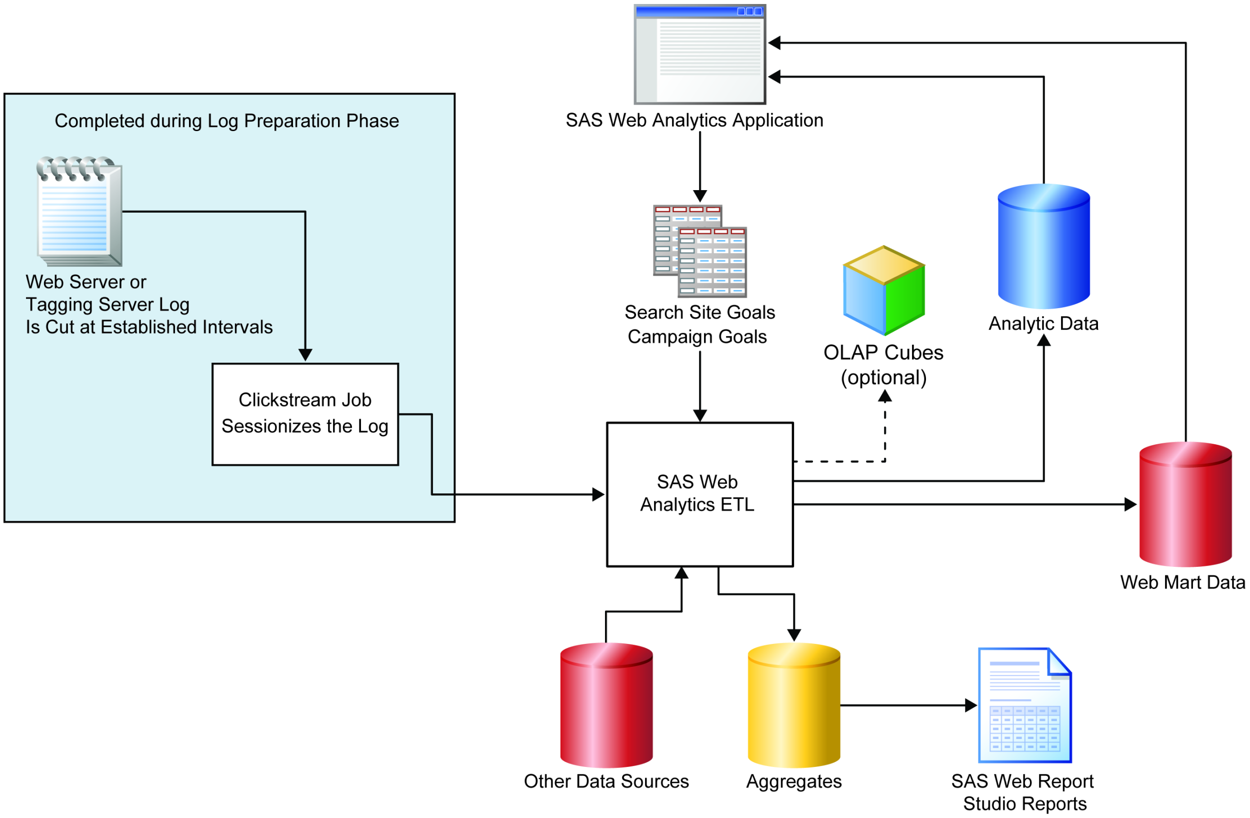
The Overall ETL Process

The overall ETL process consists of four parts:

- ❑ Web site data is acquired (external to the SAS System).
- ❑ Web site data is grouped by a single visitor, or sessionized (using SAS Data Surveyor for Clickstream Data).
- ❑ The data repository in the SAS Web Analytics Web mart is loaded (using SAS Web Analytics).
- ❑ The Web site search site goals and campaign goals are set.

Data is acquired from Web server logs or a tagging server log. A site's Web master schedules the intervals at which the Web server logs are created. The data in these logs is sessionized (grouped by a single visitor) using a customized SAS Data Surveyor for Clickstream Data job, and then transformed into a single detail output data set. The output data set is used in the SAS Web Analytics ETL process to load the tables of the Web mart. (Optional) Other data sources can be incorporated into the ETL process. After the Web mart tables are loaded, the SAS Web Analytics aggregates and analytic data are created or updated. The following figure shows how the overall process works with the SAS Web Analytics ETL.

Figure 3.1: How Data Flows in and out of the SAS Web Analytics ETL



The data that is contained in the Web mart can then be exploited and analyzed through the SAS Web Analytics application and SAS Web Report Studio reports. If OLAP Cubes are created, which is optional, they can be accessed through SAS Web Report Studio, SAS Enterprise Guide, or the SAS BI Dashboard.

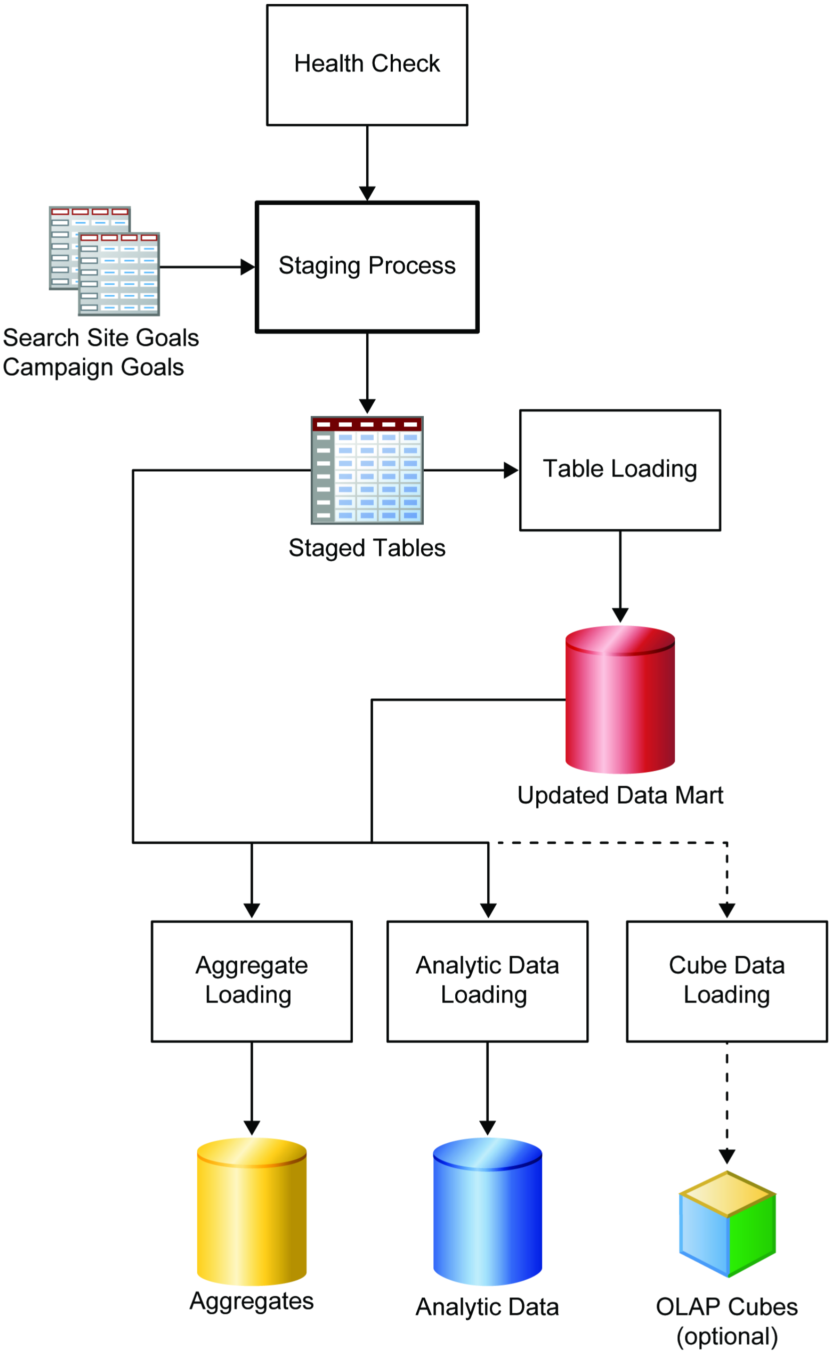
The Web Analytics ETL

The SAS Web Analytics ETL process comprises a set of jobs that are defined in SAS Data Integration Studio and the process does the following:

1. Validates certain characteristics of the data that was received from the SAS Data Surveyor for Clickstream Data job during the health check.
2. Reads the validated data while incorporating the goals that were defined in the SAS Web Analytics application. The ETL process prepares the data by creating staged tables during the staging process.
3. Loads the staged tables into the Web data mart during table loading.
4. Updates the aggregates by using the staged tables and the Web data mart during aggregate loading.
5. Updates the analytic data by using the staged tables and the Web data mart during analytic data loading.
6. (Optional) Updates the cubes by using the staged tables and the Web data mart during cube data loading.

The following figure illustrates the SAS Web Analytics ETL process.

Figure 3.2: Inside the Web Analytics ETL Process



Setting Up the ETL Process

To set up the ETL process:

1. Verify that the Site Initialization job has been run. For more information, see Chapter 2, “Setting Up a Web Mart.”
2. Add and set up the Clickstream jobs. For more information, see “Setting up the Clickstream Job” in Chapter 2. Verify that the Clickstream jobs result in at least one data set. The data set name must begin with the prefix that was

defined in the **input filename prefix** option of the Weba_1100_Health_Check job. The default is Weblog_Detail**.** The data set must be placed in the `WaSrc` directory before you run the Weba_1100 job.

3 Customize the ETL jobs. For more information, see the following section.

4 Check for missing values in the **Domain** field of the Clickstream output data set. Update the Clickstream job to set the **Domain** field value only when values are missing.

5 Determine whether custom code needs to be added to the Weba_1300_Load_Domain_Dim job to differentiate between internal and external domains in the Clickstream output data. For more information, see "Setting Internal and External Domains" on page 48**.**

6 Deploy and schedule ETL jobs in the <*Web mart*>`/5.4 Jobs/ETL Jobs` folder by using the Platform Load Sharing Facility (LSF) or another scheduler. If you have a small mart, or if you want to perform testing before scheduling the jobs, you can use a script that SAS Web Analytics provides to run the ETL process. The script deploys jobs automatically, in sequence. For more information about running the script, see Appendix 7, "The ETL Jobs."

Introduction to the SAS Web Analytics ETL Jobs

ETL jobs can be divided into groups according to their function.

❑ The Clickstream jobs read Web or tagging logs, and group into one session (sessionize) a single visitor's data. For more information, see the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

❑ The Health Check job (1100) validates the Clickstream data.

❑ Jobs 1200–1230 stage the Clickstream data.

❑ Jobs 1300–1330 load the SAS Web Analytics data mart tables.

❑ Jobs 1500–2150 update aggregate rollup tables.

❑ Jobs 1410, 1600, 2200, 2500, and 2501 prepare analytic report data.

❑ Job 2600 closes the ETL process.

Appendix 7 lists all of the jobs that make up the ETL schedule for SAS Web Analytics. If a job is optional, then the ETL administrator decides whether to run the job. For more information, see "Optional Jobs" on page 76.

If a job can be customized, then the ETL administrator can update the job to meet the needs of an individual Web site.

Note: Jobs with identical numbers can run simultaneously. For example, there are many jobs numbered 1300. All of these can run simultaneously during the ETL process, provided that all jobs numbered less than 1300 have completed successfully.

Customizing the ETL Jobs

Introduction to Customizing Jobs

Many of the SAS Web Analytics ETL jobs can be customized through options that are available. Before you make changes to any ETL job, complete the following steps:

1 Back up all metadata for the site.

2 Make a copy of the job and customize the copy rather than the original job.

3 Verify that the change will not affect other jobs.

Health Check Job

Introduction to the Heath Check Process

The health check process screens the incoming data to help prevent duplicate or incorrect data from being loaded into the Web mart. The health check process also updates the Web mart status to Active. A number of options are available to tune the Heath Check job.

Validate the Input Data

The Weba_1100_Health_Check job exits with an error status if an input file is invalid or has already been loaded. To correct this error, examine the input file, reset any relevant options, and then rerun the Weba_1100_Health_Check job.

Note: You can ignore the error and run the next job in the scheduled ETL process. However, the rest of the ETL jobs should not be run unless you fully understand the cause and implications of any error that was generated by the Health Check job.

The Health Check job keeps track of which files it has loaded or is in the process of loading, which prevents the same ETL from accidentally being triggered more than once.

The Health Check job performs several tasks:

- ❑ obtains a list of data sets that are currently in its WaSrc library.
- ❑ reduces the list to those data sets with an appropriate prefix name.
- ❑ ensures that each data set has required columns.
- ❑ determines whether any of the data sets within WaSrc have already been loaded.
- ❑ outputs information about each input file to the WaCnfg.Cnfg_Input_Hist data set. Sets the progress_check column to **1** to indicate that this file is in use by the current ETL process.
- ❑ creates the following output data sets:

  - a single data set that is read by the Weba_1200_Warehouse_Staging_Tables job in the current ETL process to create the staging tables. The data set was created from the concatenation of all the qualified data sets in the WaSrc library.

  - a data set that contains CLICK and SUBMIT event records from tagging server logs. The EvntSrc.Event_Detail data set is used in the ETL 1230 job, which populates the link staging tables.

  - a data set that contains the records for all open sessions. The WaWork.Open_Sess data set is not used by the ETL process, but is available so that all records coming from the Clickstream input data sets can be traced.

The Weba_1100_Health_Check job consists of several tables and a custom transformation, WEBA_Input_Data_Health_Check. Open the transformation Properties window in SAS Data Integration Studio to modify the default values of the options. The transformation enables you to customize the job through several options:

Record Threshold? Yes or No

checks the number of records in an input file that fall within an allowable variance (the Threshold Percentage) of the number of records in previously loaded files. A low number of records in the Weblog detail data set could indicate that there were Weblog cutting problems. For example: Assume a typical input of 750,000 records per day. A detail set for a day contains 100 records. It is possible that the log file was cut prematurely. The default is Yes.

Threshold Percentage

represents the maximum allowed percentage difference between the record count in the current file and the average record count of previously loaded files. The number of files whose record counts are averaged is determined by the **Threshold Definition** option. The number must be between 0 and 100. The default is 95.

Threshold Definition

is the number of past successful file loads that should be used for the average number of records per file in the **Record Threshold** option. The default is seven file loads.

Input filename prefix

defines the prefix portion of the name of the data sets to be considered for use in the ETL process. Typically, this is the prefix that is used to name the output data sets in SAS Data Surveyor for Clickstream Data. The default is Weblog_Detail.

Check for gaps between hours? Yes or No

checks data records for gaps in the datetime values between records. The check determines whether there are records for all hours within the Detail data set, and produces an error if a gap is too large. For example, suppose that Weblogs are cut for a site at 00:15 each day, and the expectation is that there would be records in each hour of the day. If no records reflect the hours 08:00–10:00, the ETL stops processing to allow the problem to be determined before continuing. The default is Yes.

Allowed datetime gaps in data

represents the maximum allowed gap in hours in the datetime values between records. The number must be an integer. The default is 1 hour.

Automatically adjust input file field lengths if necessary

adjusts field lengths to SAS Web Analytics specifications. Using this option might avoid warning messages in the log that indicate that the input data column lengths are different from the expected lengths. The default value is Yes.

Note:

- These warning messages are most likely to occur when Clickstream data has been adjusted for UTF-8 encoding. It is not necessary to use UTF-8 encoded input data to create a SAS Web Analytics data mart that is encoded as UTF-8; standard Clickstream data can be used as input.
- If Clickstream data is converted to UTF-8 encoding, it is not necessary to expand the columns. The Clickstream data lengths can accommodate MBCS data.
- If Clickstream data is converted to UTF-8 encoding and the columns are expanded, do not expand the columns to lengths greater than those expected by the SAS Web Analytics data mart. This will result in errors

from the Health Check job. The SAS Web Analytics data mart has already been expanded by approximately 30% on selected columns to accommodate MBCS data.

Byte denominator

is a value that determines the units that are used to store the Bytes Sent and Received fields. A value of 1 stores the values in bytes, a value of 1,024 stores the values in KB, and a value of 1,048,576 stores values in MB. The default value is 1,024.

Health Checks Performed

Table 3.1: Health Checks Performed: Automatic Checks

Check	Description	Required? (Yes/No)
Domain = ' '	Terminate the ETL process if any record in the input data has DOMAIN equal to missing.	Yes
Check for Session_IDs that have rows where Session_Closed equals both **0** and **1**	Terminate the ETL process if an input data set contains rows where SESSION_CLOSED=0 and SESSION_CLOSED=1 for the same Session_ID value. A closed session is one whose activity is completely contained in the current Clickstream input data set. An open session is one whose activity might be continued into a subsequent Clickstream input data set. If there has been a problem in the Clickstream job or if multiple Clickstream data sets have been manually concatenated into a single input data set, the mixing of records can report a session as both open and closed in the WaSrc directory. If you are using the weba_run_etl.sas script to run your ETL process and you have specified Create_Wasrc_Detail=Y and Run_Style=C, records for open sessions are automatically removed. This is done during the concatenation process to avoid this situation.	Yes
Ensure that input data exists	Terminate the ETL process if the WaSrc library does not contain at least one input data set whose name begins with the value of the Input Filename Prefix option. The default value is Weblog_Detail.	Yes
Check for previously loaded files	Terminate the ETL process if an input data set has already been loaded. This check compares information about an input data set with information about previously loaded data sets that are stored in the WaCnfg.Cnfg_Input_Hist data set. The minimum and maximum record ID values of the input data sets (generated by the Clickstream job) are compared with those of previously loaded data sets. If a match is found, it is assumed that all or part of the data set has already been loaded and the ETL process is terminated.	Yes
Check that an ETL is not already in progress	Terminate the requested ETL process if the WaCnfg.Weba_Signal data set exists. This data set is created at the end of the Weba_1100_Health_Check job, and is typically deleted by the Weba_2600_ETL_Done job after the ETL process is successfully completed. If it exists, then an ETL process is actively running, or an ETL process terminated abnormally and the Weba_2600_ETL_Done job did not run. If this is the case, resolve the problem, and then run the optional job Weba_Reset_ETL. This deletes the WaCnfg.Weba_Signal file and enables you to run a new ETL process.	Yes

Staging Jobs

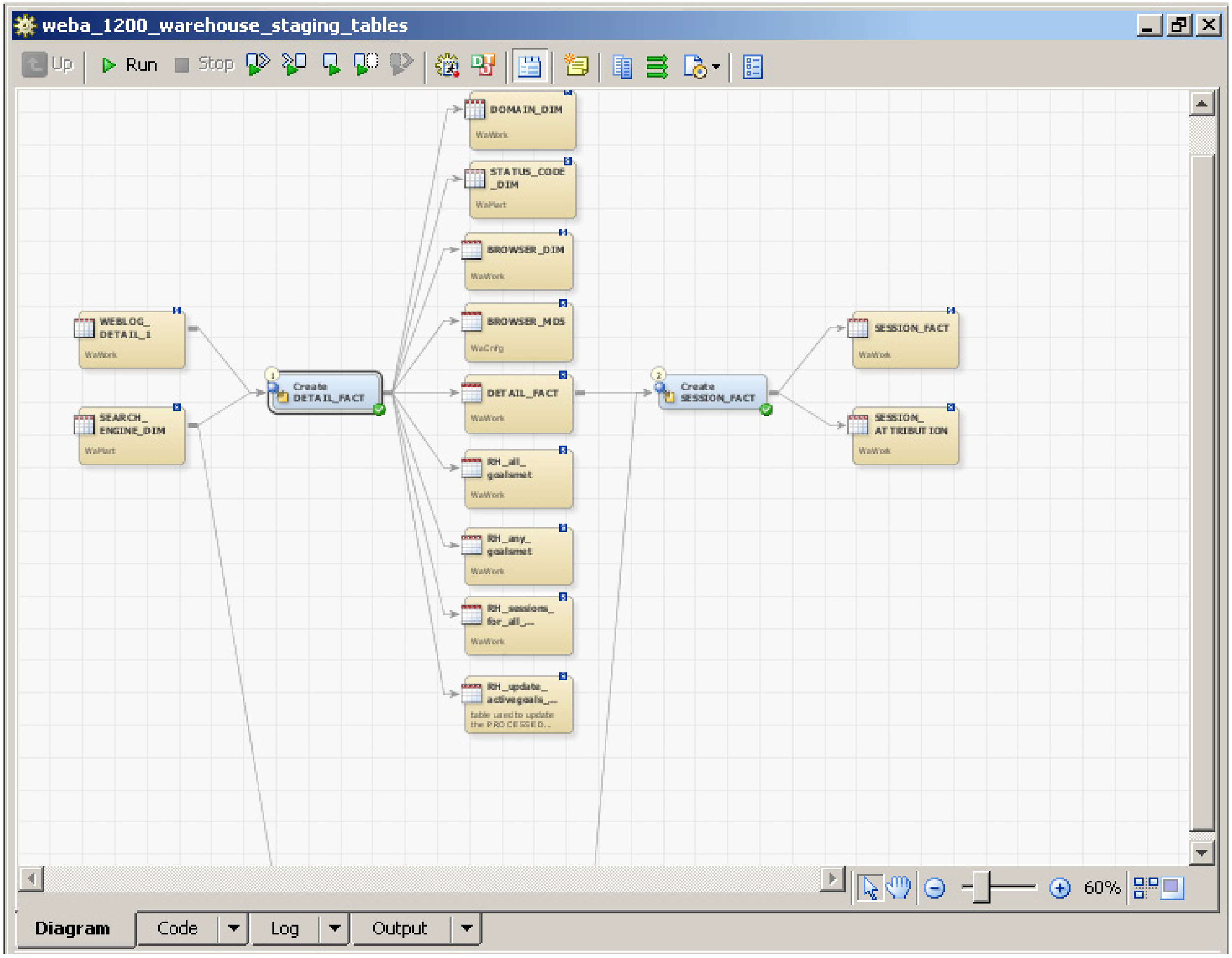
Introduction to Staging Jobs

The staging jobs are used to prepare the data for loading into the data mart tables.

- The Weba_1200_Warehouse_Staging_Tables job prepares data for loading into the Web data mart.
- The Weba_1220_Goal_Staging_Tables job prepares data for loading goal tables.
- The Weba_1230_Link_Staging_Tables job prepares data for loading link tables.

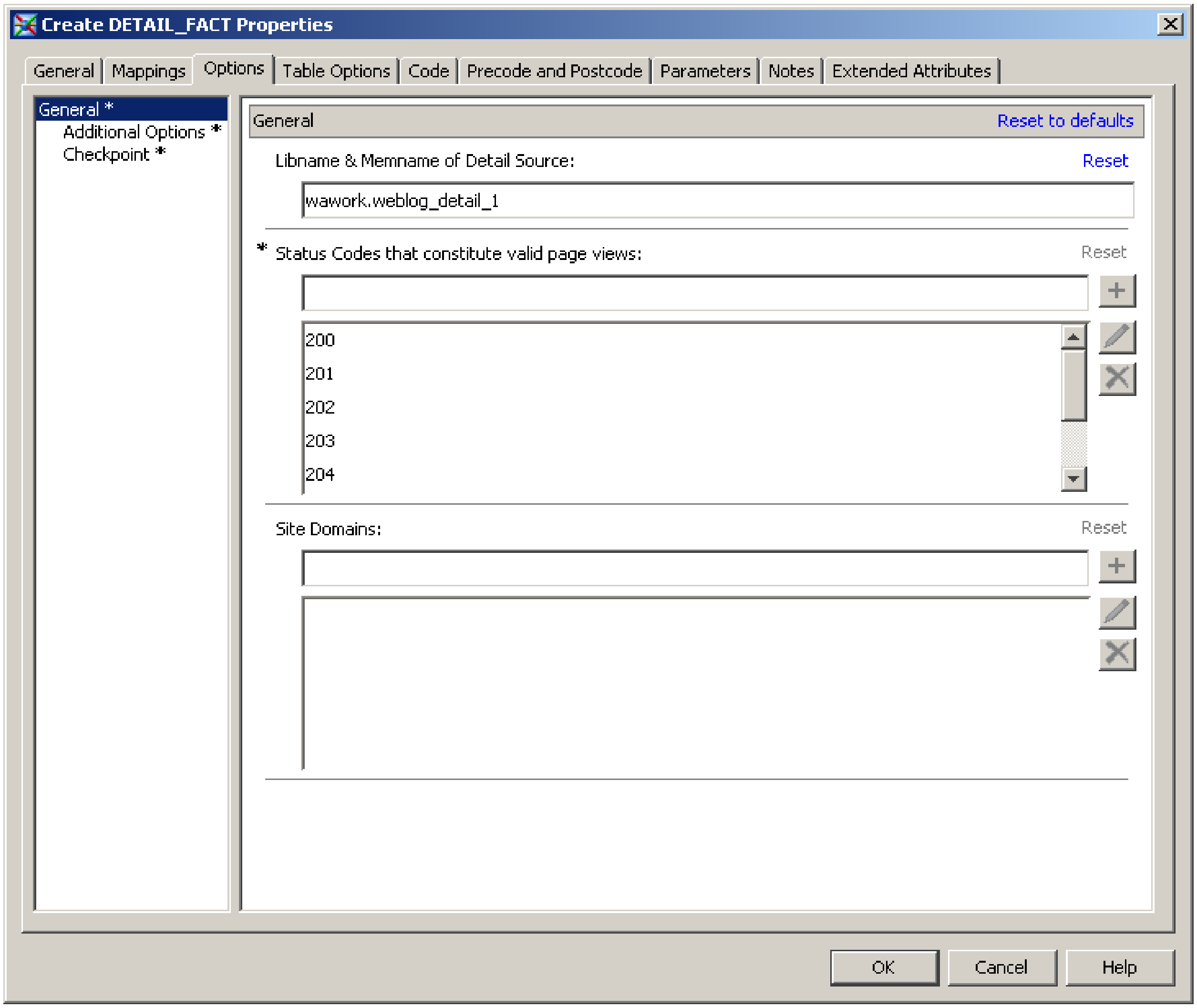
Weba_1200_Warehouse_Staging_Tables Job

The Weba_1200_Warehouse_Staging_Tables job reads the detail data set that is created by the Weba_1100_Health_Check job, creates staging tables, and prepares the data for loading into the Web data mart. The input data set name is determined by the Input filename prefix option of the Health Check job, whose default value is WaWork.Weblog_Detail_1. If this name has been changed, record the name in the **Libname & Memname of Detail Source** field.



The Weba_1200_Warehouse_Staging_Tables job creates the staging tables in the WaWork library for the following tables:

Browser_Dim
Domain_Dim
Detail_Fact
Int_Search_Term_Dim
Ip_Address_Dim
Origination_Dim
Page_Dim
Platform_Dim
Referrer_Dim
Referrer_Query_String_Dim
Search_Term_Dim
Server_Dim
Session_Attribution
Session_Fact
Session_X_Status_Code
Status_Codes
User_Agent_Dim
Visitor_Dim



Libname & Memname of Detail Source: wawork.weblog_detail_1

* Status Codes that constitute valid page views:

200
201
202
203
204

Site Domains:

Table 3.2: Parameters That Can Be Customized within the Create_Detail_Fact Node of the Weba_1200_Warehouse_Staging_Tables Job

Node	Parameter	Description	Use
Create Detail Fact	Libname and Memname of detail source	Name of the output data set from the Weba_1100_Health_Check job. The default value is WaWork.Weblog_Detail_1.	Change the name that is expected for the Detail Source data set. The Detail Source data set name should correspond to the name of the output data set that is created by the Health Check job.
	Status codes that constitute valid page views	List of status codes that indicate that the visitor of a site has viewed a page. The default values are 200, 201, 202, 203, 204, 205, 206, and 304.	Add or remove status codes that indicate a page was viewed.
	Site Domains	List of internal domains that are associated with a site. The default is blank.	List all internal domains for a Web data mart.

Weba_1220_Goal_Staging_Tables Job

Users that are assigned to either the Web Analytics Analyst or the Web Analytics Administrator group can select search term goal pages in the SAS Web Analytics application. The Weba_1220_Goal_Staging_Tables job uses the staged Detail_Fact table and the Search Term Goal Configuration table to create staging tables for the Goal_Dim and Goal_X_Session tables.

For information about setting search term goals, see the *SAS Web Analytics: User's Guide*.

Weba_1230_Link_Staging_Tables Job

This job stages the tables that will be loaded in other jobs. Its function is similar to the function of the Weba_1200 job. The Weba_1230_Link_Staging_Tables job reads the Evntsrc.Event_Detail data set that was produced within the Weba_1100_Health_Check job, which contains Click and Submit data. The Weba_1230_Link_Staging_Tables job also reads the WaWork.Detail_Fact data set from the Weba_1200_Warehouse_Staging_Tables job, which contains Load data. From these two data sets, the Weba_1230_Link_Staging_Tables job produces the link dimension (WaWork.Link_Dim) and cross-reference (WaWork.Detail_X_Link) staging tables.

The values of fields within the Link_Dim table depend on how the anchor tags (links) within a page of a Web site are set up. A Web developer adds parameters to an anchor tag to assign values to these fields:

Link_Nm	a link name.
Link_ID	a link ID.
Link_Alt_Txt	alternate link text.

Note: To display the number of clicks on a link by using the Page Overlay feature, each link's anchor tag must have an ID parameter (Link_ID) with a unique value assigned. For more information about the Page Overlay feature, see the *SAS Web Analytics User's Guide*.

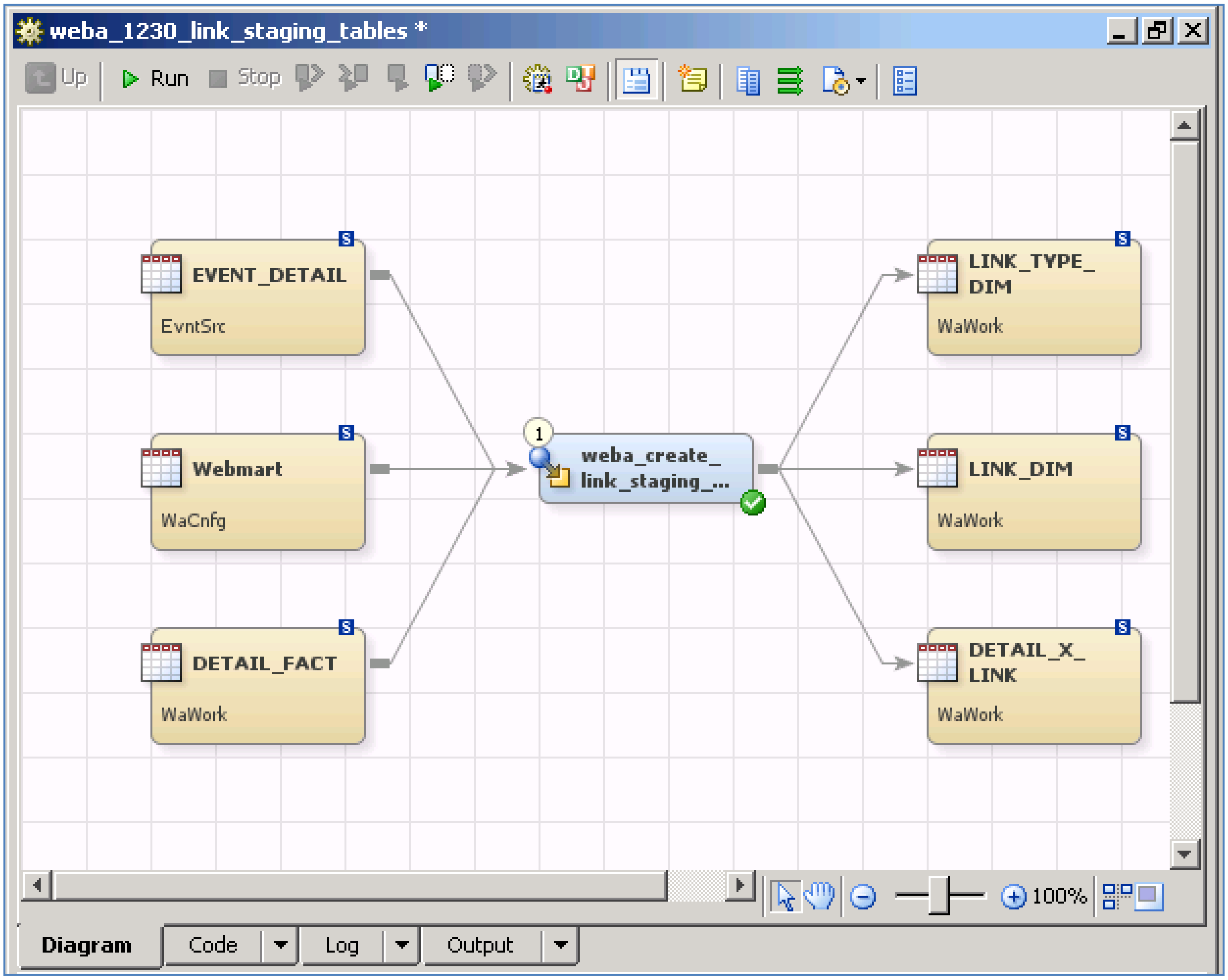
For information about page tagging and setting up links, see the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

Setting the Download Flag

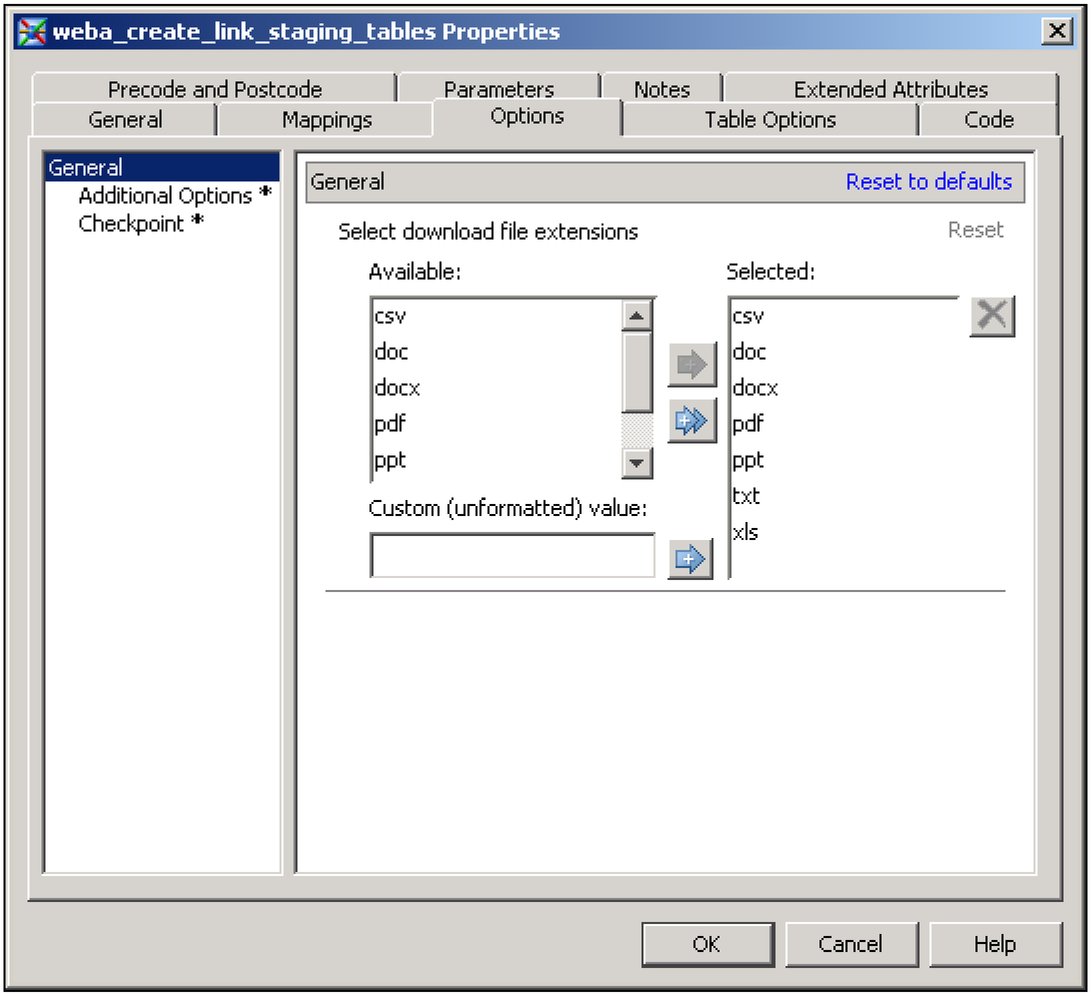
The Download_Flg field in the Link_Dim table is set using the file extension of the link URI. By default, the following file extensions are used to set the Download_Flg to 1: .csf, .doc, .docx, .pdf, .ppt, .txt, and .xls. To add new file extensions to the downloaded file extension list:

1 Open the Weba_1230_Link_Staging_Tables job.

2 Right-click the transformation.



3 Click the **Options** tab and enter values in the **Custom value** box.



Select download file extensions

Available:

csv
doc
docx
pdf
ppt

Custom (unformatted) value:

Selected:

csv
doc
docx
pdf
ppt
txt
xls

Load Jobs

Introduction to Load Jobs

The Load jobs append staged, incoming data to the existing dimension, fact, and cross-reference tables in the Web mart. Some of the Load jobs also update columns in pre-existing rows of the tables to which they add data. All Load jobs begin with Weba_13xx_. Jobs that have the same numeric value as part of their name (for example, 1300) can be run at the same time. Jobs that have higher numbers in their names (for example, 1310) must be run after all lower-numbered jobs have completed.

All the Load jobs use a custom transformation named Weba_Warehouse_Table_Loader, which is referred to as the Table Loader transformation in this document. This transformation contains options that you can set for enhanced performance. This transformation also contains the options that control updating columns in existing data rows. These options (Key Field and Field to Update) should typically not be modified.

Performance Tuning Options for the Load Jobs

Several options on the Table Loader transformation are available to potentially improve loading performance. Some options are available only when the Web mart is stored in Oracle. Other options are available for Web marts that are stored in both SAS and Oracle.

To modify the performance tuning options, open a Loading job, and then open the Properties window of the Table Loader transformation in SAS Data Integration Studio and edit the following options. Redeploy the job after making the modifications.

Bulkload records? Yes, No, Use Threshold

is an option available only for Web marts that are stored in Oracle. When set to Yes, this option causes SAS/ACCESS Interface to Oracle to call the Oracle SQL*Loader (SQLLDR). The Oracle bulk loader provides superior load performance, so you can rapidly move data from a SAS file into an Oracle table. Setting this option to No bypasses the Oracle bulk loader. Setting the option to Use Threshold indicates that a value will be specified for the Threshold option, defining a number of records in a staging data set to be appended. Data sets with more than this number of records will be appended using the Oracle bulk loader. Data sets with fewer than this number of records will be appended without using the Oracle bulk loader. The default value is Yes.

Note: The SQL*Loader direct-path load has a number of limitations. For more information, including tips to boost performance, see the Oracle utilities documentation. You can also view the SQL*Loader log file instead of the SAS log for information about the load when you use bulk load. Also see the SAS/ACCESS Interface to Oracle documentation.

When Bulkload is selected as a loading option, either explicitly or conditionally, SAS saves the Oracle bulk load log separately from the SAS log. For information about saving the log in an alternate location using the ORA_LOGDIR option, see “Specifying an Alternate Directory for ORA_LOGDIR” in Appendix 4.

Drop indexes?

is an option available for Web marts that are stored in either SAS or Oracle. If indexes are kept during an append, the append process is slower than if the indexes are dropped. However, the lack of indexes and any integrity constraints that use the indexes might allow undesired data records to be added to the data file. These undesired data records would violate the specifications of the indexes and constraints. When the indexes and constraints are rebuilt after the append process is complete, errors could result. It is then the administrator's responsibility to restore the data to a previous version, resolve the problem, and rerun the ETL process. The default value is No.

Threshold

is an option available only for Web marts that are stored in Oracle. This option specifies the minimum number of records a staging data set must have for the Oracle bulk loader to be used, when the Bulkload Records option equals Use Threshold. The default value is 0.

For a list of all load jobs, see Appendix 7.

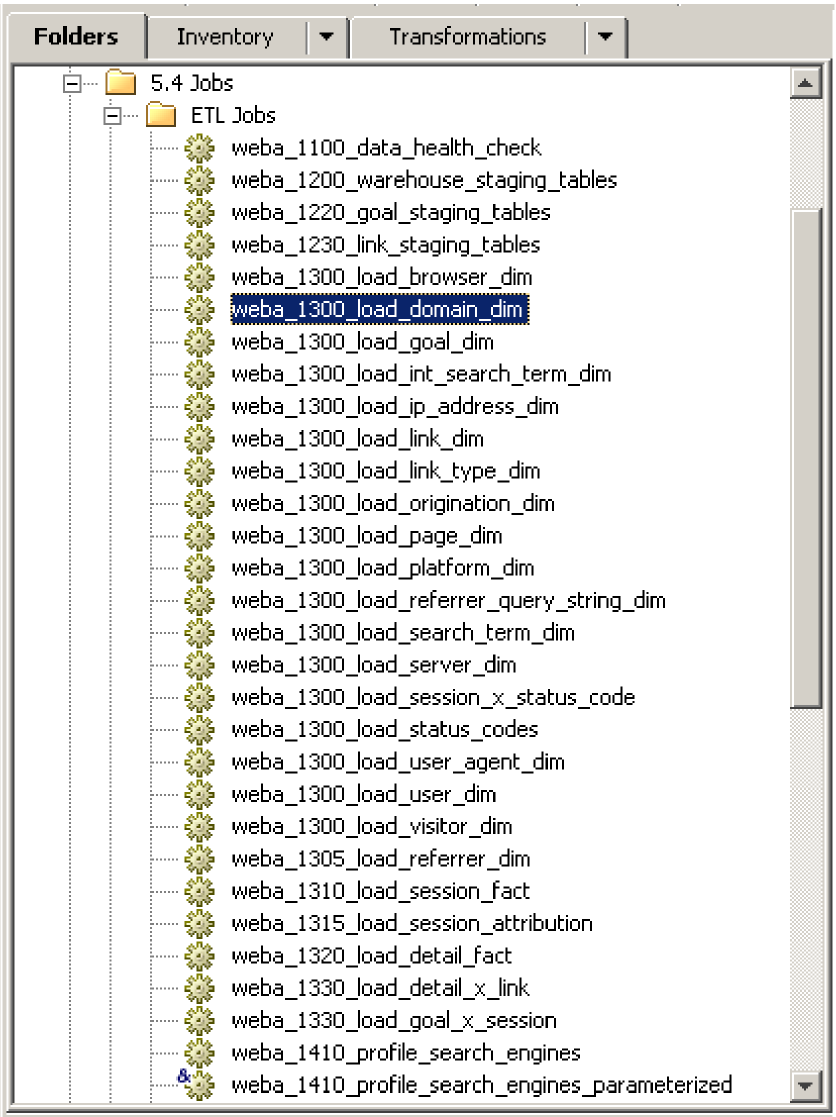
Setting Internal and External Domains

The Domain_Dim table contains the domains of the pages that were loaded into the Page_Dim table. By default, all domains are considered part of the Web site; that is, internal. However, it is possible that page tag logs contain pages that are not part of the Web site. This means that the logs are part of an external domain. The external page records can show up within the page tag for these reasons:

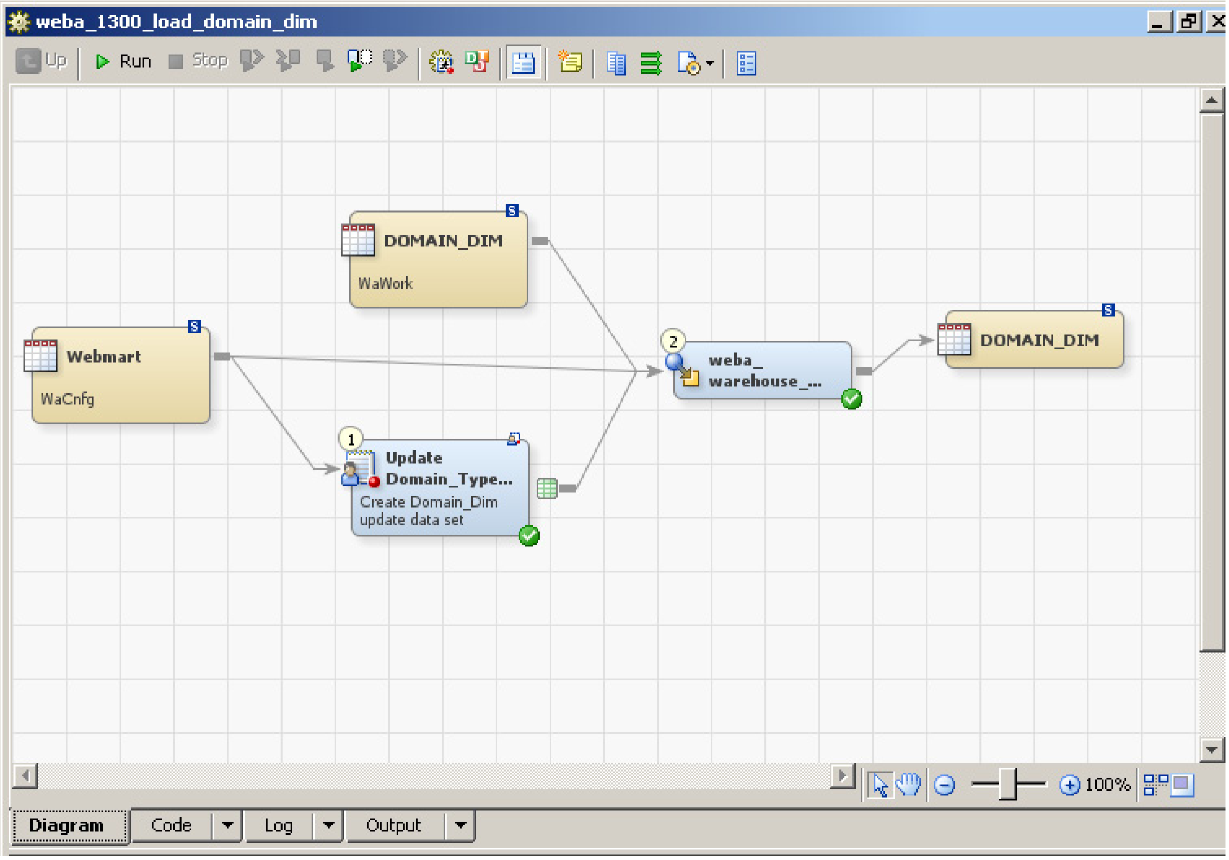
- The collection server is not set correctly within the page tag.
- A single collection server is being used for multiple domains.
- There is interest in tracing the visitor behavior between domains.

To designate a domain as external, the Domain_Type_Cd column in the Domain_Dim table needs to change from I (the default) to E. You can add code to set the Domain_Type_Cd to E, by modifying the Weba_1300_Load_Domain_Dim job. To modify the job:

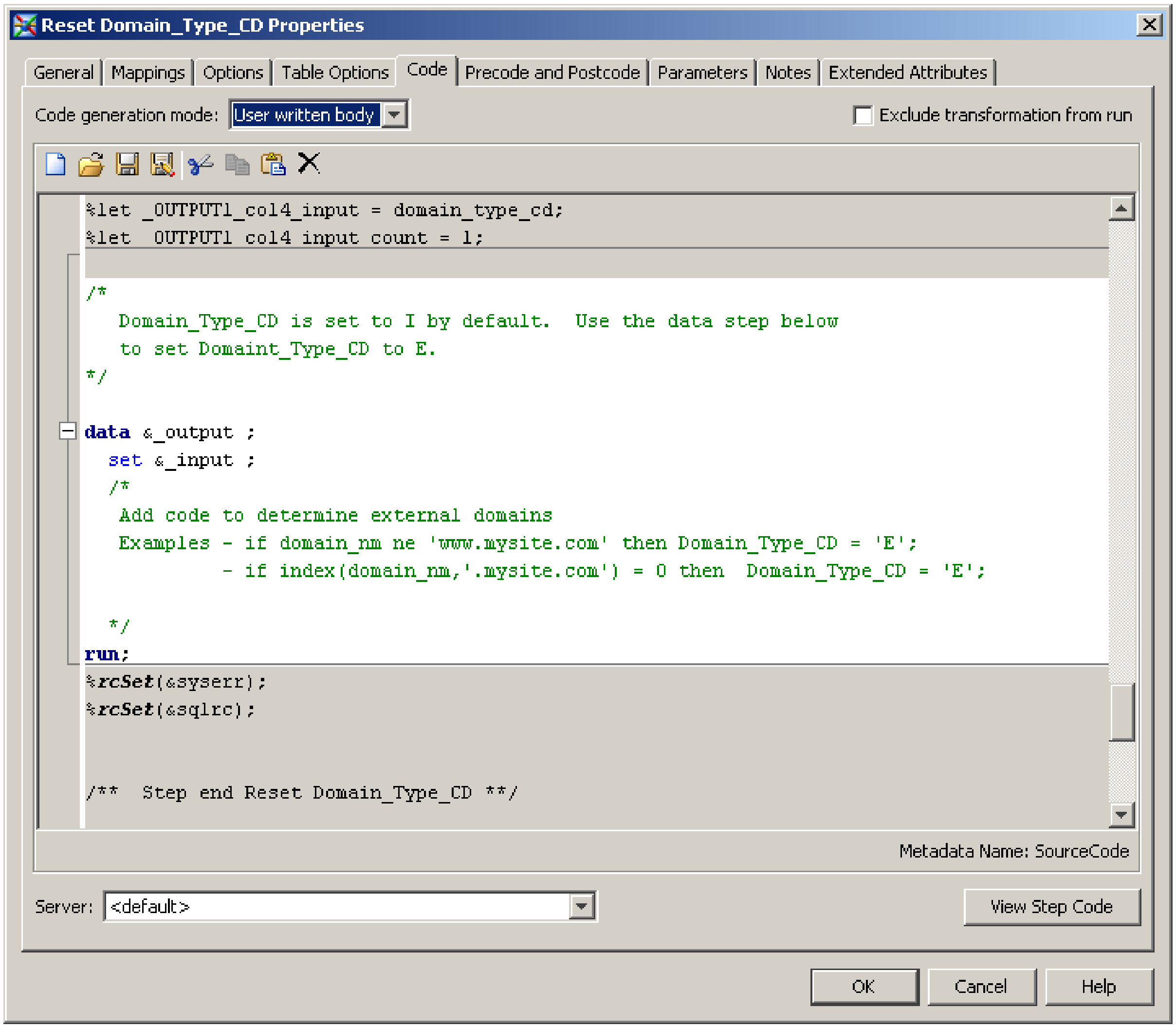
1 Open Data Integration Studio and navigate to the folder <*Web mart*>**/5.4 Jobs/ETL Jobs.**


Folders
Inventory
Transformations
5.4 Jobs
ETL Jobs
weba_1100_data_health_check
weba_1200_warehouse_staging_tables
weba_1220_goal_staging_tables
weba_1230_link_staging_tables
weba_1300_load_browser_dim
weba_1300_load_domain_dim
weba_1300_load_goal_dim
weba_1300_load_int_search_term_dim
weba_1300_load_ip_address_dim
weba_1300_load_link_dim
weba_1300_load_link_type_dim
weba_1300_load_origination_dim
weba_1300_load_page_dim
weba_1300_load_platform_dim
weba_1300_load_referrer_query_string_dim
weba_1300_load_search_term_dim
weba_1300_load_server_dim
weba_1300_load_session_x_status_code
weba_1300_load_status_codes
weba_1300_load_user_agent_dim
weba_1300_load_user_dim
weba_1300_load_visitor_dim
weba_1305_load_referrer_dim
weba_1310_load_session_fact
weba_1315_load_session_attribution
weba_1320_load_detail_fact
weba_1330_load_detail_x_link
weba_1330_load_goal_x_session
weba_1410_profile_search_engines
weba_1410_profile_search_engines_parameterized

2 Open the Weba_1300_Load_Domain_Dim job.



3 Double-click **node 1** (named Reset Domain_Type_Cd), and then click the **Code** tab.


Reset Domain_Type_CD Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
Metadata Name: SourceCode
View Step Code
OK
Cancel
Help

Code generation mode: User written body

□ Exclude transformation from run

```
%let _OUTPUT1_col4_input = domain_type_cd;
%let _OUTPUT1_col4_input_count = 1;

/*
   Domain_Type_CD is set to I by default.  Use the data step below
   to set Domaint_Type_CD to E.
*/

data &_output ;
  set &_input ;
  /*
   Add code to determine external domains
   Examples - if domain_nm ne 'www.mysite.com' then Domain_Type_CD = 'E';
            - if index(domain_nm,'.mysite.com') = 0 then  Domain_Type_CD = 'E';

  */
run;
%rcSet(&syserr);
%rcSet(&sqlrc);


/**  Step end Reset Domain_Type_CD **/
```

Server: <default>

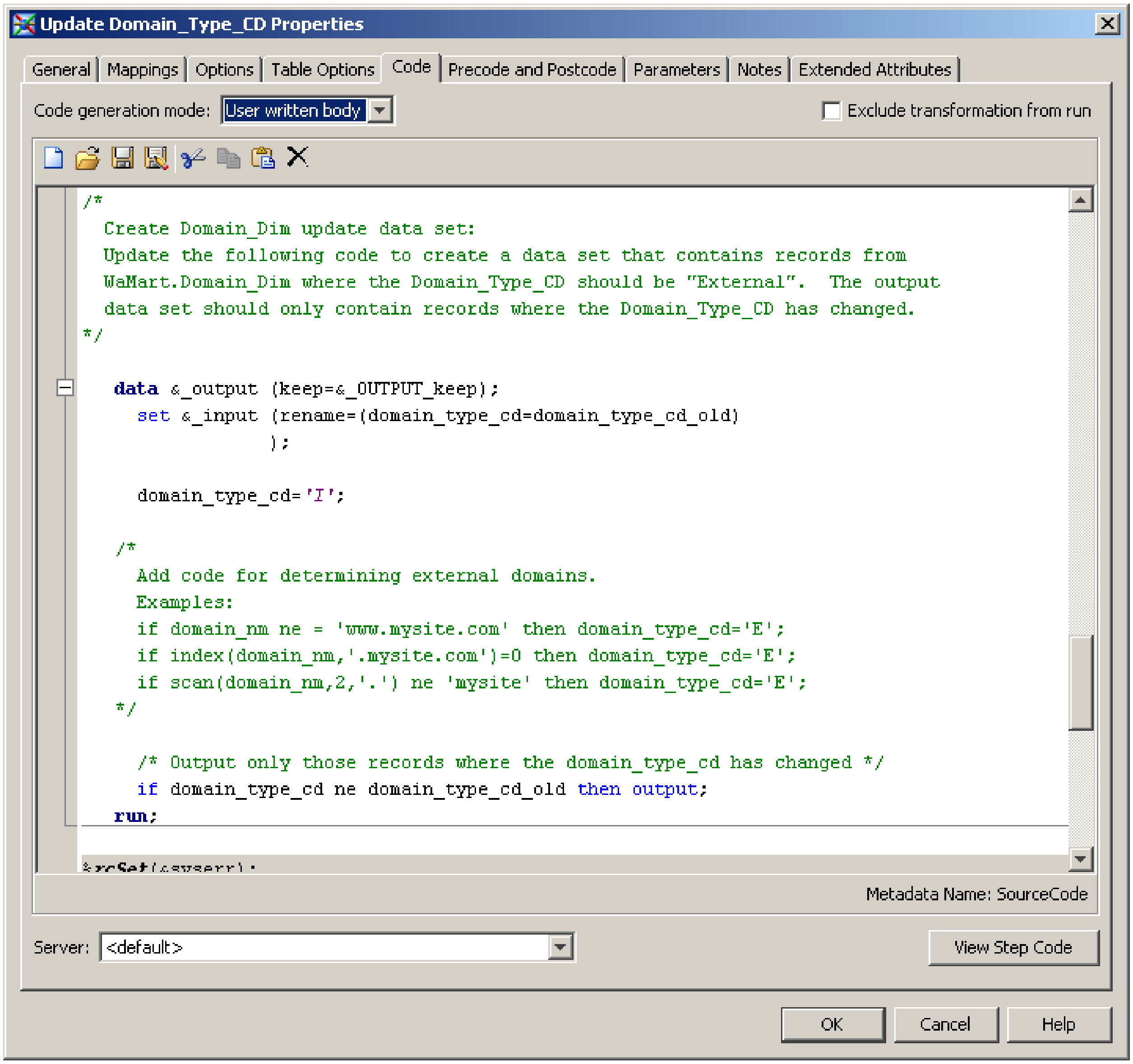
4 Edit the DATA step within the **Code** tab to set the value of Domain_Type_Cd to E (external).

5 After the code has been updated, click **OK** to save the change and close the node.

6 Double-click **node 2** (named Update Domain_Type_Cd), and then click the **Code** tab.

7 Edit the DATA step within the **Code** tab to set the value of Domain_Type_Cd to E. Add the new code below the **Add code to determine the external domains comment** comment block. Click **OK** to save the change and close the node.

8 Save the job.



Code generation mode: User written body

☐ Exclude transformation from run

```
/*
  Create Domain_Dim update data set:
  Update the following code to create a data set that contains records from
  WaMart.Domain_Dim where the Domain_Type_CD should be "External".  The output
  data set should only contain records where the Domain_Type_CD has changed.
*/

   data &_output (keep=&_OUTPUT_keep);
     set &_input (rename=(domain_type_cd=domain_type_cd_old)
                 );

     domain_type_cd='I';

   /*
     Add code for determining external domains.
     Examples:
     if domain_nm ne = 'www.mysite.com' then domain_type_cd='E';
     if index(domain_nm,'.mysite.com')=0 then domain_type_cd='E';
     if scan(domain_nm,2,'.') ne 'mysite' then domain_type_cd='E';
   */

     /* Output only those records where the domain_type_cd has changed */
     if domain_type_cd ne domain_type_cd_old then output;
   run;

&rcSet(&syserr);
```

Metadata Name: SourceCode

Server: <default>

9 Test the job in a non-production ETL.

a. Create the **WaWork.Domain_Dim** table:
   1. Place a Clickstream output data set in the WaSrc library.
   2. Run the Weba_1100_Data_Health_Check job.
   3. Run the Weba_1200_Warehouse_Staging_Tables job.

b. Open the **weba_1300_load_domain_dim** job.
   1. Right-click **node 2** and select **Exclude From Run**.
   2. Run the job.
      Examine the log. To examine the output tables from nodes 1 and 2, right-click the output table object, and select **Open**.

c. Debug as needed.

d. Right-click **node 2** and select **Exclude From Run** to deselect this option.

e. Save the job.

Profile Jobs

What Are Profiles?

Profiles flag and subset visitors' sessions so the sessions can be analyzed for patterns. An example of a profile is a category such as "sessions that are driven to a site by the Google search engine." If you choose this profile, then the path, funnel, or overlay report results will be limited to sessions that match the category; that is, sessions that were sent to the site by Google.

How Are Profiles Used?

There are two types of profiles, visitor and event, which can be used independently or combined. A visitor profile includes information about a Web site's visitors. An event profile includes characteristics about how and when visitors browsed the site. An example of a profile combination might be visitors who are male between the ages of 20 and 30 (visitor component), who had a session between date X and Y (event component 1), and who accessed page Z (event component 2).

Two or more profiles can be defined and combined when there are overlapping matches in the categories. For example, a Males profile can define all male visitors, and a separate Young Adults profile can define all visitors between 20 and 30 years of age. Users of the SAS Web Analytics application can then choose to select data when it matches the Males profile or the Young Adults profile. Users can also select data that matches one profile or the other (all males or 20- to 30-year-olds), or both (all 20- to 30-year-old males).

To maintain profile jobs, get a list of sessions in a category and run a SAS Data Integration Studio transformation on the list to flag sessions in the data mart. Schedule the profile job to execute on a regular basis, and as a result, sessions are flagged as they are appended to the data mart tables.

The ETL contains the sample Profile job Weba_1410_Profile_Search_Engines. For more information, see Appendix 8, "Defining Profiles."

Aggregate Jobs

Creating Aggregate Jobs

The Aggregate jobs create summarized data that is used in many Web Analytics reports. These jobs are numbered from 1500–2150 and 2500.

All the Aggregate jobs use a custom transformation named Aggregate or Weba_Aggregate, which are referred to as the Aggregate transformation in this document. This transformation contains options that can be configured for enhanced performance. It also contains options that can control the columns to be aggregated and the statistics to be derived.

CAUTION:
Removing existing aggregations might result in errors in existing reports.

Example: Creating a Custom Aggregate

1 Open SAS Data Integration Studio and select the folder where a new job would be located.

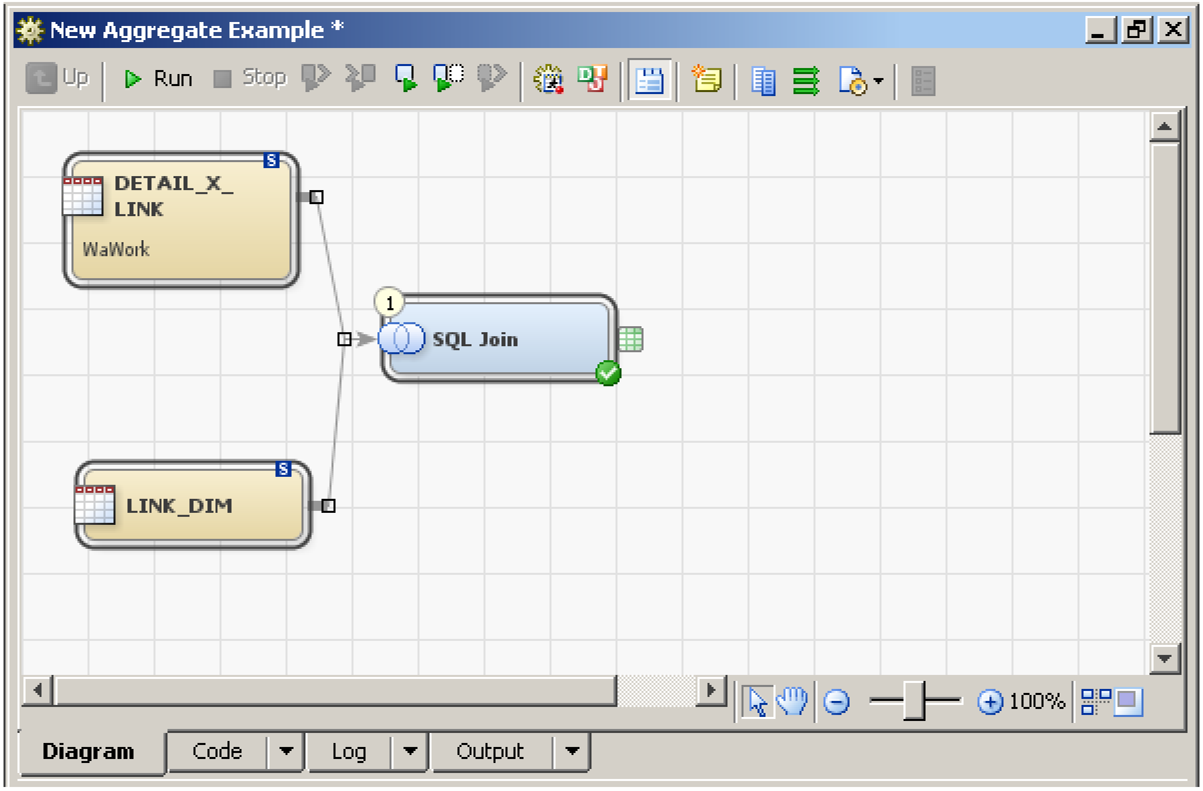
2 Right-click the folder and select **New ▸ Job**. Name the new aggregate job Weba_2510 or higher so that it will run before the Weba_2600 job and after the other aggregate jobs.

3 Drag the SQL Join transformation or other transformation onto the job palette to create the daily summary that uses the following tables:

WaWork versions of Detail_Fact, Session_Fact, Detail_X_Link, or other table

WaMart dimension tables

Here is an example of a join between the Detail_X_Link table and Link_Dim:



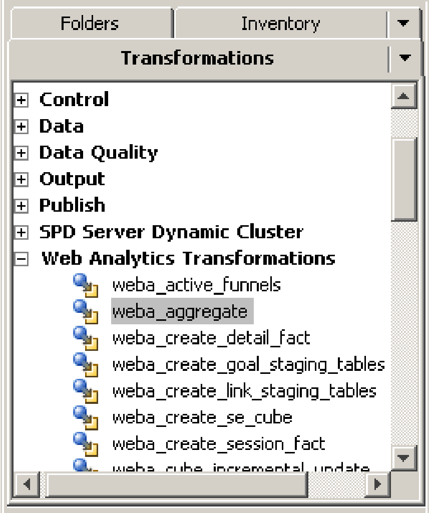
4 The output table from the transformation is a summary data set that is summarized by day. It should contain at least three fields: Num_Clicks, Session_Dt, and Link_Sk. Adjust the options of the SQL Join transformation so that the generated SQL represents the logic of the query:

```
Proc sql;
    Create table (work table) as
    Select link_sk,
           session_dt,
           count(1) as num_clicks
    from wawork.detail_fact a,
           wawork.detail_x_link b
```

```
      where a.detail_sk = b.detail_sk
      group by link_sk, session_dt;
  quit;
```

Save the changes.

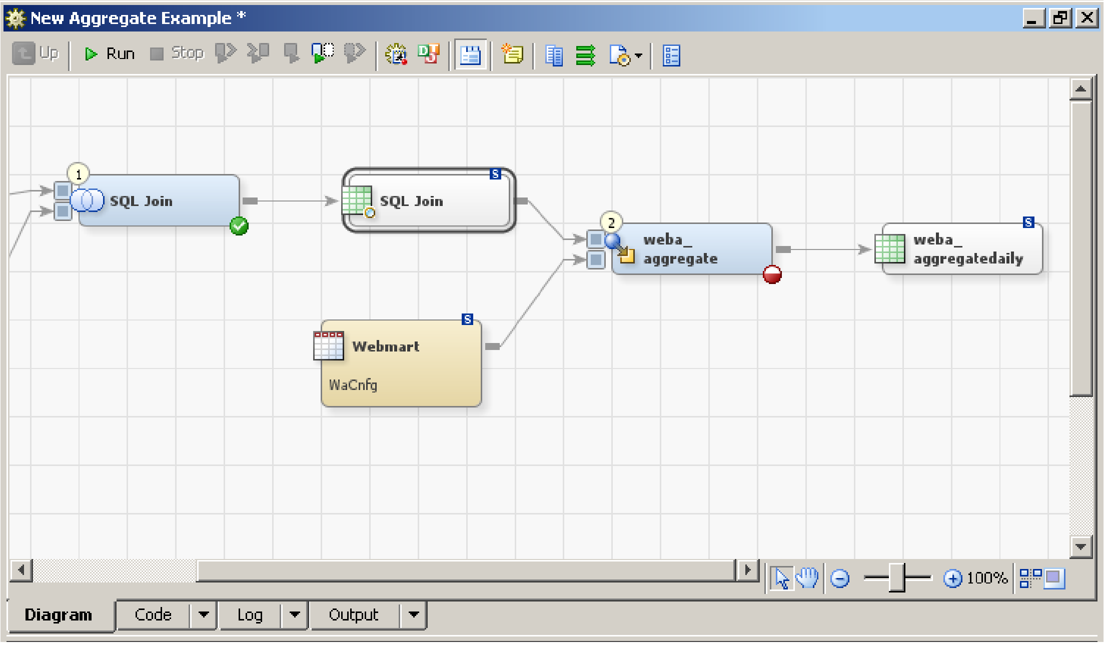
5 Select the **Weba_Aggregate** transformation that is located on the **Transformation** tab under Web Analytics Transformations.



6 Drag the transformation onto the job palette.

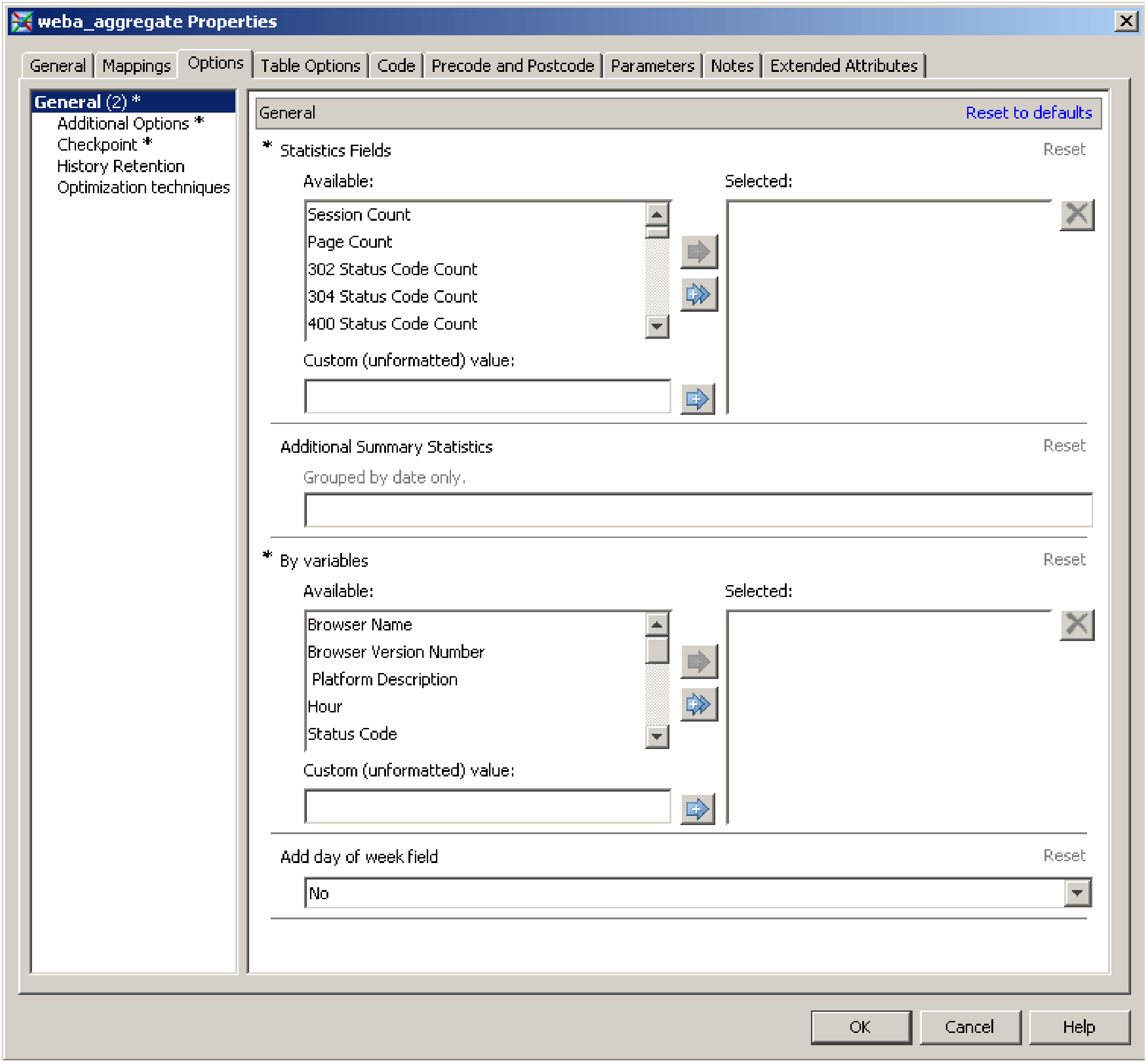
a. Connect the following two inputs:
   - ❑ Output data set from SQL join
   - ❑ WaCnfg.Webmart

b. Create the following five outputs by right-clicking **Weba_Aggregates** and selecting **Ports ▸ Add Output Ports**.
   - ❑ Daily
   - ❑ Weekly
   - ❑ Monthly
   - ❑ Quarterly
   - ❑ Annually



7 Right-click the **Weba_Aggregate** transformation and select **Properties**. Click the **Options** tab.

8 In the Statistics Field area, enter **`num_clicks`** in the **Custom (unformatted) value** field. Click the right arrow to add it to the Selected box. In the **By variables** area, enter **`link_sk`** in **the Custom (unformatted)** field and click the right arrow to add it to the Selected box. Click **OK** to save the changes.



* Statistics Fields

Available:

Session Count
Page Count
302 Status Code Count
304 Status Code Count
400 Status Code Count

Selected:

Custom (unformatted) value:

Additional Summary Statistics

Grouped by date only.

* By variables

Available:

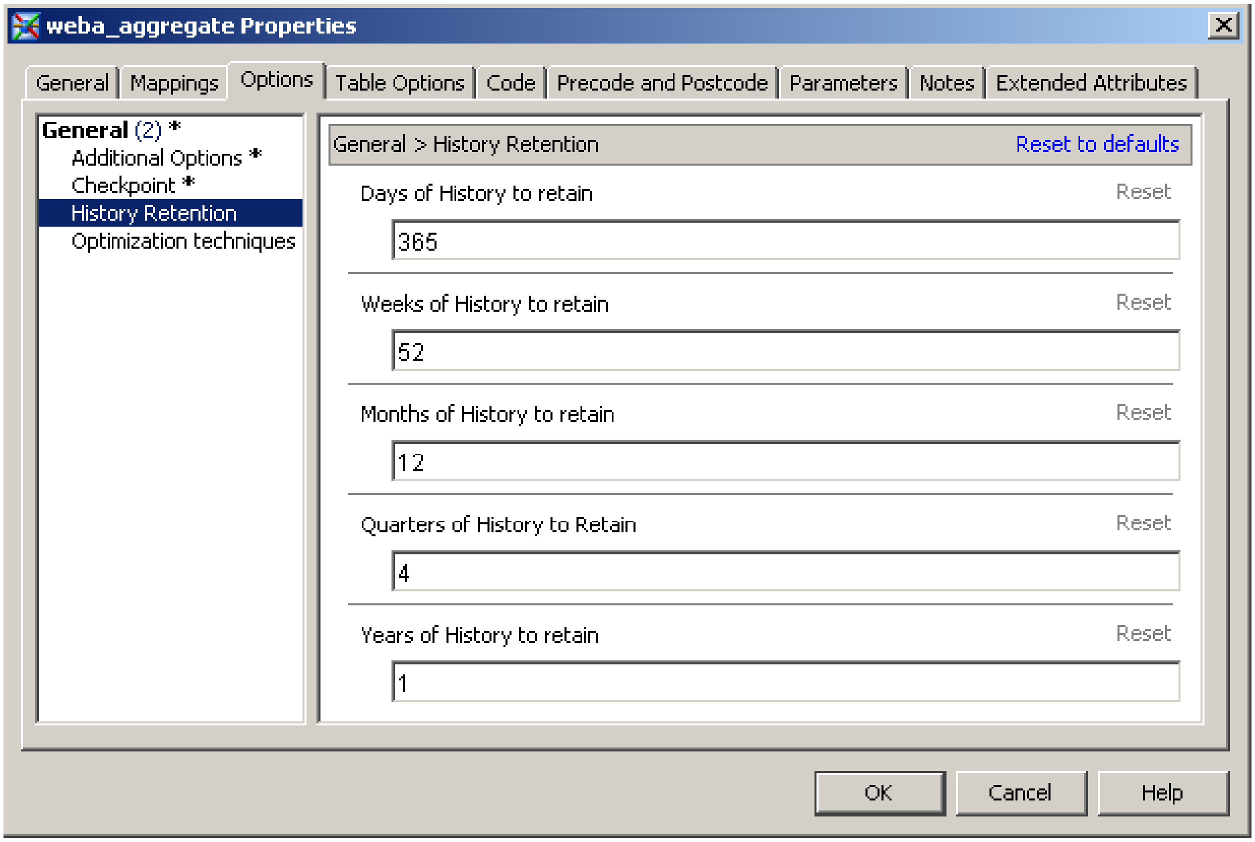
Browser Name
Browser Version Number
Platform Description
Hour
Status Code

Selected:

Custom (unformatted) value:

Add day of week field: No

9 Select the **History Retention** option from the left pane. For each field, specify the amount of history to retain for each output table that pertains to day, week, month, quarter, and year.



Days of History to retain: 365

Weeks of History to retain: 52

Months of History to retain: 12

Quarters of History to Retain: 4

Years of History to retain: 1

10 Test the Aggregate job.

11 Add the Aggregate job to the ETL schedule.

Creating New Fields

The Aggregate transformation uses two input tables:

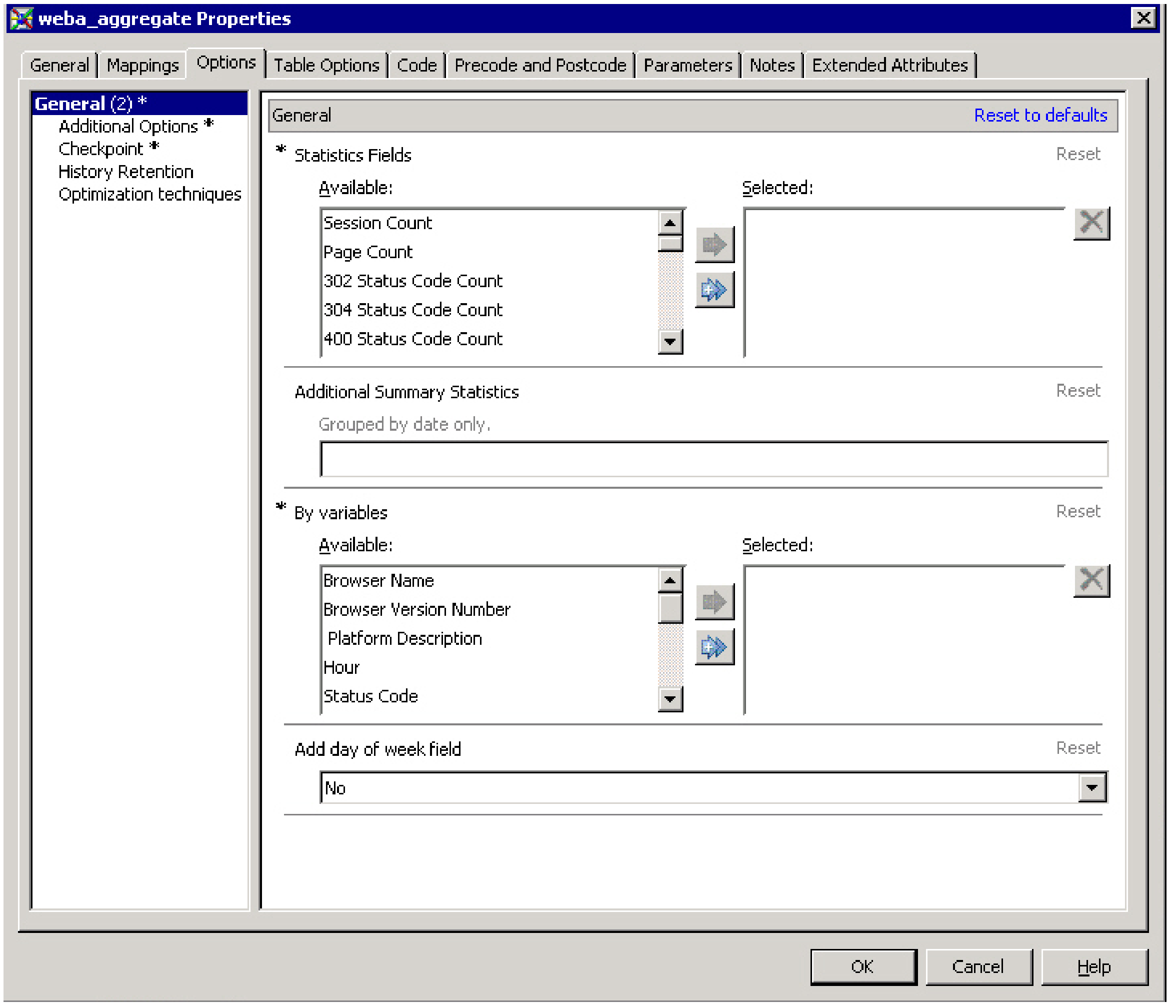
- ❑ Input table 1 contains an iteration of summarized data.
- ❑ Input table 2 references WACNFG.WEBMART to obtain the identifier of the warehouse that is being loaded.

The Aggregate transformation updates and appends five output tables with the data that was provided by input table 1. The five output tables have the same data rolled up by day, week, month, quarter, and year, respectively.

The input tables and the output tables are subject to the following rules:

- ❑ The aggregate's output tables must contain a common set of fields.
- ❑ Aggregation processes are incremental loads and do not backload past history. Therefore, if the options settings are changed in the Aggregate transformation object, you must ensure that the changes do not negatively affect the correctness of the data or incremental loading. For example, if you change a grouping element or add or remove a metric after running aggregate loads for a period of time, then you must add the appropriate column to the upstream aggregate tables and also account for past history.
- ❑ Input and output tables must contain the fields that are referenced in the Aggregate transformation's selected columns in order to function properly.

- When adding output tables, delete the table loader process that is automatically included for each job. The Aggregate transformation re-creates tables dynamically. The extra code from the table loader process is unnecessary and can compromise the accuracy of the process.
- Do not add a date index to the output tables, because a date index is automatically defined for the aggregates. Doing so explicitly in the SAS Data Integration Studio Indexes property of the output tables results in errors in the job.
- Business rules that control the aggregate statistics are generated from the user's selections and entries on the **Options** tab of the Aggregate transformation. To view the options, right-click on the **Aggregate transformation**, select **Properties**, and then click the **Options** tab.



* Statistics Fields

Available: Session Count, Page Count, 302 Status Code Count, 304 Status Code Count, 400 Status Code Count

Selected:

Additional Summary Statistics

Grouped by date only.

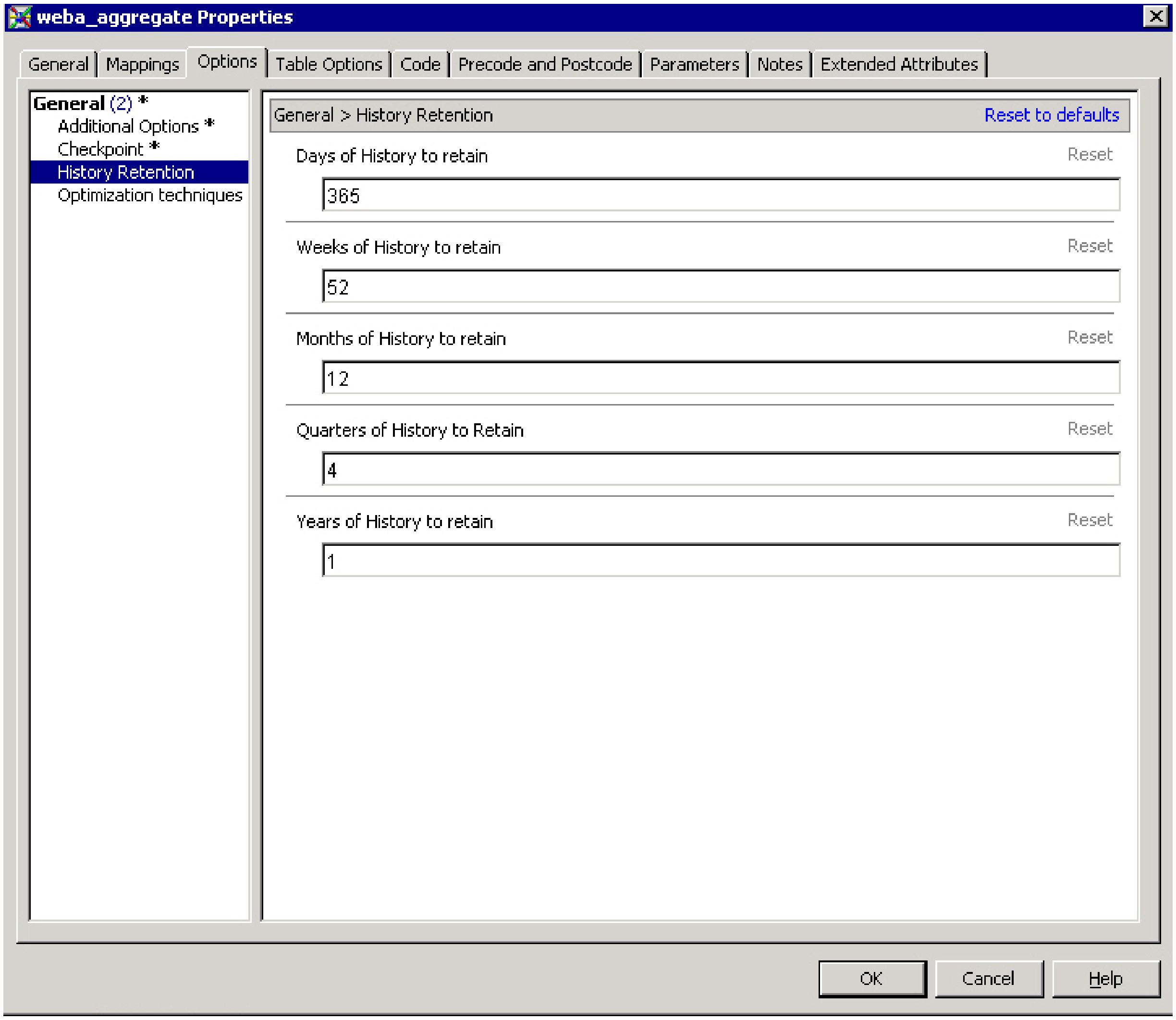
* By variables

Available: Browser Name, Browser Version Number, Platform Description, Hour, Status Code

Selected:

Add day of week field: No

- Click in the panel on the left side of the window to see other options for history retention and optimization techniques.



Days of History to retain: 365

Weeks of History to retain: 52

Months of History to retain: 12

Quarters of History to Retain: 4

Years of History to retain: 1

Optimizing Performance in an Aggregate Job

Several options on the Aggregate transformation are available to improve loading performance. Some of these options are available only when the Web mart is stored in Oracle. Other options are available for Web marts that are stored in both SAS and Oracle.

To modify the performance tuning options:

1. Open Data Integration Studio and navigate to the folder <*Web mart*>`/5.4 Jobs/ETL Jobs`.
2. Open an Aggregate job (identified by _agg_ in the name).
3. Right-click the Aggregate job and select **Properties**.
4. Click the **Optimization Techniques** tab and edit the following options.

Note: Redeploy the job after you make the modifications.

Bulkload? Yes, No, Conditional

is an option available only for Web marts that were stored in Oracle. When set to Yes, this option causes SAS/ACCESS Interface to Oracle to call the Oracle SQL*Loader (SQLLDR). The Oracle bulk loader provides superior load performance, so you can rapidly move data from a SAS file into an Oracle table. Setting this option to No bypasses the Oracle bulk loader. Setting the option to Use Threshold indicates that a value will be specified for the Threshold option, defining a number of records in a staging data set to be appended. Data sets with more than this number of records will be appended using the Oracle bulk loader. Data sets with fewer than this number of records will be appended without using the Oracle bulk loader. The default value is Yes.

Note: The SQL*Loader direct-path load has a number of limitations. For more information, including tips to boost performance, see the Oracle utilities documentation. You can also view the SQL*Loader log file instead of the SAS log for information about the load when you use bulk load. Also see the SAS/ACCESS Interface to Oracle documentation.

When Bulkload is selected as a loading option, either explicitly or conditionally, SAS saves the Oracle bulk load log separately from the SAS log. For information about saving the log in an alternate location that uses the ORA_LOGDIR option, see “Specifying an Alternate Directory for ORA_LOGDIR” in Appendix 4.

Upsert records? Yes, No

enables the aggregate job to use the Oracle UPSERT ability to update and insert records in a single step. This option is available only for Web marts that are stored in Oracle. The default value is No.

Drop and recreate indexes? Yes, No

is an option available for Web marts that were stored in either SAS or Oracle. If indexes are kept during an append, the append process is slower than if the indexes are dropped. However, the lack of indexes and any integrity constraints that use indexes might allow undesired data records to be added to the data file. These undesired data records would violate the specifications of the indexes and constraints. When the indexes and constraints are rebuilt after the append process is complete, errors could result. It is then the administrator's responsibility to restore the data to a previous version, resolve the problem, and rerun the ETL process. The default value is No.

Threshold (for Conditional Bulkload only)

is an option available only for Web marts that were stored in Oracle. This option specifies the minimum number of records a staging data set must have for the Oracle bulk loader to be used, when the Bulkload Records option equals Conditional. The default value is 0.

Preparing Analytic Data

Introduction to Preparing Analytic Data

Several jobs prepare special subsets of data, which are used for analyses in various reports:

- ❑ Weba_1600_Active_Funnels
- ❑ Weba_2200_Create_Path_Data

- Weba_2500_Insight_Monitor1
- Weba_2501_Prep_Insight_Monitor

The Active Funnels job is controlled entirely by the definition of active funnels from the Web application, and has no job-specific options. Options for the other two jobs in this category are documented in "Customizing the Job to Create Path Data."

Customizing the Job to Create Path Data

The Weba_2200_Create_Path_Data job has parameters on the Stat Pathing transformation that control several important characteristics of the data that is used for Path reports. These parameters can be modified to make more or less data available for general path analysis and for path analysis based on a sample subset of path data. Here are the parameters:

- **Number of dates in pathing data**

  sets the number of days' worth of data that will be stored in the WaAnly library in data sets named Path_yyyymmdd. Because only the specified number of days' worth of data is stored, older data is deleted as new data is created in the Web mart. Increasing the job parameter makes a wider range of data available. Decreasing the parameter value reduces the amount of space that is used by the WaAnly.Path_yyyymmdd data sets but makes a narrower range of data available. The default is 30.

  Note: Users are unable to create a Path report in SAS Web Analytics if they request a path analysis on dates that are outside the most recent number of days specified in this parameter. Users can create the report by modifying the dates for analysis so that there is overlap with dates that have available data.

  For example, suppose that a Web mart contains data for 01/01/2010 through 03/31/2010, and the path data parameter was set to its default value of 30 when the data mart was loaded. A Path report request for data from 02/01/2010 through 02/28/2010 would fail with an error because only data from 03/02/2010 through 03/31/2010 would be available. However, if the requested report dates were changed to 02/01/2010 through 03/02/2010, a report would be produced based on one day's worth of data in the WaAnly.path_20100302 data set. Information about the actual dates that were used to produce the report is included in a message in the Web application.

- **Minimum # of obs in Path_yyyymmdd data sets to create Path_Sample**

  sets the number of observations that are required to create or add data to the WaAnly Path_Sample data set. The WaAnly Path_Sample data set is used when the Path report is run on sample data. If the number of detail pages that can be included in the pathing data for this ETL run is less than the number of observations, the job ends with an error. If you receive the following error and expect to continue to have low numbers of valid pages in your incoming data, lower the value of this parameter.

  ```
  ERROR: (WEBA:WA_STAT_PATHING) The PATH_SAMPLE data set cannot be
  created because there are not enough PATH_yyyymmdd obs.
  ```

  ```
  ERROR: (WEBA:WA_STAT_PATHING) To create PATH_SAMPLE either decrease
  the value of the WAB_PATHING_SAMPLE_MIN_OBS parameter or wait until
  there is more data.
  ```

  The default is 1000.

- **Percent Sample Size (10=10%)**

defines the size of the sample selected for the WaAnly.Path_Sample data set. The value represents a percentage of the total number of observations for a single day. The default value is 2%.

- ❑ **Identify sample classes**

  enables you to include the stratification of sessions to be selected for inclusion in the random sample, based on the number of pages viewed within a session (the value of the Valid_Page_Cnt column in the Session_Fact table, or Session_Fact.Valid_Page_Cnt). The default value, "low-high='all'", performs stratification by the session date. The value of the option must be a character string suitable for use in PROC FORMAT. For more information, see the *Base SAS Procedures Guide*.

  This option is used to group the values of Session_Fact.Valid_Page_Cnt. For example, "low-2='low' 3-5='med' 6-high-'high'" groups sessions into three groups based on values of Valid_Page_Cnt. The sample selection will be stratified across these groups for each date that is encountered in the data that is being loaded.

- ❑ **PROC SurveySelect options**

  is any option that is valid in the PROC SURVEYSELECT statement. For more information, see the *SAS/STAT User's Guide*.

To change the values of the parameters:

1. Open SAS Data Integration Studio and navigate to <Web mart>/5. 4 Jobs/ETL Jobs.
2. Click the **weba_2200_create_path_data** job.
3. Right-click the **Stat Pathing** transformation, and then select **Properties**.
4. Click the **Options** tab and update the value of the desired parameter.
5. Save your changes.
6. Redeploy the job.

Customizing KPIs from a New Data Source

This section describes how you can modify the Weba_2501_Prep_Insight_Monitor job to customize KPIs from a new data source.

A new data source can consist of an extension of a fact table or a brand-new source. The Insight Monitor Rollup transformation works by summing up all of the numeric fields that are passed into it by Session_Dt. The input from the Insight Monitor Rollup transformation must consist of a data source with one record per date, and only numeric fields.

To add KPIs, create a table that contains the required statistics and Session_Dt by using the Session_Fact table or other table as a source. It is recommended that you create a new job that can:

1. Create a new table that contains only the KPIs and Session_Dt. The KPIs must be numeric and summed by date.
2. Summarize the new KPIs by Session_Dt and save the results to a permanent data set.

The new job has these attributes:

- ❑ It consists of any input source the user wants to use (preferably one updated at least as often as the warehouse).

- ❑ It consists of new transformations, user-written code, SQL joins, or any combination thereof.

Note: There must be a maximum of one record per date.

The existing job must be modified to merge the new KPIs with the traditional ones. The resulting data set contains one record per date, and the numbers can be summed to roll up the statistics by week or month. Therefore, the Weba_2501_Prep_Insight_Monitor job must be modified because it can handle only one input data set that contains the superset of information. To make the changes:

1. Disconnect the Insight Monitor Metrics_Output1 link from the Insight Monitor Metrics transformation to the Insight Monitor Rollup transformation.
2. Drop the new table with the new KPIs into the job.
3. Join the Insight Monitor Metrics_Output1 and the other data sources (you can use more than one) from step 2. Drag the Weba_Data_Step_Join transformation into the job and use the two tables (Output1 and your new table) as input. Select the options on the transformation as necessary to join the tables by the variable Session_Dt.
4. Ensure that the output from step 3 has only one record per Session_Dt variable.
5. Designate the output of the join as the input to the Insight Monitor Rollup transformation. Be sure to map the new metrics into the transformation's three output data sets.
6. Be sure to map all of the new metrics within the WaAnly.Anly_<*interval*>_Metrics output data sets into the views created by the Extract nodes (nodes 3 - 5).

   CAUTION:

   This step must be performed because Performance Monitor and Insight reports both use the views within the analyses.
7. Update the WaCnfg.Cnfg_Metrics data set by adding the metric name to it. The list of available metrics for the Performance Monitor and Insight reports is controlled by WaCnfg.Cnfg_Metrics. You will need to add a new record to Cnfg_Metrics with the following field information:
8. **locale** - (leave blank)
9. **name** - variable name
10. **label** - descriptive label for metric

Closing the ETL Process

The Weba_2600_ETL_Done job is the final job of the ETL process and performs these tasks:

- ❑ deletes the data set WaCnfg.Cnfg_Signal_File, which identifies the input data sets that are used by the current ETL process.
- ❑ changes the Progress_Check column in the WaCnfg.Cnfg_Input_Hist data set from 1 to 0, indicating that the data set has been loaded successfully.
- ❑ (Optional) deletes the WaSrc data sets that have been loaded. This option is defined in the Weba_Post_ETL_Cleanup transformation of the Weba_2600_ET_Done job. The default is No.

- ❑ for a site running its first ETL process, changes the IsActive attribute of all SAS and Oracle marts from 0 to 1, making the site available for reporting.

Best Practices for Working with the ETL Process

Profile Jobs

The ETL contains an example profile job, Weba_1410_Profile_Search_Engines. Information about creating custom profile jobs is available at `http://support.sas.com/resources/papers/tnote/webanalytics.html`.

Ongoing ETL Processes

How often the ETL job should be run within a 24-hour period depends on the volume of traffic on a Web site and other factors. A typical ETL job processes data from the previous day. However, different intervals might be necessary depending on the needs of the data users.

Consider these factors when you are planning the ETL processing window:

- ❑ If the site has multiple Web servers, or subsidiary domains (for example, **`www.sas.com`** and **`support.sas.com`**), or both, then how much of a time gap can there be between the creation of each log?

  Example 1 – Site A has three servers in its server farm. When should the server logs be cut? A Web master might want to cut them at different times of the day. Should you insist that they be cut at the same time?

  Example 2 – Site B has a server farm with three servers and a site made up of a single domain.
- ❑ How large is the nightly batch window available for Weba ETL? Conversely, when must the data be available in the morning?
- ❑ Is parallel processing of jobs within the ETL possible? All jobs that have the same job number can be run concurrently.

Note: There could be an adverse impact on individual job performance, because all the WaWork and WaMart data sets are on the same respective mount points by default. Each site should assess this capability against the backdrop of its hardware configuration.

Back Loading Detail Data

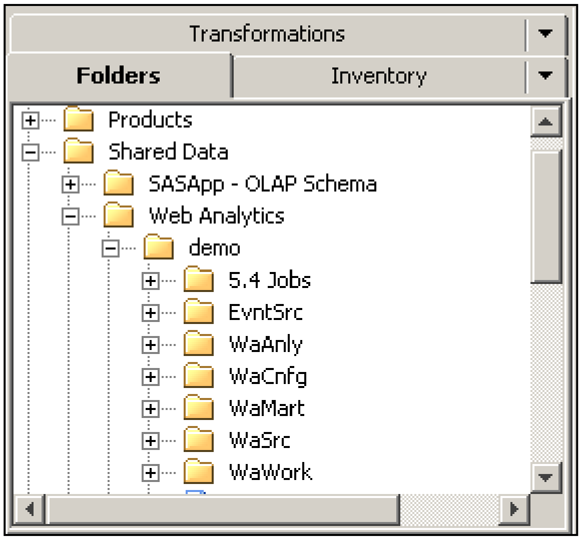
If you have a repository of detail (that is, Clickstream) data, then do not simply concatenate the data sets for individual dates into a single ETL input data set. Doing so creates problems when visits span the individual source data sets. Such sessions will contain records showing that the session is both opened and closed, which causes the Weba_1100_Data_Health_Check job to fail. Instead, place all of the individual data sets into the WaSrc directory, use the Weba_Run_ETL.sas script with CREATE_WASRC_DETAIL=Y and RUN_STYLE=C, or create a customized concatenation job that will exclude the records during open visits (for example, Session_Closed eq `0`). These records would have been excluded by the Web Analytics ETL, so their absence does not affect the accuracy of the data.

Assigning a Libref

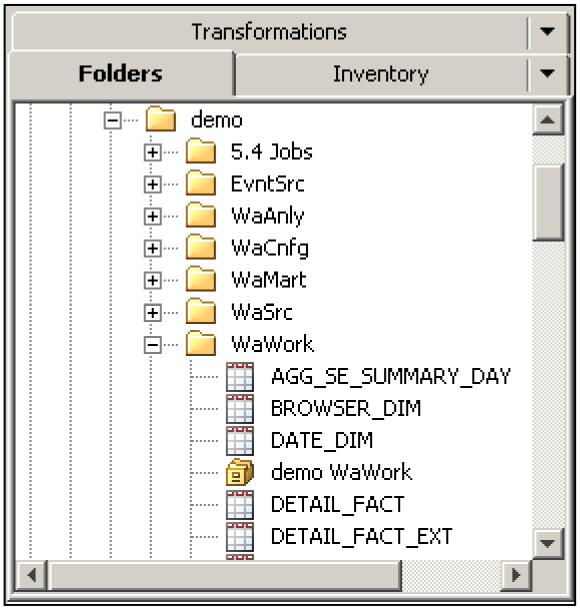
By default, the site initialization job assigns a site's library locations. The default library locations are a set of subdirectories under the root operating system directory specified within the job. Especially for large sites, it may be desirable to move one or more of the libraries to a different drive to reduce I/O competition. The WaWork library is a natural candidate for this move.

To change a library location for a site:

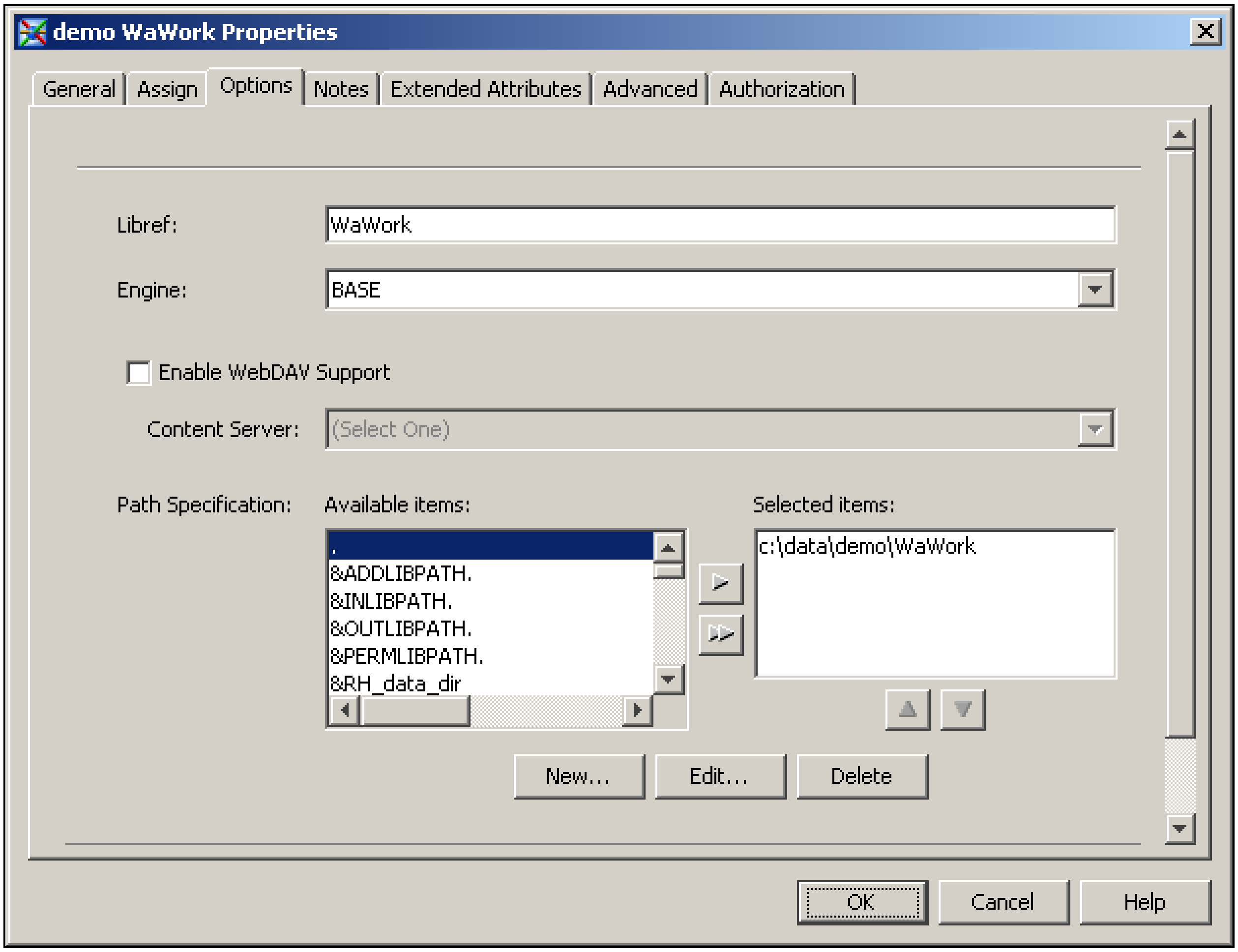
1. Open SAS Data Integration Studio.
2. Navigate to the metadata folder for a site.
3. Expand the folder. Six subfolders appear. The subfolders that contain the library objects are **WaAnly**, **WaCnfg**, **WaMart**, **WaSrc**, and **WaWork**.



4. Expand the folder that contains a library object.



5 Double-click the library object to open the Properties window and click the **Options** tab.



demo WaWork Properties

Libref: WaWork

Engine: BASE

☐ Enable WebDAV Support

Content Server: (Select One)

Path Specification:

Available items:
.
&ADDLIBPATH.
&INLIBPATH.
&OUTLIBPATH.
&PERMLIBPATH.
&RH_data_dir

Selected items:
c:\data\demo\WaWork

6 For the **Path Specification** option, deselect the current storage location in the Selected Items box. Select a new location from the Available items list (note that you might need to add a new one by clicking the **New** button), and move it to the Selected Items box. Click **OK** to save your change. For more information about updating SAS library options, see the SAS Data Integration Studio Help.

For information about using an Oracle library, see the SAS Data Integration Studio Help.

7 Be sure to redeploy all jobs after changing the location of any library.

Extending the Web Data Mart

Overview

SAS Web Analytics offers data model flexibility by enabling you to supplement fact and dimension tables with additional data.

Web mart tables contain a predefined set of fields. To extend the web mart to contain additional fields for any table, create an extension table for fact tables, or add the fields directly to dimension tables. Complete the following steps before modifying any web mart tables or ETL jobs.

1 Determine what metrics or attribute fields need to be added and where they will come from. Often, additional fields come directly from the Web or tagging server log. However, there could be instances where the additional field would come from another data source.

2 Define any business logic that is required to implement the additional fields.

3 Determine to which web mart fact extension table (Detail_Fact_Ext or Session_Fact_Ext) or dimension table (*dimension*_Dim) the field should be added. Note that only metric data should be added to the fact extension tables. Attribute data (descriptive information) should be added to the appropriate dimension table. Do not attempt to add columns existing in the supplied dimension tables to a fact extension table, as errors will result.

4 If necessary, modify the clickstream jobs to capture the new fields. Or, create a new job that will update the output of the Clickstream jobs to contain the new fields. Any new fields must be available when the Weba_1200_Warehouse_Staging_Tables job is run. After these planning steps are completed, follow the instructions below to create or update the appropriate tables and modify the ETL jobs to populate the new fields.

General Steps for Adding New Fields

Step 1 - Update the Metadata for the Table

Update the metadata for the table identified in the planning steps.

1 In SAS Data Integration Studio, select the appropriate <*Web mart*>**/WaMart** folder.

2 Right-click the appropriate table and select **Properties.** Select the **Columns** tab and add the new columns.

Step 2 - Create the Physical Table

If you are using SAS as the data repository, you can create the new *Fact*_Ext table or modified dimension table by running a SAS program in an interactive SAS Session, batch job, or SAS Enterprise Guide.

If an Oracle database is used, then check with a database administrator about Oracle table creation practices.

In either case, be sure to define all of the existing columns listed in the table metadata with the correct names and attributes, as well as the new columns.

Newly created extension tables should have their surrogate key (detail_sk or session_sk) defined as the primary key for the table. Any dimension table which is modified must retain all keys defined in the original dimension table, typically only the primary key. The source code for creating the default dimension tables and their keys can be found in Weba_Ddl_Dds.Sas and Weba_Ddl_Indexes (SAS repository) and in Weba_Oracle_Ddl_Dds.Sas (Oracle repository) in the directory:

Windows	**!sasroot\SASFoundation\9.2\weba\ucmacros**
UNIX	**!sasroot/SASFoundation/9.2/ucmacros/weba**

These programs can be used as a starting point to create the modified dimension tables or new Fact_*Ext* tables.

Step 3 – Modify the Jobs to Load the Table

When extending a dimension table, only the load job for that table will need to be modified (Weba_1300_Load_*tablename*). When extending a fact table, the Weba_1200_Warehouse_Staging_Tables job will also need to be modified.

Because the Fact_*Ext* tables are not loaded as part of the original Web mart, you will need to modify and deploy one of the sample jobs for loading an extension table. The sample jobs are located under the <*Web mart*> folder in the Optional Jobs folder. The sample jobs perform the following tasks:

1. Data prep. The sample jobs use the appropriate SAS Data Integration Studio transformation to create all metrics to be loaded into the Session_Fact_Ext or Detail_Fact_Ext table.
2. Data load. The sample jobs use either the Table Loader or the weba_warehouse_table_loader transformation that was generated by SAS Web Analytics to load the WaMart.Session_Fact_Ext or WaMart.Detail_Fact_Ext table.

Examples

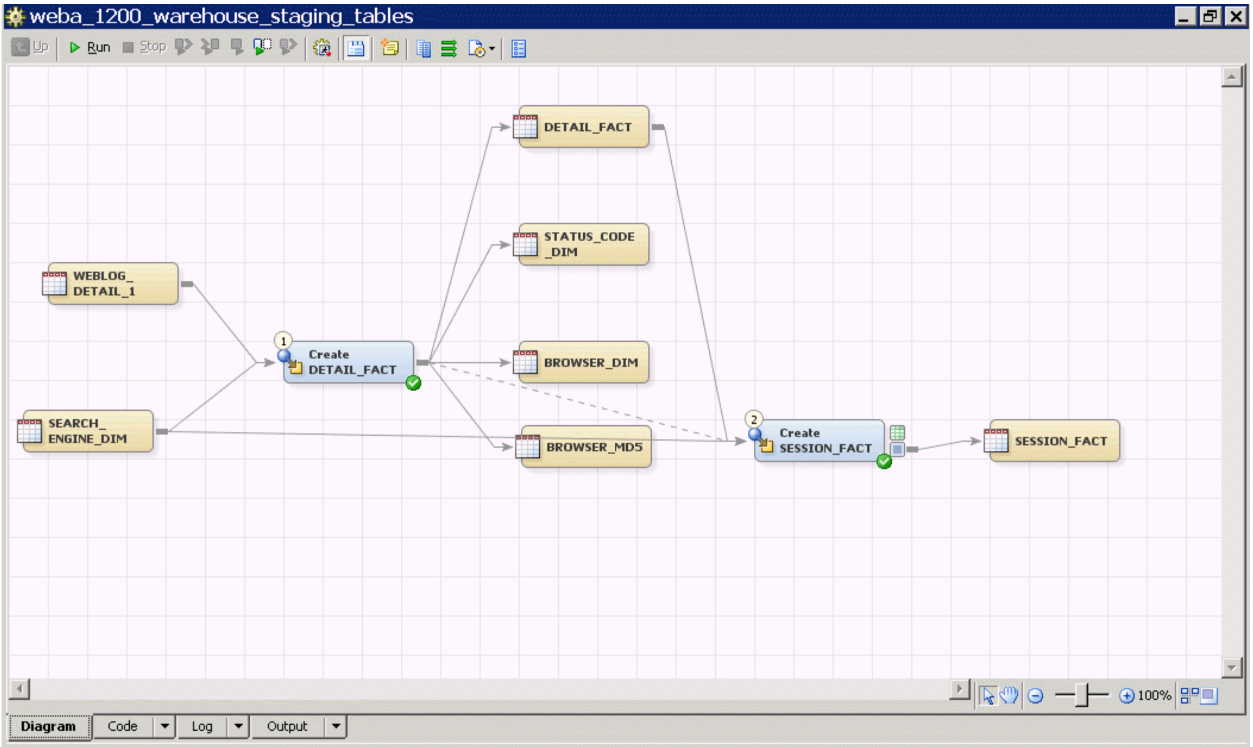
Example1: Create a Detail_Fact_Ext Table

To create a Detail_Fact_Ext table, you must modify a load job. Before you modify the load jobs, complete steps 1 and 2 under “General Steps for Adding New Fields” on page 68 to update the WaMart.Detail_Fact_Ext metadata table and create the physical table.

Modify the Weba_1200_Warehouse_Staging_Tables Job

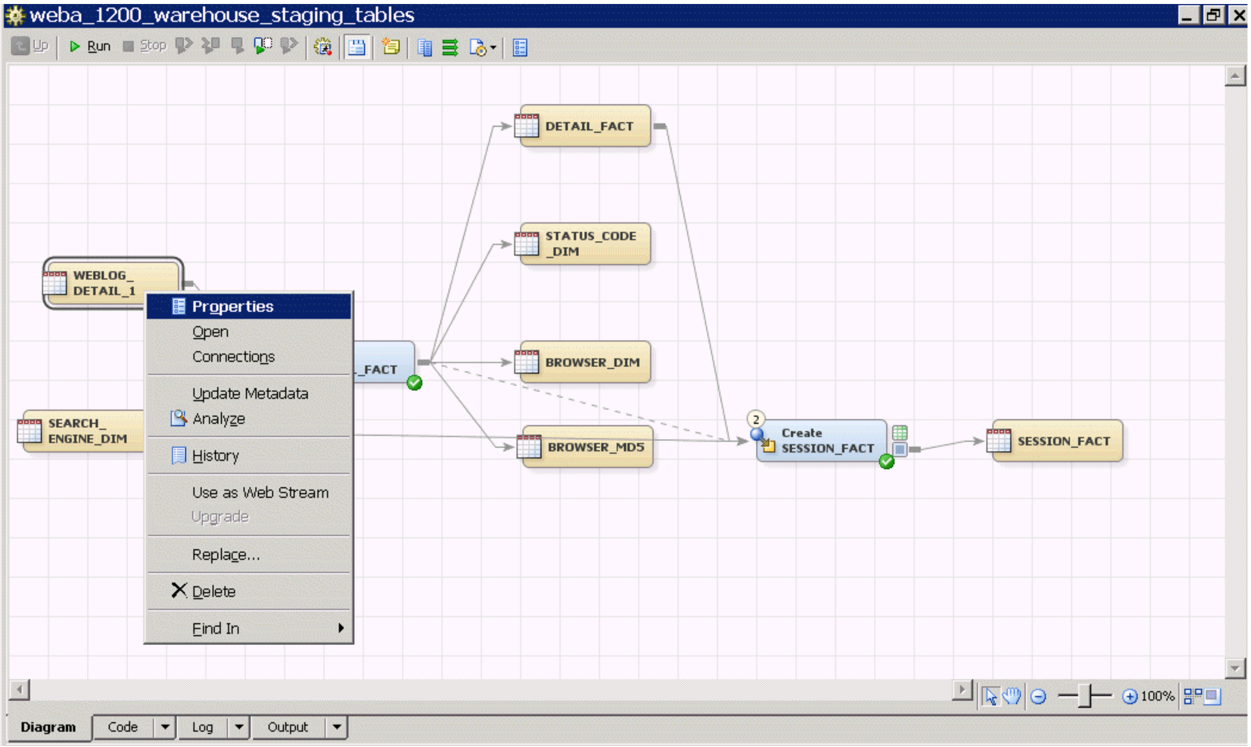
To add variables to the WaWork.Detail_Fact Table:

1. In SAS Data Integration Studio, from the <*Web mart*>`/5.4 Jobs/ETL Jobs` location, open the **weba_1200_warehouse_staging_tables** job in the Web mart in which you want to create an extension table.

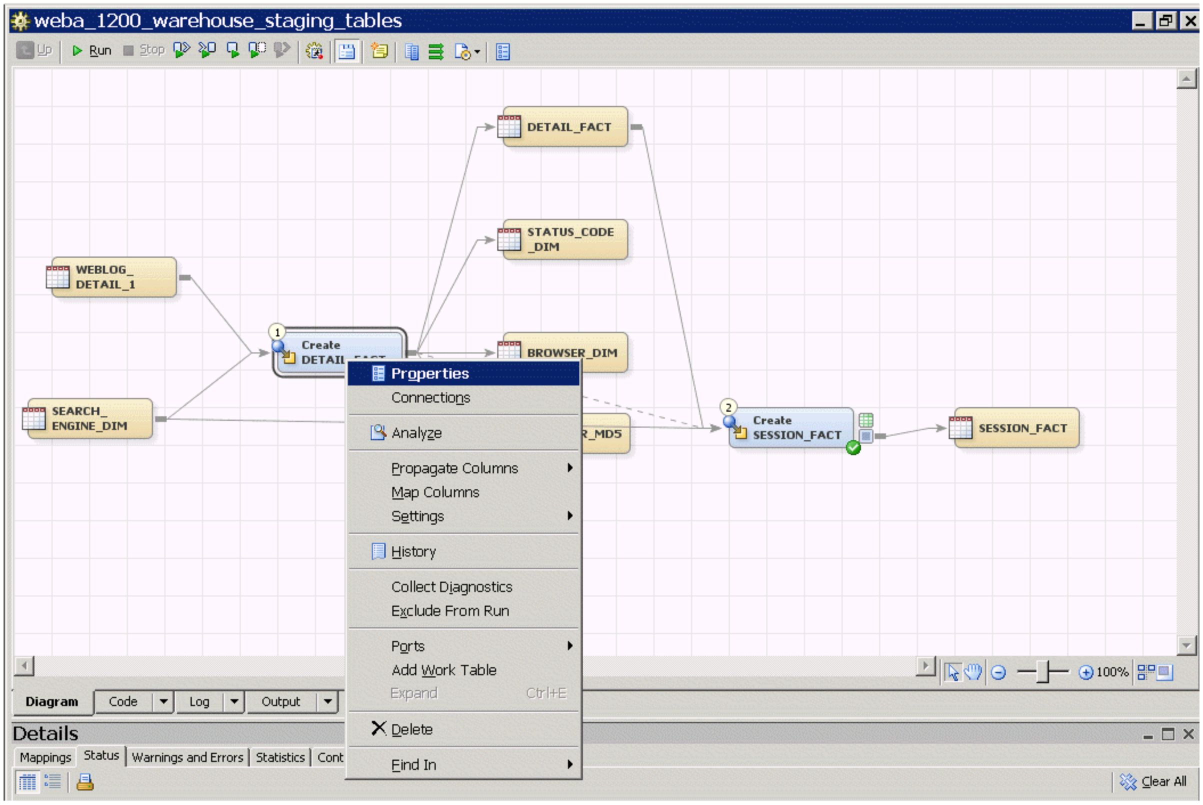


2 Add a preprocessing step for the Weblog_Detail_1 table, if necessary.

3 Right-click the Weblog_Detail_1 table and select **Update Metadata.** This selection updates the columns within the table by adding the custom fields to the table metadata that is within the actual data set. Use the Properties window to enter additional columns.



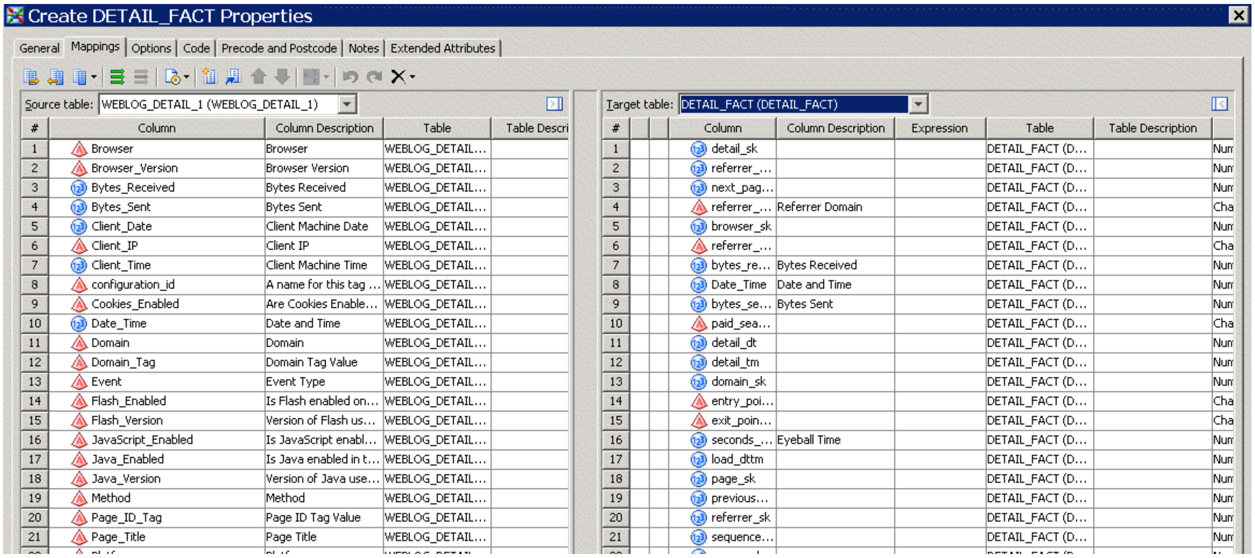
4 Double-click **Create Detail Fact** to open the Properties window.



5 Click the **Mappings** tab.

6 Select Source Table - Weblog_Detail_1 and Select Target Table - Detail_Fact.

Note: Columns are not mapped.



Source table: WEBLOG_DETAIL_1 (WEBLOG_DETAIL_1)

#	Column	Column Description	Table	Table Descri
1	Browser	Browser	WEBLOG_DETAIL...	
2	Browser_Version	Browser Version	WEBLOG_DETAIL...	
3	Bytes_Received	Bytes Received	WEBLOG_DETAIL...	
4	Bytes_Sent	Bytes Sent	WEBLOG_DETAIL...	
5	Client_Date	Client Machine Date	WEBLOG_DETAIL...	
6	Client_IP	Client IP	WEBLOG_DETAIL...	
7	Client_Time	Client Machine Time	WEBLOG_DETAIL...	
8	configuration_id	A name for this tag ...	WEBLOG_DETAIL...	
9	Cookies_Enabled	Are Cookies Enable...	WEBLOG_DETAIL...	
10	Date_Time	Date and Time	WEBLOG_DETAIL...	
11	Domain	Domain	WEBLOG_DETAIL...	
12	Domain_Tag	Domain Tag Value	WEBLOG_DETAIL...	
13	Event	Event Type	WEBLOG_DETAIL...	
14	Flash_Enabled	Is Flash enabled on...	WEBLOG_DETAIL...	
15	Flash_Version	Version of Flash us...	WEBLOG_DETAIL...	
16	JavaScript_Enabled	Is JavaScript enabl...	WEBLOG_DETAIL...	
17	Java_Enabled	Is Java enabled in t...	WEBLOG_DETAIL...	
18	Java_Version	Version of Java use...	WEBLOG_DETAIL...	
19	Method	Method	WEBLOG_DETAIL...	
20	Page_ID_Tag	Page ID Tag Value	WEBLOG_DETAIL...	
21	Page_Title	Page Title	WEBLOG_DETAIL...	

Target table: DETAIL_FACT (DETAIL_FACT)

#			Column	Column Description	Expression	Table	Table Description	
1			detail_sk			DETAIL_FACT (D...		Num
2			referrer_...			DETAIL_FACT (D...		Num
3			next_pag...			DETAIL_FACT (D...		Num
4			referrer_...	Referrer Domain		DETAIL_FACT (D...		Cha
5			browser_sk			DETAIL_FACT (D...		Num
6			referrer_...			DETAIL_FACT (D...		Cha
7			bytes_re...	Bytes Received		DETAIL_FACT (D...		Num
8			Date_Time	Date and Time		DETAIL_FACT (D...		Num
9			bytes_se...	Bytes Sent		DETAIL_FACT (D...		Num
10			paid_sea...			DETAIL_FACT (D...		Cha
11			detail_dt			DETAIL_FACT (D...		Num
12			detail_tm			DETAIL_FACT (D...		Num
13			domain_sk			DETAIL_FACT (D...		Num
14			entry_poi...			DETAIL_FACT (D...		Cha
15			exit_poin...			DETAIL_FACT (D...		Cha
16			seconds_...	Eyeball Time		DETAIL_FACT (D...		Num
17			load_dttm			DETAIL_FACT (D...		Num
18			page_sk			DETAIL_FACT (D...		Num
19			previous...			DETAIL_FACT (D...		Num
20			referrer_sk			DETAIL_FACT (D...		Num
21			sequence...			DETAIL_FACT (D...		Num

7 On the left side, right-click **Column** to add to the Detail_Fact table. Select **Propagate ▸ Select Source Columns ▸ To Targets**.

CAUTION:

Do not map the following fields from the Weblog_Detail_1 table to Detail_Fact table. These fields are renamed during the processing of the Weba_Create_Detail node for loading into the staged versions of various dimension tables.

Browser_Version
Bytes_Received
Bytes_Sent
Domain
Client_Ip
Platform
Query_String
Referrer
Referrer_Domain
Server
User_Agent
UserName
Int_Search_Term
Client_Date
Client_Time
Collection_Id
Cookies_Enabled
Flash_Enabled
Flash_Version
Java_Enabled
Javascript_Enabled
Java_Version
Screen_Size
User_Language

8 Verify that the new column has been added to the Detail_Fact table by checking the mapping arrow from the Weblog_Detail_1 to the Detail_Fact table (not shown).

9 Click **OK** to save the selections in the transformation.

10 Save the job.

11 Run the job in SAS Data Integration Studio to verify that the job runs without problems and redeploy the job through the scheduler.

Modify the Sample Job to Load the Detail_Fact_Ext Table

The table loading must occur after the Weba_1320_Load_Detail_Fact job has been run. To load the Detail_Fact_Ext table:

1 Under the <*Web mart*>**`/5.4 Jobs/Optional Jobs`** folder, select **sample job weba_1330 for loading detail_fact_ext**. This job is a template for creating a job that will load the Detail_Fact_Ext table.

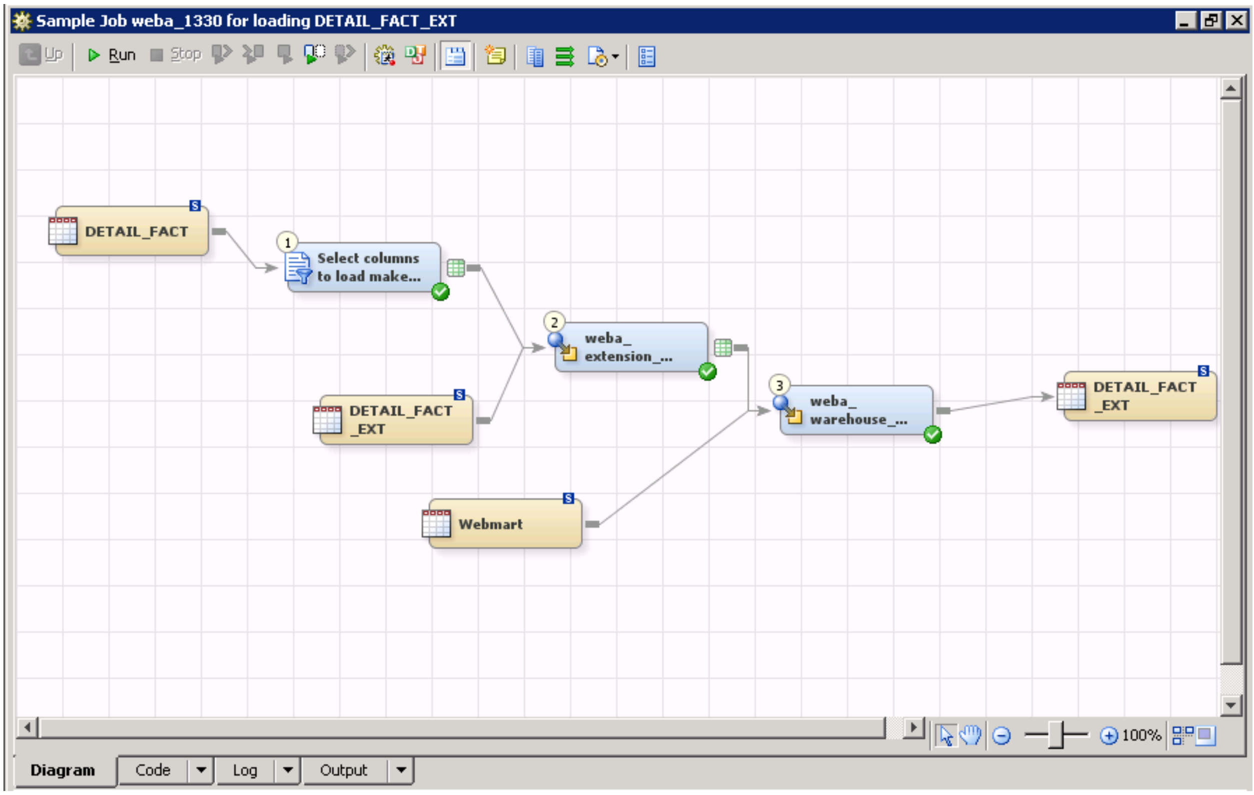
Recommendations:

- Make a copy of this template job. Name the job using a number greater than 1320 in the name, to indicate where the job should run within the ETL process (e.g. Weba_1330_Load_Detail_Fact_Ext).
- Move the copy under the **`5.4 Jobs/ETL Jobs`** folder.
- This job should run within the ETL after the Weba_1320_Load_Detail_Fact job is run.

2 In the SQL Join transformation, select the columns from the WaWork.Detail_Fact table that should be loaded into WaMart.Detail_Fact_Ext.

3 Right-click **weba_warehouse_table_loader** and select **Properties**. Click the **Mapping** tab and map the columns from the loading table to the columns within Detail_Fact_Ext.

4 Save the job and deploy it for scheduling.



Example 2: Create a Session_Fact_Ext Table

To create a Session Fact_Ext table, you must modify a load job. Before you modify the load jobs, complete steps 1 and 2 under "General Steps for Adding New Fields" on page 68 to update the WaMart.Session_Fact_Ext metadata table and create the physical table.

Modify the Weba_1200_Warehouse_Staging_Tables Job

For a description about how to make modifications, see the section on the Detail_Fact_Ext table.

Modify the Sample Job to Load the Session_Fact_Ext Table

The table loading must occur after the Weba_1320_Load_Detail_Fact job has been run. To load the Session_Fact_Ext table:

1 Under the <*Web mart*>`/5.4 Jobs/Optional Jobs` folder, select **sample job weba_1335 for loading session_fact_ext**. This job is a template for creating a job that will load the Session_Fact_Ext table.

  Recommendations:

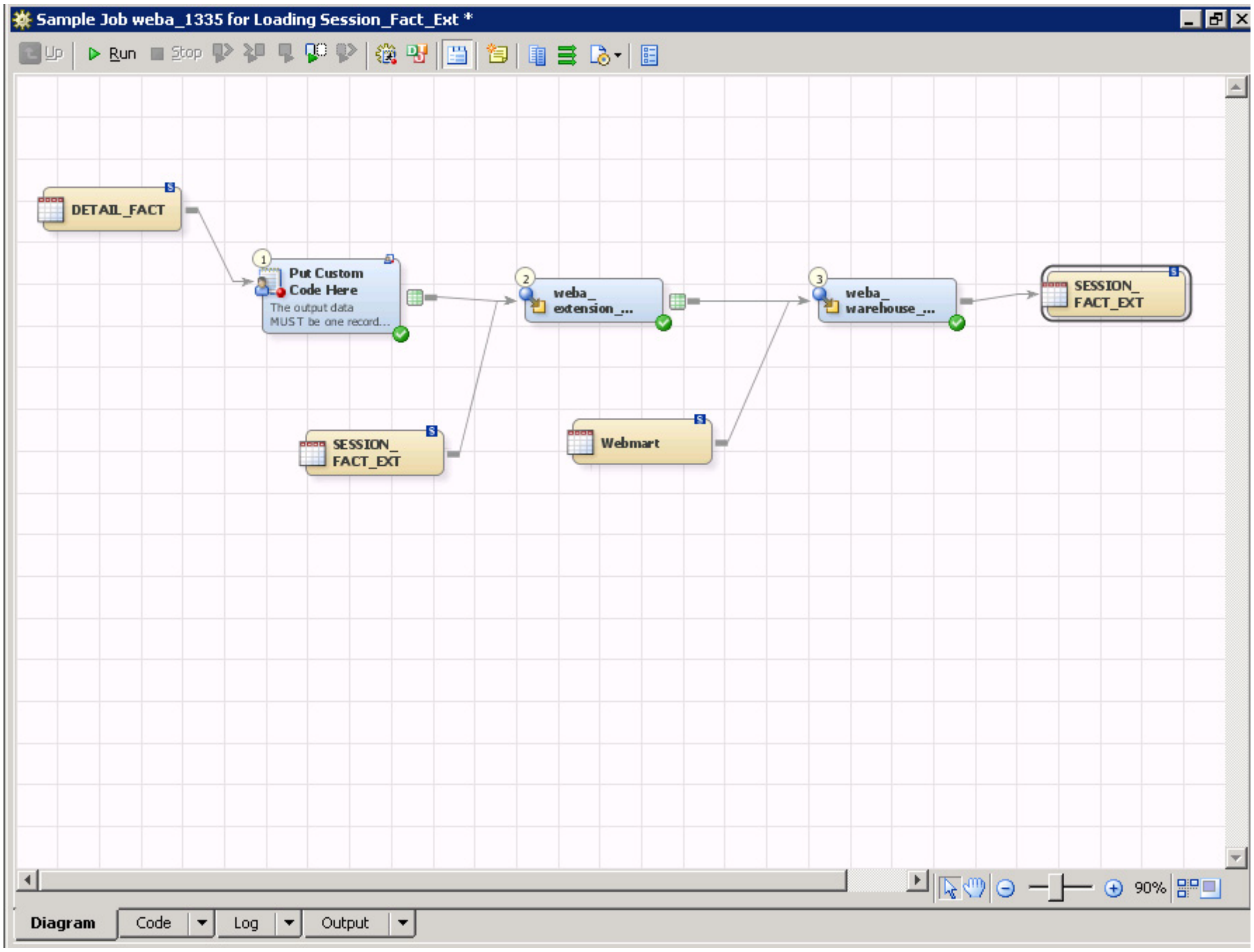
  - Make a copy of this template. Name the job using a number greater than 1320 to indicate where the job should run in the ETL process (for example, Weba_1335_Load_Session_Fact_Ext).
  - Move the copy under the `5.4 Jobs/ETL Jobs` folder.
  - This job should run within the ETL after the Weba_1320_Load_Detail_Fact job is run.

2 In the SQL Join transformation, select the columns from the WaWork.Session_Fact table that should be loaded into WaMart.Session_Fact_Ext.

3 Right-click **weba_warehouse_table_loader** and select **Properties**. Click the **Mapping** tab and map the columns from the loading table to the columns within Detail_Fact_Ext.

4 Save the job.

5 Run the job in SAS Data Integration Studio to verify that the job runs without problems and redeploy the job through the scheduler.



Example 3: Add Fields to a Dimension Table

Each dimension table load job in the Weba_1300_Load job series appends a WaWork table to a dimension table. It also (optionally) updates records in the dimension table with the records that are supplied in a separate update table.

To add fields to a dimension table, you must modify a load job. Before you modify the load jobs, complete steps 1 and 2 under “General Steps for Adding New Fields” on page 68 to update the WaMart.*Tablename*_Dim Metadata table and create the physical table.

1 In SAS Data Integration Studio, from the <*Web mart*>**/5.4 Jobs/ETL Jobs** location, open the **weba_1300_load_*tablename*_dim** job in the Web mart in which you want to create an extension table.

2 Modify the **WaWork** data set in the dimension table load job to ensure that it contains the new required fields. The **WaWork** data set is the input to the **weba_warehouse_table_loader** transformation.

Note: To join the work table with a third-party data source, you will need to add steps to the dimension table load job. For information, see the SAS Data Integration Studio: User's Guide.

3 Right-click the **weba_warehouse_table_loader** or Table Loader transformation and select **Properties**. Click the **Mapping** tab and map the columns from the Work table to the columns on the dimension table.

4 Save the job.

5 Run the job in SAS Data Integration Studio to verify that the job runs without problems and redeploy the job through the scheduler.

The Weba_Warehouse_Table_Loader transformation enables you to update one field on a target table. If you need to update a field in an existing row, complete these steps in SAS Data Integration Studio:

1 Add an input data set to the **weba_warehouse_table_loader** transformation. The input data set should contain a key field and the field to update. All keys that are supplied in this data set need to exist in the target table with the same names and data characteristics.

2 On the **Options** tab for **weba_warehouse_table_loader**, enter values for **Key Field** and **Field to update**.

3 Save the job.

4 Run the job in SAS Data Integration Studio to verify that the job runs without problems and redeploy the job through the scheduler.

Optional Jobs

The following is a list of optional jobs and where you can find more information about them in this document.

Table 3.3: Optional Jobs List

Sample job Weba_1330 for loading Detail_Fact_Ext	See “Example1: Create a Detail_Fact_Ext Table” on page 69.
Sample job Weba_1335 for loading Session_Fact_Ext	See “Example 2: Create a Session_Fact_Ext Table” on page 74.
Weba_1101_Load_Search_Engine_Dim	See the topic, “Adding Search Engines to an Existing Search_Engine_Dim Table” in Appendix 4.
Weba_2550_MA_Response_History_Update	See the topic, “Configure SAS Web Analytics to Monitor Campaign Goals” in Chapter 7.
Weba_9999_Maxmind_Load	See the topic, “Loading Maxmind Data” in Appendix 4.
Weba_Reset_ETL	See “ETL Job Failures” on page 77.

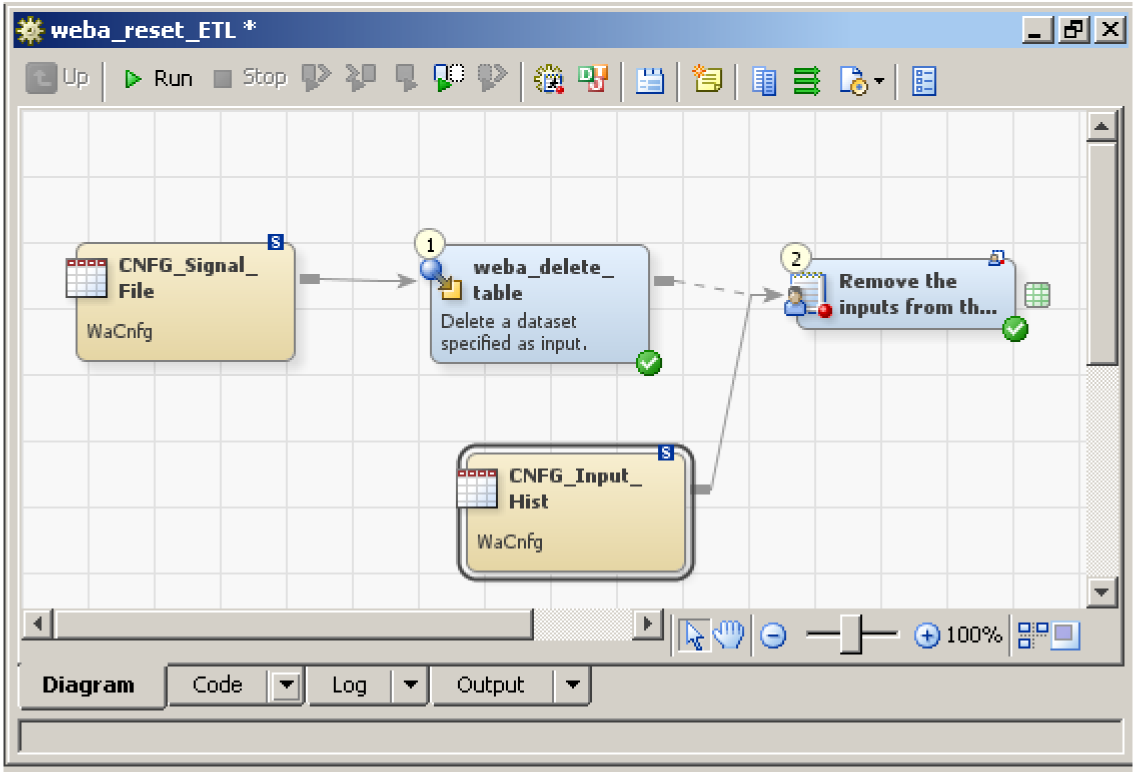
Troubleshooting the ETL Jobs

ETL Job Failures

In the case of an ETL job failure, diagnose the cause carefully. Most of the time job failures are related to either a bad input file or the result of a disk filling up. Correct the cause of the error, and then restart the ETL process, starting with the job that failed.

Note: A known cause for ETL failure is when a libref is changed in metadata but the Data Integration Studio jobs are not redeployed. If the libref for a library is changed, all of the ETL jobs need to be redeployed in order to pick up the change.

To troubleshoot, read the SAS log, find the first error message, and correct it. Restore data as necessary, and repeat the process. If this is not possible, or if the warehouse has been partially loaded, then the administrator will have to roll back to a valid warehouse. In that case, run the optional job Weba_Reset_ETL, and then restart the ETL process from the beginning.



Error Messages in ETL Jobs

These are the types of error messages that are issued when running ETL jobs and where you can find more information about the error messages:

- SAS error messages found in the SAS log – see the *SAS 9.2: Companion for Windows*, “Overview of SAS Error Messages.”
- Data Integration Studio error messages, displayed when submitting jobs interactively from within SAS Data Integration Studio – see the SAS *Data Integration Studio 4.2: User’s Guide,* “Diagnosing and Correcting an Unsuccessful Job.”
- Platform Load Sharing Facility (LSF) job scheduler error messages for scheduled batch jobs - See *Scheduling in SAS 9.2*, “Resolving Issues Related to Scheduling with Platform Suite for SAS.”

WaCnfg.Webmart Table Does Not Exist

If the WaCnfg.Webmart table does not exist, then complete these steps to run a SAS program to create it:

1. In SAS Management Console, expand the **Environment Management**, Data **Library Manager**, and **Libraries** tabs. Locate the library object named WEBA. Open **Properties** for the object, and then click the **Options** tab.
2. Use the value in the **Path Specification: Selected items** box for the value of *path* in the following program. Use the name of your mart for the value of

your-mart-name, and match the case to that used when the mart was created.

3 Submit the program in an interactive SAS session. The WaCnfg.Webmart data set that is created should have one observation and two variables. If it does not, contact your SAS Web Analytics administrator.

```
libname weba '<path>';
data wacnfg.webmart(keep=guid version);
  set weba.webmarts;
  where webmart='your-mart-name';
run;
```

CHAPTER 4

Using SAS Web Analytics with SAS Business Intelligence Tools

Configure Your Site for a SAS Business Intelligence Dashboard

Dashboard Setup Introduction

A SAS BI Dashboard consists of portlets, and each portlet contains one or more indicators. Each indicator has a data source, one or more gauges, and optional hyperlinks. SAS BI Dashboard users can create their own dashboards from various data sources and display them in a SAS Information Delivery Portal. For setup and administration documentation, see the *SAS 9.2 Intelligence Platform: Web Application Administration Guide*.

Setting Dashboard Permissions

Two groups are available for SAS BI Dashboard: BI Dashboard Users and BI Dashboard Administrators. Users who view dashboards only should be restricted to the BI Dashboard Users group. Users who develop dashboard controls should be members of the BI Dashboard Administrators group.

Dashboard Performance Considerations

Performance can vary greatly as a result of a dashboard's definition and settings. The following considerations affect performance:

- Caching. By default, a dashboard obtains data each time the dashboard is viewed. Because each request runs at least one query, multiple queries might be running at all times, which hinders performance and reduces scalability potential. SAS BI Dashboard uses an in-memory least recently used (LRU) cache. If the underlying data is not "stale," then SAS BI Dashboard caches a query's result set. Proper configuration and ample memory enables all queries to obtain results from the cache each time, which avoids performance-intensive queries.

  Caching requires SASTRUST or another user ID to have permission to run all queries. Without this permission, there is a strong possibility that SAS BI Dashboard will not scale, and results will not be returned in a timely manner. The performance will negatively affect other facets of the SAS Web Analytics system because the dashboard also queries the data warehouse.
- SAS Web Analytics information map query parameters. The information maps that are provided with SAS Web Analytics require query parameters, which SAS BI Dashboard cannot accept. Modifying the information maps to accept default parameters, and then configuring them within SAS BI Dashboard will adversely affect the SAS Web Report Studio reports, which depend on the same maps. Therefore, it is strongly recommended that you consult the *SAS 9.2 Intelligence Platform: Web Application Administration Guide* before modifying the query parameters.
- Polling data. Alert objects within SAS BI Dashboard can be configured to poll data constantly to check for an event. However, doing so on a repeated basis can adversely affect the performance of all other facets of the system. It is recommended that you set the heartbeat interval to be based on the guidelines for administering SAS BI Dashboard in the *SAS 9.2 Intelligence Platform: Web Application Administration Guide*. Note that it is not necessary to adjust the heartbeat interval to a setting smaller than an interval that spans two ETL processes. For example, if the ETL process runs daily, set the heartbeat interval to daily also.

Design Considerations

SAS Web Analytics allows for multiple sites per installation, and each site owns a collection of tables that are unique to that site. However, SAS BI Dashboard does not accept parameters, and requires that data must come from a fixed location. Therefore, a dashboard needs to be created for each site. Alternatively, the user can create an aggregation that selects data from multiple sites and use that as a data source for the SAS BI Dashboard.

Choose carefully when selecting an input source. All of the sites' warehouse tables are registered in metadata, and some of them already contain aggregate data. Queries that use the pre-aggregated data to their advantage will probably execute faster than their warehouse counterparts. Also use a boundary, preferably by date, at a minimum. Without a boundary, the system will query the entire table. See Appendix 1 for a complete list of aggregates.

The following table lists all of the statistical fields available throughout the default aggregates and contains various notes on each field. Note that (*) is a wildcard.

Table 4.1: Site Aggregate Data Tables and Statistical Fields

Table	Variable	Description
Daily Total Day Aggregate – Day	Unique_Visitors	Number of unique visitors for a day.
Daily Total Day Aggregate – Day	Repeat_Visitors	Number of repeat visitors for a day.
Daily Total Day Aggregate – Day	Page_View_Med_Session_Cnt	Number of sessions that reached a medium number of pages during a visit.
Daily Total Day Aggregate – Day	Page_View_Low_Session_Cnt	Number of sessions that reached a low number of pages during a visit.
Daily Total Day Aggregate – Day	Page_View_High_Session_Cnt	Number of sessions that reached a high number of pages during a visit.
Daily Total Day Aggregate – Day	Session_Count	Number of visits.
Daily Total Day Aggregate – Day	Page_Count	Number of page views.
Daily Total Day Aggregate – Day	Duration	Average visit time.
Daily Total Day Aggregate – Day	File_Count	Number of files shown.
Daily Total Day Aggregate – Day	One_Hit_Session_Count	Number of visits that open a single page and no more.
Daily Total Day Aggregate – Day	Total_Bytes_Sent	Number of bytes sent to a browser.
All Pages Aggregate	Session_Count	Number of visits.
All Pages Aggregate	Page_Count	Number of page views.
All Pages Aggregate	Entry_Page_Count	Number of times a page was an entry page for a visit.
All Pages Aggregate	Exit_Page_Count	Number of times a page was an exit page for a visit.
All Pages Aggregate	Prosp_Bounce	Number of first valid pages opened in a visit (prospective bounces).
All Pages Aggregate	One_Hit_Session_Count	Number of bounces.

Table	Variable	Description
Page Aggregate – Day	Page_Total	Number of pages for a day (used with page_count to calculate percentages).
Page Aggregate – Day	Status_Cd	Status code (a grouping variable and a statistic).
Platform Aggregate – Day	Session_Count	Number of visits.
Platform Aggregate – Day	Page_Count	Number of pages.
Platform Aggregate – Day	Session_Sum	Total number of visits per day (used with session_count to calculate a percentage).
Platform Aggregate – Day	Page_Sum	Total number of pages viewed per day (used with page_count to get a percentage).
Referrer Search Term Aggregate – Day	Session_Count	Number of visits.
Referrer Search Term Aggregate – Day	Session_Total	Total number of visits per day (used with session_count to calculate a percentage).
AGGR_SEBM_Day	Pg_St_Sessions	Number of visits where visitors entered the site through a search term from an external referrer, and a goal page was viewed for a particular day.
AGGR_SEBM_Day	Ref_St_Cnt	Number of external search engine search visits for a given search term.
AGGR_SEBM_Day	Session_Count	Visit count.
AGGR_SEB_CAMPAIGN_Day	Goal_Visits	Goal page visit count.

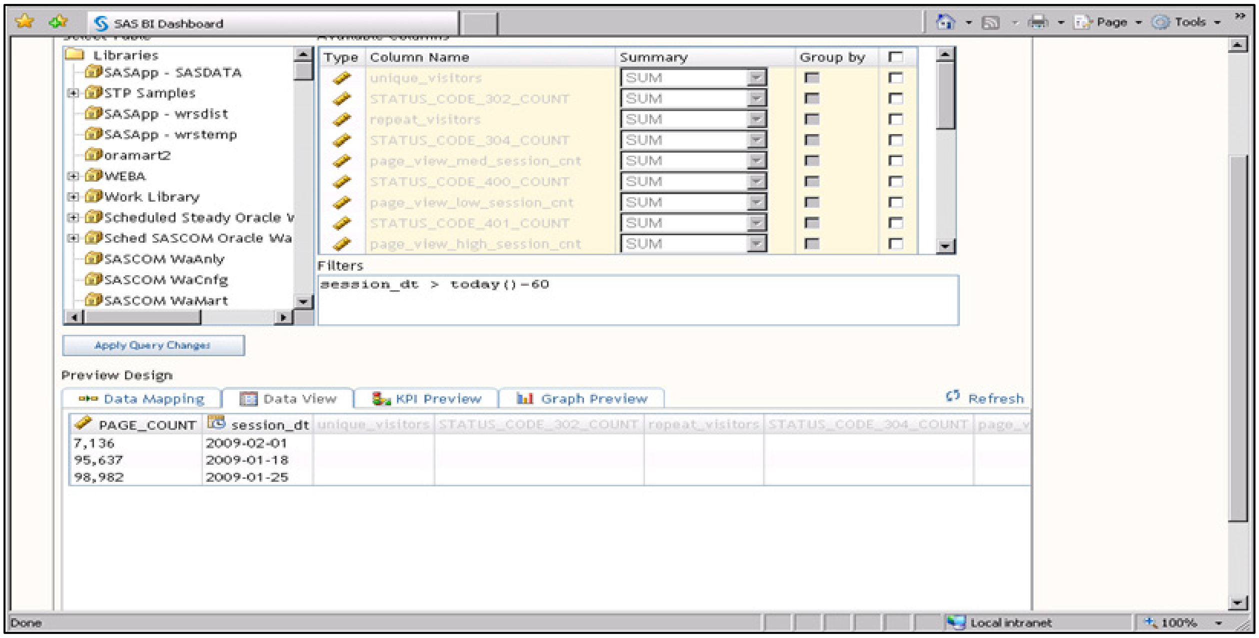
Example: Setting Up a Dashboard

The following example shows the steps to complete to set up a dashboard for users who want to view page views for a Web site, aggregated by week:

1 Create a data model. Use the Daily Total Aggregate – Week table to aggregate page views by week. For more information about aggregates, see “Design Considerations” on page 82.

2 Because all queries should be bound, use session_dt as a limitation.

3 The today() function returns today’s date. This can be applied as shown, where the previous 60 days of data was filtered as **today()-60**.



Libraries
- SASApp - SASDATA
- STP Samples
- SASApp - wrsdist
- SASApp - wrstemp
- oramart2
- WEBA
- Work Library
- Scheduled Steady Oracle V
- Sched SASCOM Oracle Wa
- SASCOM WaAnly
- SASCOM WaCnfg
- SASCOM WaMart

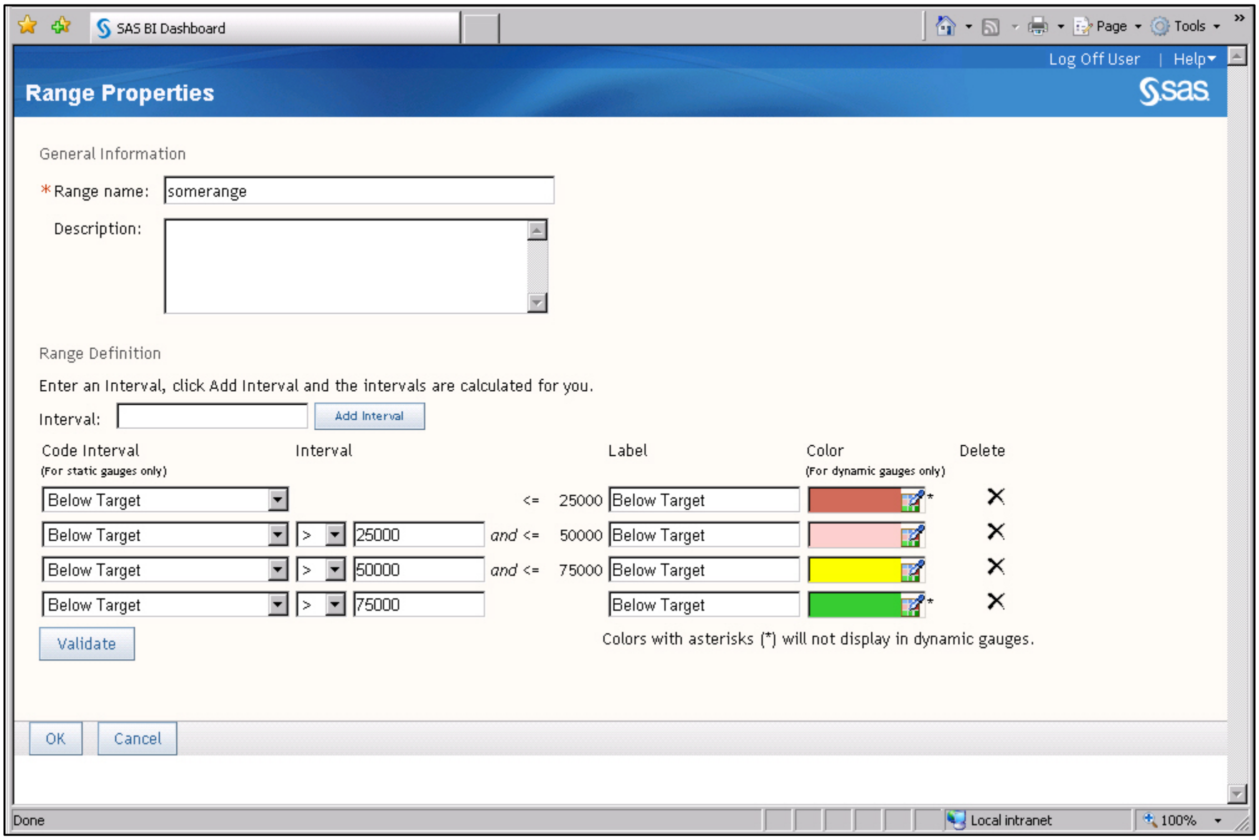
Type	Column Name	Summary	Group by	□
	unique_visitors	SUM	□	□
	STATUS_CODE_302_COUNT	SUM	□	□
	repeat_visitors	SUM	□	□
	STATUS_CODE_304_COUNT	SUM	□	□
	page_view_med_session_cnt	SUM	□	□
	STATUS_CODE_400_COUNT	SUM	□	□
	page_view_low_session_cnt	SUM	□	□
	STATUS_CODE_401_COUNT	SUM	□	□
	page_view_high_session_cnt	SUM	□	□

Filters

```
session_dt > today()-60
```

PAGE_COUNT	session_dt	unique_visitors	STATUS_CODE_302_COUNT	repeat_visitors	STATUS_CODE_304_COUNT	page_v
7,136	2009-02-01					
95,637	2009-01-18					
98,982	2009-01-25					

4 Enter a range for the colors.



Range Properties

General Information

*Range name: somerange

Description:

Range Definition

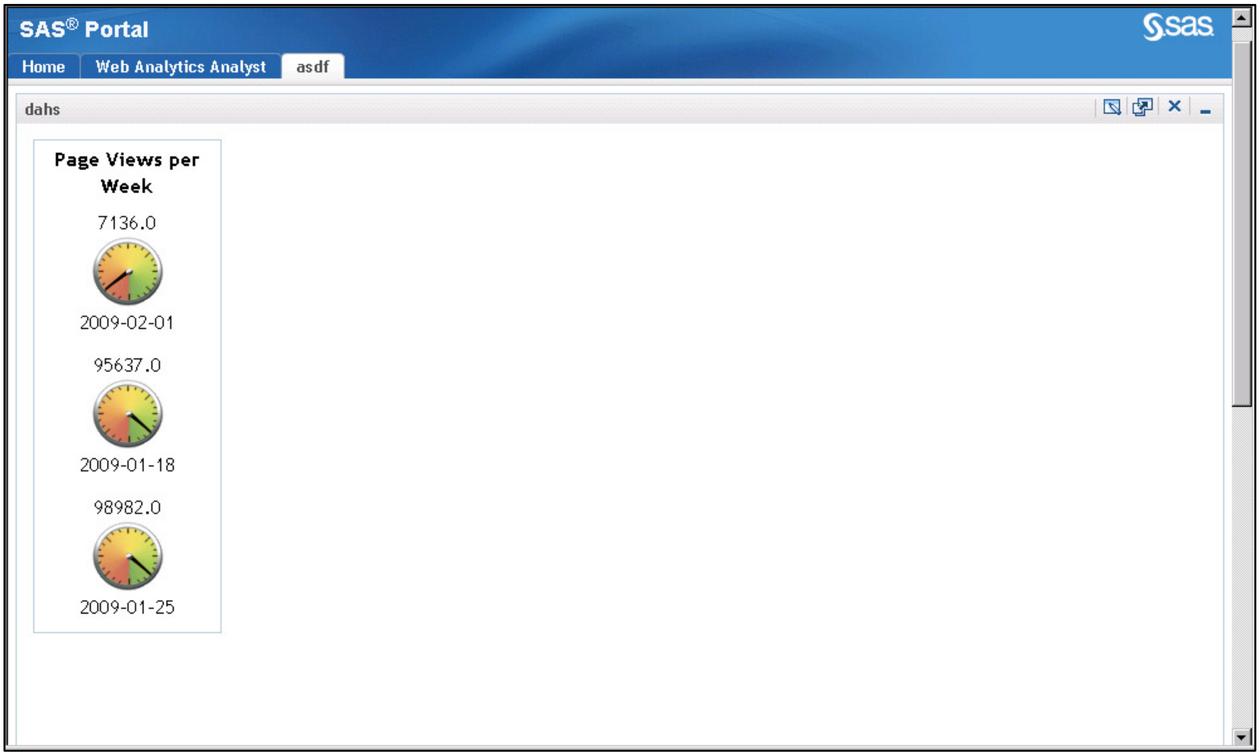
Enter an Interval, click Add Interval and the intervals are calculated for you.

Interval:

Code Interval (For static gauges only)	Interval				Label	Color (For dynamic gauges only)	Delete
Below Target			<=	25000	Below Target	🟥*	✕
Below Target	>	25000	and <=	50000	Below Target	🟥	✕
Below Target	>	50000	and <=	75000	Below Target	🟨	✕
Below Target	>	75000			Below Target	🟩*	✕

Colors with asterisks (*) will not display in dynamic gauges.

5 Check your results.



Page Views per Week

7136.0

2009-02-01

95637.0

2009-01-18

98982.0

2009-01-25

Use SAS Information Map Studio with SAS Web Analytics

Introduction

SAS Web Analytics conforms to the SAS 9.2 Business Intelligence architecture. Therefore, you can use any of the SAS BI tools. For example, if you were working with a Web site named MySite, you would have a set of libraries defined in the metadata repository. From the library MySite WaMart, you could select any set of tables that you want to create your information map from and then use the power of SAS Information Map Studio to add filters, functions, or computed columns.

SAS Information Maps That Are Provided

SAS Web Analytics 5.4 provides numerous SAS Web Report Studio reports, shipped with the product, which are built on SAS Information Maps. These information maps are designed to support a wider set of requirements than custom information maps would support.

The SAS Information Maps that are shipped with SAS Web Analytics 5.4 are deployed when the product is installed. They are located in a reserved area within the metadata repository: **`/Products/SAS Web Analytics/Data Sources/Report Data`**.

Table 4.2: SAS Web Report Studio Reports and Information Maps Shipped with SAS Web Analytics 5.4

SAS Web Report Studio Report Name	Information Map Name	Aggregate Source Stored Process	Stored Process Code	Source Aggregate
Available_data	map_weba_dates	stp_weba_dates	stp_weba_dates.sas	date_dim
Bounce Rate	map_weba_bouncerate	stp_weba_bouncerate	stp_weba_bouncerate.sas	aggr_bounce_day
Browsers	map_weba_platform_browser	stp_weba_platform_browser	stp_weba_platform_browser.sas	aggr_browser_day
Day Of Week	map_weba_daily_total	stp_weba_daily_total	stp_weba_daily_total.sas	aggr_daily_total_day
Error Status	map_weba_pages_client_error	stp_weba_pages_client_error	stp_weba_pages_client.sas	aggr_status_day
Exit Pages	map_weba_pages_exit_pages	map_weba_pages_exit_pages	stp_weba_exit_pages.sas	aggr_exit_pg_day
Hourly Metrics	map_weba_hourly_total	stp_weba_hourly_total	stp_weba_hourly_total.sas	aggr_hourly_total_day
Organic Goal Page Summary	map_weba_organic_goal_pages	stp_weba_organic_goal_pages	stp_weba_organic_goal_pages.sas	aggr_sebm_day
Organic Search Summary	map_weba_organic_search_summary	stp_weba_organic_search_summary	stp_weba_organic_search_summary.sas	(three, including aggr_se_summary_day)
Organic Search Word Effectiveness	map_weba_organic_search_word_effectiveness	stp_weba_organic_search_word_effectiveness	stp_weba_organic_search_word_effectiveness.sas	aggr_sebm_day
Organic Search Word Overview	map_weba_organic_keyword_overview	stp_weba_organic_keyword_overview	stp_weba_organic_keyword_overview.sas	aggr_sebm_day
Pages	map_weba_pages_visits	map_weba_pages_visits	stp_weba_pages.sas	aggr_all_page_day
Platform	map_weba_browser_platform	stp_weba_platform_browser	stp_weba_platform_browser.sas	aggr_platform_dat
Top Referrer Entry Pages	map_weba_pages_entry_pages	stp_weba_pages_entry_pages	stp_weba_pages.sas	aggr_entry_pg_day
Traffic Heatmap - US		stp_weba_us_heatmap	stp_weba_us_heatmap.sas	
Site Metrics	map_weba_daily_total	stp_weba_daily_total	stp_weba_daily_total.sas	aggr_daily_total_day
Status Codes Per Hour	map_weba_status_codes	stp_weba_status_codes	stp_weba_status_codes.sas	aggr_hourly_status_day
Status Codes	map_weba_status_codes	stp_weba_status_codes	stp_weba_status_codes.sas	aggr_hourly_status_day
Visitor Frequency	map_weba_frequency	stp_weba_frequency	stp_weba_frequency.sas	aggr_visitor_daily
Visitor Recency	map_weba_recency	stp_weba_recency	stp_weba_recency.sas	aggr_visitor_daily
Referrer Entry Pages	map_weba_referrer_entry_pages	stp_weba_referrer_entry_pages	stp_weba_first_pages.sas	aggr_entry_pg_day

SAS Web Report Studio Report Name	Information Map Name	Aggregate Source Stored Process	Stored Process Code	Source Aggregate
Search Engine Bid Campaigns	map_weba_sebd _campaign _report	stp_weba_sebd_campaign _report	stp_weba_sebd _campaign_report.sas	aggr_sebd_campaign_ ad_summary
Search Engine Paid Keyword Performance	map_weba_sebd _keyword _performance _with_trending	stp_weba_sebd_keyword _performance_with _trending	stp_weba_sebd_keyword _with_trending.sas	anly_sebd_keyword

CAUTION:

You should not modify these information maps, reports, or aggregates because it could have a negative effect on queries. Many of the information maps use a stored process as their data source, which allows more efficient queries and more advanced prompting. Information maps offer a greater ability to reuse queries, because the same information map or report can dynamically choose a Web site for its data source.

For information about using a SAS Stored Process as a data source for an information map, see the Information Map Studio product Help.

Use SAS Web Analytics with the SAS Information Delivery Portal

Introduction to Using SAS Web Analytics with the SAS Information Delivery Portal

The SAS Information Delivery Portal provides a convenient collection, organization, and launch interface that displays content to authorized users by using role-based security.

SAS Web Analytics provides two Portal Template pages: Web Analytics Analyst 5.4 and Web Analytics Report Users 5.4.

For documentation about the SAS Information Delivery Portal, see the online Help that is accessible from within the product.

Steps for Accessing SAS Web Analytics through the SAS Information Delivery Portal

To access SAS Web Analytics through the SAS Information Delivery Portal:

1 Assign users to either the Web Analytics Analyst or the Report User group. Assign users to either group but not both.

2 Log on to the SAS Information Delivery Portal. The SAS Web Analytics Portal page that displays on one of the tabs is appropriate for your Web Analytics user group. If your user group is Web Analytics Analyst, then the Analyst page displays. If your user group is Web Analytics Report Users, then the Reports page displays.

3 Log in to the SAS Information Delivery Portal as a content administrator and then log off. Users should now be able to view the Portal Template pages. For more information see the SAS Information Portal chapter in the *SAS 9.2 Intelligence Platform: Web Application Administration Guide*.

CHAPTER 5

Working with Search Engine Bid Data

Introduction to Search Engine Bid Management

Search Engine Bid Management (SEBM) involves organizing and managing the keyword buys from the various search engine providers. Many search engine providers offer an application that enables the user to create campaigns and place bids on keywords. Whether a bid is accepted is based on algorithms used by the search engine provider. For a summary of how the auction process works with Google AdWords, see `http://googleblog.blogspot.com/2008/05/how-auctions-set-ad-prices.html`.

In addition, search engine providers have developed their own application programming interfaces (APIs) that allow automated bid creation, updates, and access to bid response data for further analysis. An example of an access application is Google AdWords.

Search Engine Bid Data

Search engine bid data (SEBD) is collected from the search engine provider API. Each provider has steps that users must follow in order to access their API. The type of data available depends on the search engine provider. In SAS Web Analytics 5.4, users can choose to download Google AdWords report data by using the Google AdWords API and the provided ETL jobs to load the data into a data mart. For information about using Google API Web Services, see `http://code.google.com/apis/adwords/docs/developer/index.html`.

Search Engine Data Source

The response data available from Google AdWords includes the keyword, destination, URL, cost, duration, and so on. Here are resources for API information:

- *Google AdWords API Developer's Guide*. Available at `http://code.google.com/apis/adwords/docs/developer/index.html`
- *Google AdWords API ReportService*. Available at `http://code.google.com/apis/adwords/docs/developer/ReportService.html`
- *Google AdWords API Help*. Available at `http://www.google.com/support/adwordsapi/bin/index.py?fulldump=1`
- *Google AdWords API Report Rules*. Available at `http://code.google.com/apis/adwords/docs/developer/adwords_api_reports.html`
- *Google AdWords API Error Codes*. Available at `http://code.google.com/apis/adwords/docs/developer/adwords_api_error_codes.html`

Overview of Search SEBD Setup and ETL Process

This is the basic process for setting up the SEBD data and ETL processes. The details of each step are described in the following sections in this chapter.

1. Set up a Google AdWords account and sign up to use the Google AdWords API. See `http://adwords.google.com` `https://adwords.google.com/select/ApiWelcome`.
2. Determine whether the SEBD data should be stored in SAS Unicode server or as Base SAS. For information about using SAS Unicode, see Chapter 1.

3 (UNIX only) Make sure your UNIX environment is set up for windowing services so you can run the SEBD initialization program.

4 Initialize the SEBD mart.

5 Set up logon information for Google AdWords.

6 Prime the data tables.

7 Schedule and run the ETL jobs.

Initializing the SEBD Mart

Running the SEBD Initialization Job in UNIX

You can run the SAS Web Analytics SEBD Initialization job only within an environment where windowing services are available. Typically, a valid display environment means the following:

- ❑ An X11 server is running (for example, on a UNIX server or on a PC that runs Exceed).
- ❑ The user environment that is invoking the SAS session must have X11 permissions to connect to and use the X11 resources. If necessary, permissions are granted through the **`xhost`** command. The DISPLAY environment variable is set and exported to a valid display on the X11 server (for example, **localhost:0.0**). This display environment is similar to the one users need in order to run in an interactive SAS session or an xterm session, for example.

  To run the Site Initialization job from SAS Data Integration Studio, the SAS Workspace Server must be set up so that when the SAS session is invoked within the workspace server context, the display environment settings are valid as described earlier. Set up the SAS Workspace Server by editing the WorkspaceServer_usermods.sh file and possibly the shell profile of the user that runs the SAS Workspace Server (for example, sassrv).

If you are unable to modify your workspace server's environment, schedule the Site Initialization job, and then deploy and run the job outside of SAS Data Integration Studio. SAS Web Analytics macros need to be available and the display environment should have been set up. For information about scheduling a job, see the SAS Data Integration Studio online Help.

There are three ways to invoke the Site Initialization job once it is deployed:

- ❑ Run the program by using the **`sasbatch`** command found in <*your-server-installation-location*>/**`Config/Lev1/SASApp/BatchServer`** (for example, <*your-server-installation-location*>**`/Config/Lev1/SASApp/BatchServer/sasbatch.sh -`*`sys`*`in`** <*path-to-program*>**`/SEBD_Initialization.sas`**) **`–log`** <*path-to-program*>**`/SEBD_Initialization.log`**).
- ❑ Run the program by using the SAS command found in the installation directory and specify the SASApp application server configuration file (for example, <*your-installation-dir*>**`/sas -config`** <*your-server-installation-location*>**`/Config/Lev1/SASApp/sasv9.cfg path-toprogram/SEBD_Initialization.sas –log path-to-program/SEBD_Initialization.log`**).

❑ Invoke an interactive SAS session. Open and submit the SEBD_Initialization.sas program.

Run the SEBD Initialization Job

1 Invoke SAS Data Integration Studio and navigate to **`/Products/SAS Web Analytics/Sample Jobs`**.

2 Double-click the **SEBD Initialization** job.

3 Double-click the **Weba_SEBD_Initialization** transformation within the selected job, and then click the **Options** tab. Fill in the required information:

a. **User Id** – enter the metadata user ID that is a member of the Web Analytics group or that has Read and Write access to the metadata folder that is specified in the metadata root directory option.

b. **Password** – enter the password that is associated with the user ID.

c. **Physical root path of SEBD datamart specific tables** – select or enter the top level directory where SAS Web Analytics will create SEBD subdirectories and store configurations and other data sets.

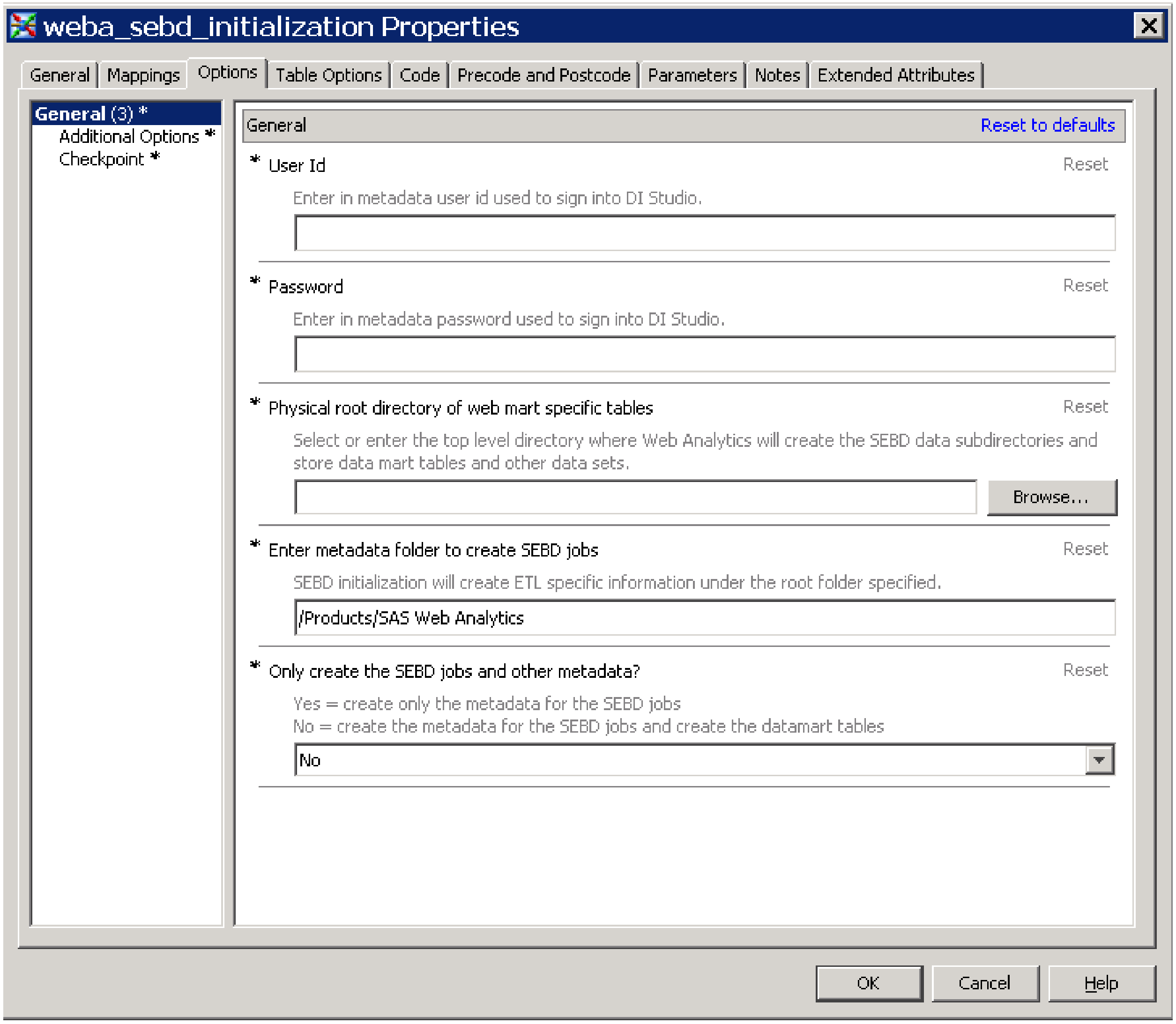
d. **Metadata folder** – by default, SEBD initialization creates the SEBD-specific metadata objects within the **`/Products/SAS Web Analytics`** folder. The user can specify any folder to import the metadata objects.

e. **Only create the SEBD jobs and other metadata?** – enables the user to re-create only the metadata objects. Although the default is No, Select **Yes** in the following scenarios:

❑ A user wants to create the SEBD directories and SEBD data mart tables separately. This scenario could happen within UNIX environments where system administrators are responsible for setting up directories with the appropriate operating system permissions.

❑ A user is moving data from one machine to another (for example, from a development box to a test box). In this case, only the metadata needs to be created.

❑ A user wants to restore jobs and reports to an out-of-the-box condition.


weba_sebd_initialization Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
General (3) *
Additional Options *
Checkpoint *
General
Reset to defaults
Reset
Browse...
OK
Cancel
Help

* User Id

Enter in metadata user id used to sign into DI Studio.

* Password

Enter in metadata password used to sign into DI Studio.

* Physical root directory of web mart specific tables

Select or enter the top level directory where Web Analytics will create the SEBD data subdirectories and store data mart tables and other data sets.

* Enter metadata folder to create SEBD jobs

SEBD initialization will create ETL specific information under the root folder specified.

/Products/SAS Web Analytics

* Only create the SEBD jobs and other metadata?

Yes = create only the metadata for the SEBD jobs
No = create the metadata for the SEBD jobs and create the datamart tables

No

4 Run the job.

(For Windows) Click **Run** in the SAS Data Integration Studio Job pane. If the SEBD Initialization job runs successfully, then a new folder structure is created under the folder that you specified on the **Options** tab.

(For UNIX) Run the job in UNIX by using an interactive SAS session. For more information, see the preceding topic.

5 If the SEBD Initialization job runs successfully, then the SEBD metadata objects will have been imported into the folder that was specified in step 3d.

Note: Importing of the package will not complete if the Web server for SAS Web Report Studio is not running. The following error will occur within the import section of the log if the SAS Web Report Studio server is down:

```
14:48:47 ERROR (com.sas.metadata.promotion)
   org.springframework.ws.client.WebServiceIOException:
I/O error: Connection refused: connect; nested exception
```

```
   is java.net.ConnectException: Connection refused: connect
org.springframework.ws.client.WebServiceIOException: I/O
   error: Connection
nested exception is java.net.ConnectException: Connection
refused: connect
```

Verify Imported Metadata

Using SAS Management Console, Check that these metadata objects were imported during the SEBD Initialization job:

- □ Folder
- □ Library
- □ Table
- □ Stored process
- □ Information map
- □ Report
- □ Job

The following is an inventory of all objects that should be imported into the specified metadata root. You specify the root directory in the SEBD Initialization job. (For more information, see step 3d in "Run the SEBD Initialization Job" on page 93.)

Data Sources

- □ Report Data

  -map_weba_sebd_campaign_report

  -map_weba_sebd_keyword_performance_with_trending

  -Stored Processes

  stp_weba_sebd_campaign_report

  stp_weba_sebd_keyword_performance_with_trending
- □ Work

  -stp_sebd_campaign

  -stp_sebd_keyword_summary

Reports

- □ Search Engine Bid Reports

  -Search Engine Bid Campaigns.srx

  -Search Engine Paid Keyword Performance.srx

SEBD

- □ **`5.4 Jobs`** - see Table 5.1 on page 109 for a complete list of jobs and folder structure. The following folders should be present:

  -Aggregates - contains aggregate jobs

  -Common Jobs - jobs that will be used by any SEBD ETL

  -Extract Google

  AdWords v2009 - contains jobs that will extract data from the Google AdWords API v13 and load the data into the SEBDMART tables

-Warehouse Priming - contains a job that primes the tables within the SEBDMART

- ❑ Data Sources - should contain the following tables and libraries:

  -Extract Google

  -SEBDCNFG

  -SEBDDQ

  -SEBDEXTR

  -SEBDMART

  -SEBDWORK

  Staging Google AdWords

Set Up a Login for Google AdWords API and Manage Account Information

Google AdWords API Login Requirements

Google AdWords Login API requires the following information:

- ❑ My Client Center (MCC) User ID – used to log on to your Google AdWords account
- ❑ MCC Password – used to log on to your Google AdWords account
- ❑ Developer Token – provided by Google when you set up your AdWords account for API access
- ❑ Application Token – provided by Google when you set up your AdWords account for API access

For information about signing up to use the AdWords API and obtaining your application and development tokens, see the *Google AdWords API Developer's Guide* at **http://code.google.com/apis/adwords/docs/developer/index.html#adwords_api_intro_tokens**.

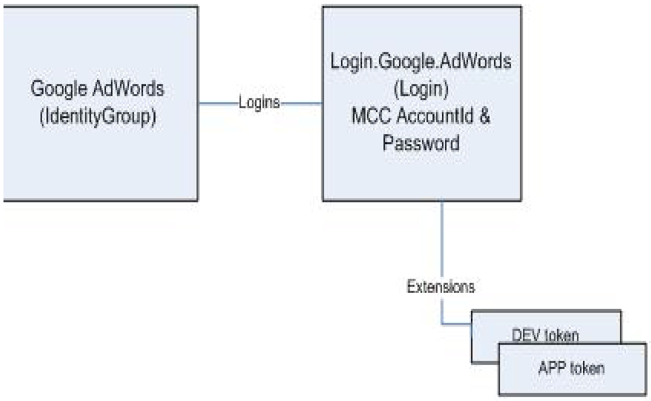
Managing Google AdWords My Client Center Account Login Information

Overview

Google AdWords My Client Center (MCC) account information is stored within the metadata and can be partially accessed by the User Manager in SAS Management Console. A utility macro (discussed in this section) is available to create and update the necessary metadata objects. Figure 5.1 illustrates how the tokens are associated with a specific logon within the metadata.

The ETL downloads data for all AdWords client accounts that are associated with the MCC account.

Figure 5.1: How Tokens Are Associated with Logins within Metadata


Google AdWords (IdentityGroup)
Logins
Login.Google.AdWords (Login) MCC AccountId & Password
Extensions
DEV token
APP token

Administering MCC Accounts

To administer MCC accounts:

1 **Enter Google AdWords MCC logins**

A macro, weba_google_adwords_login_info, is supplied to add Google AdWords MCC logins to the User Manager in SAS Management Console. Run the macro from an interactive SAS session that is set up to access SAS Web Analytics macros. (See “Accessing SAS Web Analytics Macros” in Appendix 4.) Copy an example of macro usage (below) into the Program Editor window of the interactive SAS session and enter the requested information. Run the macro from the SAS session.

This macro requires the following information:

❑ **mcc_userid** = Google AdWords MCC user ID

❑ **mcc_password** = Google AdWords MCC password in clear text (do not encode)

❑ **mcc_apptoken** = Google AdWords application token

❑ **mcc_devtoken** = Google AdWords developer token. For information about obtaining the application and developer tokens, see the *Google AdWords API Developer’s Guide* at **http://code.google.com/apis/adwords/docs/developer/index.html#adwords_api_intro_tokens**.

The macro performs the following actions:

❑ Creates the Google AdWords identity group, if necessary.

Note: The SAS Administrator might prefer to create the identity group and to add members before adding the MCC account information.

❑ Adds or updates an individual login that contains the Google AdWords MCC information.

Note: The password needs to be entered into the macro as actual text during the metadata update process. The password is automatically encoded or encrypted depending on whether SAS/SECURE is installed. For more information, see the *SAS Intelligence Platform: Security Administration Guide*.

Here is an example of macro usage:

```
%let metauser = <user who has user administration privileges, for
example sasadm@saspw>;

options metaserver="<metadata server>"

        metaport=<metadata port, typically 8561>

        metauser="&metauser"

        metapass="<password for metauser>";

;

      %weba_google_adwords_login_info(mcc_userid= <enter mcc user id>

                                  ,mcc_password= <enter mcc password>

                                  ,mcc_apptoken= <enter mcc app token>

                                  ,mcc_devtoken= <enter mcc dev token>

                                  );
```

The following example illustrates how to use the macro to add multiple logins from a CSV file:

```
%let metauser = =<user who has user administration privileges, for
example sasadm@saspw>;

options metaserver="<metadata server>"

        metaport=<metadata port, typically 8561>

        metauser="&metauser"

        metapass="<password for metauser>"

;

filename mccaccts "<file that contains MCC accoount information>";

data logins;

   infile mccaccts dsd delimiter=',' end=eof;

   informat user password app dev $100.;

   input user password app dev;

   if eof then call symput('logins',put(_n_,best.));

 run;

%macro load;

  %do i = 1 %to &logins;

    data _null_;

      set logins;

      if _n_=&i;

      call symput(‘userid’,strip(user));
```

```
        call symput('password',strip(password));
        call symput('app',strip(app));
        call symput('dev',strip(dev));
      run;
      %weba_google_adwords_login_info(mcc_userid=&userid
                                 ,mcc_password=&password
                                 ,mcc_apptoken=&app
                                 ,mcc_devtoken=&dev
                                 );
    %end;
  %mend;
  %load;
```

After the macro has completed, the Google AdWords group and MCC user ID and password are available within the User Manager in SAS Management Console. The application and developer tokens are not visible within the User Manager because they are stored as extensions to the login object within the metadata. An error message is displayed if the tokens were not added or updated. You can view the Work.Google_Adwords_MCC_Accts data set at the completion of the macro to view all logins that are associated with the Google AdWords group.

2 Add members to the Google AdWords identity group

Add the members (users or groups) to the Google AdWords identity group by using the User Manager within SAS Management Console. Add these users to the Google AdWords group:

- ❑ Web Analytics Administrators group
- ❑ Individual users that will be used to run the SEBD ETL

3 Verifying that the Google AdWords API connection works

To verify that the Google AdWords MCC information for all of the entered accounts works, run the following code in an interactive SAS session that has access to SAS Web Analytics macros:

```
%let  metauser=<user>; /*user that is a member of Google AdWords group
not sasadm@saspw or sasdemo*/
  options metaserver="<metadata server>"
          metaport=<metadata port, typically 8561>
          metauser="&metauser"
          metapass="<password for metauser>"
;
%google_adwords_mcc_accts(type=WORK,lib=WORK);
%google_adwords_api_login_test(indsn=work.google_api_acct);
```

Troubleshooting the weba_google_adwords_login_info Macro

No records in GOOGLE_ADWORDS_MCC_ACCTS table

This table could have no records for the following reasons:

- ❑ The Google AdWords user group does not exist.
- ❑ No MCC accounts were entered into the Google AdWords account.
- ❑ If the Google AdWords user group exists and the MCC accounts were entered, then verify that the user ID that is associated with the SASApp Logical Workspace Server is a member of the Google AdWords group or Web Analytics Administrator group.

To determine which user is used to log on to the SASApp Logical Workspace Server:

1. Open SAS Management Console.
2. Expand **Server Manager**.
3. Expand **SASApp.**
4. Expand **SASApp - Logical Workspace Server**.
5. Select **SASApp - Workspace Server** in the window on the right. The **Connection: SASApp - Workspace Server** displays.
6. Right-click **Connection: SASApp - Workspace Server** and select **Properties**.
7. Click the **Options** tab.

If the Authentication domain is DefaultAuth, then SAS Trusted User is typically the user that is used to log on to the SAS Workspace Server. Try adding SAS Trusted User as a member to the Google AdWords group, and then reopening the table.

If another authentication domain is specified, then you need to determine which users are associated with that authentication domain. Add those users to the Google AdWords group.

Google AdWords user group does not exist or has been renamed

If the Google AdWords group does not exist within the User Manager, then create it. Do not rename this group.

Unable to log into metadata server

Check the SAS Administrative user ID and password.

Unable to update MCC account information

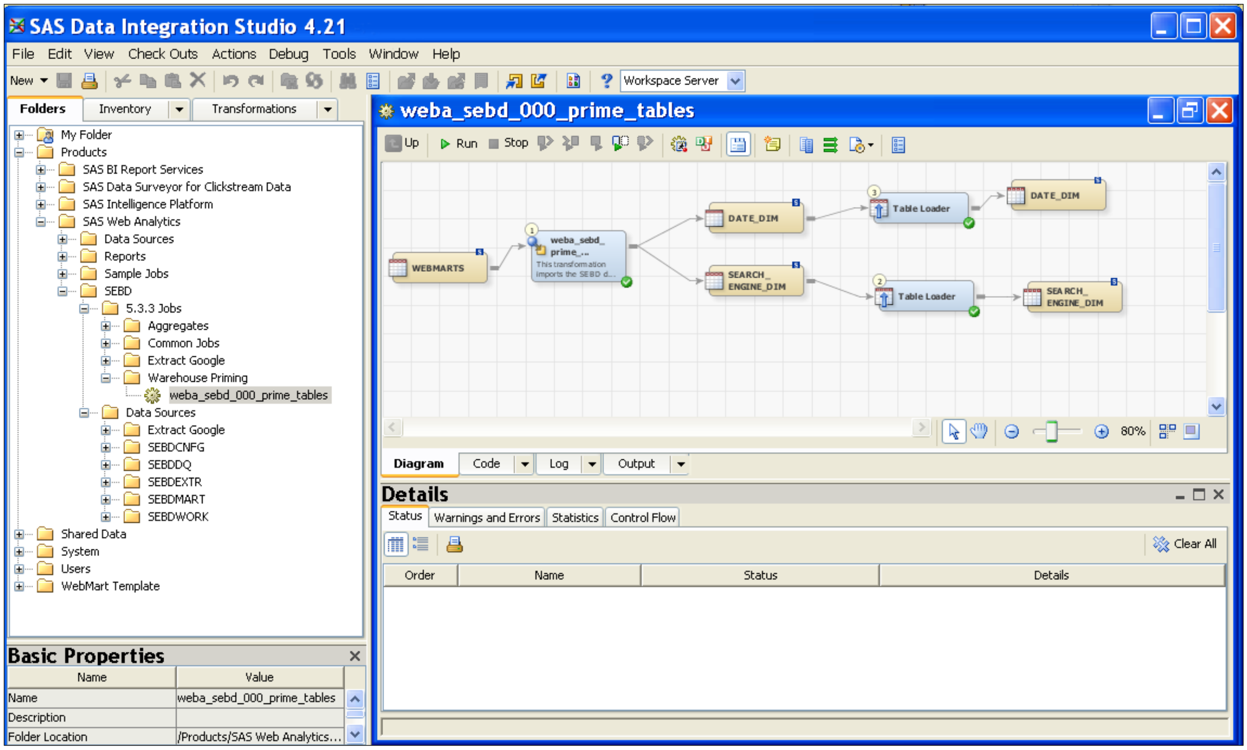
Verify that the SAS Admin user that was specified has Unrestricted metadata access. The default user is sasadm@saspw, which is the preferred user for this job.

Prime the SEBD Tables

CAUTION:
The following steps should occur only if SEBD data mart tables were created during the initialization.

1 Open SAS Data Integration Studio and navigate to the metadata root directory that was specified in the SEBD Initialization job. For more information, see step 3d in "Run the SEBD Initialization Job" on page 93.

2 Double-click the **weba_sebd_000_prime_tables** job in **`/SEBD/5.4 Jobs/Warehouse Priming`**.

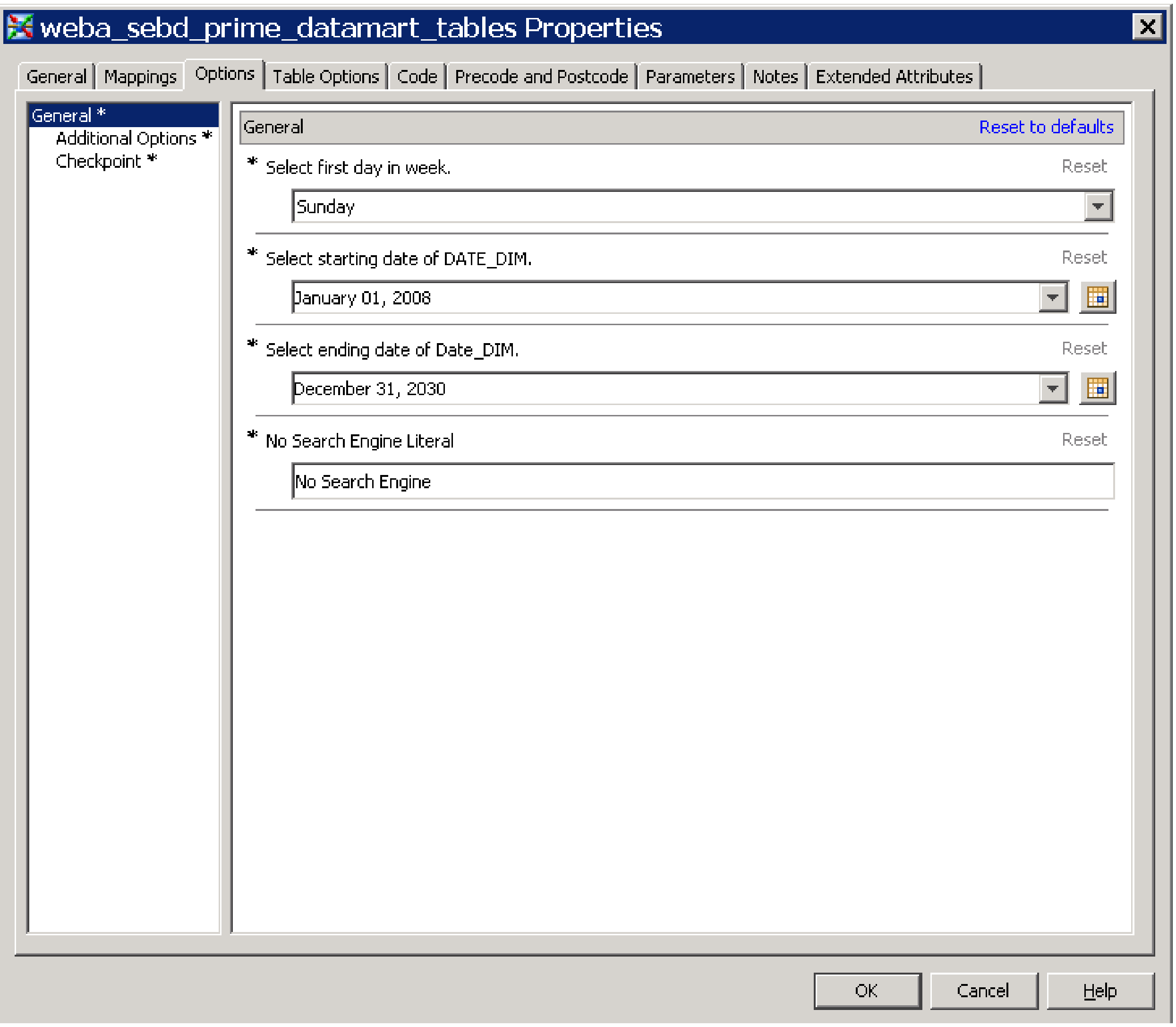


Name	Value
Name	weba_sebd_000_prime_tables
Description	
Folder Location	/Products/SAS Web Analytics…

3 Double-click the **weba_sebd_prime_datamart_tables** transformation within the selected job, and then click the **Options** tab.

4 Select the DATE_DIM table population options:

- ❑ Date range to populate within the table
- ❑ Day to make the first day of the week


weba_sebd_prime_datamart_tables Properties

* Select first day in week.: Sunday

* Select starting date of DATE_DIM.: January 01, 2008

* Select ending date of Date_DIM.: December 31, 2030

* No Search Engine Literal: No Search Engine

5 Click **OK**. Save your changes, and then run the job.

Working with SEBD Data Sources

Google AdWords Data

Google AdWords is a pay-per-click search engine provider. The Google AdWords structure is based on campaigns within an account. For each campaign, you place an AdGroup with its own set of keywords. The structure is described in the All Exam Study Materials 2.1, 2.2, and 2.3 at **http://www.google.com/adwords/learningcenter/text/index.html**.

The data supplied by Google contains the following fields that are documented in these links:

a) Keyword summary and Google AdWord Data Creative and Keyword **http://code.google.com/apis/adwords/docs/developer/adwords_api_report_creative.html**

b) Search Query summary
http://code.google.com/apis/adwords/docs/developer/adwords_api_report_query.html

c) Campaign information
http://code.google.com/apis/adwords/docs/developer/Campaign.html

d) Account information
http://code.google.com/apis/adwords/docs/developer/AccountInfo.html

Extracting Google AdWords Data

The Extraction Process

The Google AdWords API provides information about Google AdWords MCC accounts, keyword campaigns, and search queries used by Google visitors. The data extract process occurs when a Java applet is run that connects to the Google AdWords server. The Java applet executes a set of queries to download data into flat files for each Google AdWords MCC customer client account. The download of Google AdWords data occurs during the Weba_Sebd_002_Google_Adwords_Extract job.

To complete the extract process:

1. Log on to the Google API and download the client accounts that are associated with the Google AdWords MCC account.
2. For each client account, query the Google AdWords API for the keyword campaign and search the query information.
3. Download the information into a set of flat files that are used by the ETL to load the data model.

Troubleshooting AdWords API Errors

When you are working between SAS Web Analytics and the Google AdWords interface, you might encounter an error. To troubleshoot the error, determine which system the error originated from, or whether the error occurred on the network.

To find error messages to assist you, check the extraction log. The errors that are related to the AdWords service are labeled **AdWords API ERROR**. Examples of errors are:

- ❑ a temporary outage
- ❑ a network error
- ❑ an authentication problem

The text of the message points to where the problem is and next steps.

Accessing Additional Features of the Google AdWords API

A fully-functional Google AdWords 2009(09) JAR file is available in the SAS Versioned JAR Repository on the data (server) tier so that you can access additional features of the Google AdWords API. To access the additional features, you can write a Java program. For more information, see the Google AdWords API documentation (available at **http://code.google.com/apis/adwords/**) and the documentation for the JAR file (available at `http://code.google.com/p/google-api-adwords-java/`).

The JAR file is located in the SAS installation area in the following directory:

```
SASVersionedJarRepository/9.2/eclipse/plugins/GoogleAdWords_6.4.0.0_SAS_20100223153431
```

Columns Extracted

The extract process uses a java applet that uses adwords_col.xml, which is created when the SEBD Initialization job is run. The XML file controls what fields are extracted from each Google Adwords table and the format that should be used to read the field into a SAS data set. The adwords_col.xml file is stored with the <SEBDROOT>**`/sebdcnfg`** directory and can be edited to include additional fields.

Caution

Do not delete fields from the Adwords_col.xml file.

ACCT_*IDENTIFIER*

- CustomerId
- Email
- ClientEmail
- CurrencyCode
- DescriptiveName

CMPGN_*IDENTIFIER*

- CustomerId
- CampaignId
- Campaign
- CampaignStatus
- StartDay
- EndDay
- NetworkTypes

AD_*IDENTIFIER*

The following fields are always returned:

- ResponseDate
- CampaignId
- Campaign
- AdGroupId
- AdGroup
- KeywordId
- Keyword
- DescriptionLine1
- DescriptionLine2
- DescriptionLine3
- VisibleUrl
- DestinationURL

- CreativeId
- AdWordsType
- DailyBudget
- MaximumCPC
- MaxContentCPC
- KeywordDestUrlDisplay
- AdGroupStatus
- CreativeDestUrl
- Impressions
- Clicks
- CTR
- CPC
- Cost
- AveragePosition
- Conversions
- ConversionRate
- CostPerConversion
- CostPerTransaction
- FirstPageCpC
- QualityScore

The following fields are returned only if Google AdWords determines that the fields are appropriate for the requested time period:

- CPM
- MaximumCPM

SQ_*IDENTIFIER*

- ResponseDate
- CampaignId
- AdGroupId
- Query
- MatchType
- CreativeId
- AdWordsType
- CreativeDestUrl
- Impressions
- Clicks
- CTR
- CPC
- Cost
- AveragePosition

The following fields are returned only if Google AdWords determines that the fields are appropriate for the requested time period:

- Conversion
- ConversionRate

Google AdWords Data Quality Rules

Data Quality Checks

Data quality checks were added to:

- ❑ Verify that there were no issues with the data downloaded using the Google API.
- ❑ Handle missing values within data fields.

AD_*IDENTIFIER*

The following data quality (DQ) checks are considered to be critical errors. If the errors occur, the record will be excluded from SEBDWORK.GOOGLE_AD.

CampaignId

If the CampaignId field does not exist in SEBDWORK.GOOGLE_CAMPAIGN, then the record is written to SEBDDQ.DQ_AD_*IDENTIFIER* instead of SEBDWORK.GOOGLE_AD.

There are known conditions where Google AdWords return a CampaignId in AD_*IDENTIFIER* but not in CMPGN_*IDENTIFIER*:

From the Google AdWords API Forum:

> The API currently does not support site-CPC campaigns, although support will be added in the future. For now, you will not see them in the getAllAdWordsCampaigns() response, and any attempts to modify them through the API will return faults. See **site-CPC campaigns issue**.

Missing Values

If any of the following fields are blank, then the record is written to SEBDDQ.DQ_AD_*IDENTIFIER* instead of SEBDWORK.GOOGLE_AD :

- ❑ CampaignId
- ❑ Campaign
- ❑ AdGroupId
- ❑ AdGroup
- ❑ KeywordId
- ❑ Keyword
- ❑ DescriptionLine1
- ❑ CreativeId

SQ_*IDENTIFIER*

CampaignId

If CampaignId does not exist in SEBDWORK.GOOGLE_CAMPAIGN, then the record is written to SEBDDQ.DQ_AD_*IDENTIFIER* instead of SEBDWORK.GOOGLE_SQ.

Missing Values

If any of the following fields are blank, then the record is written to SEBDDQ.DQ_SQ_*IDENTIFIER* instead of SEBDWORK.GOOGLE_SQ:

- ❑ CampaignId

- ❑ AdGroupId
- ❑ CreativeId
- ❑ Query
- ❑ MatchType

Setting Up the SEBD Mart ETL

Overview

All SEBD ETL jobs are located in SAS Data Integration Studio in the **`/Products/SAS Web Analytics/SEBD/5.4 Jobs folder`**. The naming convention used was to start the name of the job as follows:

```
weba_sebd_xxx_<job desc>
```

In the naming convention, xxx is the order the job should be run within the ETL schedule.

The following types of jobs can be divided into a Google AdWords extract of specific and general jobs. If a job is specific to Google AdWords, then Google will be part of the job description within the job name.

Optional

jobs that are not necessary for loading the SEBD data mart.

Associated

jobs within SAS Data Integration Studio that are included in a main job through the loop transformation. Associated jobs have _parameterized as the ending of the job name and do not need to be deployed by SAS Data Integration Studio when scheduling the SEBD ETL.

Dependency

jobs that need to finish successfully in order for the job to run.

Job List

Table 5.1: Job List

Job Name	Folder	Optional?	Associated Job	Dependency	Description
Weba_Sebd_001_Create_Zz_Tables	/SEBD/5.4 Jobs/Common Jobs	No	None		Creates ZZ_* versions of all SEBDMART tables to be updated if necessary.
Webd_Sebd_002_Google_Adwords_Extract	/SEBD/5.4 Jobs/Extract Google/ Adwords v2009	No	None	MCC accounts entered in Google AdWords group and developer and application token associated with account. weba_sebd_001_create_zz_tables	Extracts ad, campaign, and account information from Google AdWords for each MCC account and customer combination. The extract program creates three flat files (.dat, .dds and .cntl) for each data type extracted.
Weba_Sebd_003_Google_Extract_Read	/SEBD/5.4 Jobs/Extract Google/ Adwords v2009	No	None	Webd_Sebd_002_Google_Adwords_Extract	Reads all unread *.dat files using the corresponding dds file to create a SAS data set. The corresponding cntl file is used to confirm that the SAS data set contains all records in the .dat file.
Weba_Sebd_011_Google_Read_Signal_File	/SEBD/5.4 Jobs/Extract Google/ Adwords v2009	No	None	Weba_Sebd_003_Google_Extract_Read	Generic job that determines whether there are any Google extract files to process and whether all expected files are available. Deletes SEBDWORK.GOOGLE_* data sets. Determines whether another ETL is currently running and stops if it is.
Weba_Sebd_021_Google_Acct_Extract	/SEBD/5.4 Jobs/Extract Google/ Adwords v2009	No	Weba_Sebd_021_Google_Acct_ Extract _Parameterized	Weba_Sebd_011_Google_Read _Signal_File	Reads in all ACCT_XXXXXXXXXXXXX and appends and updates them to a new version of GOOGLE_ACCT. Note XXXXXXXXXXXXX is the number of milliseconds since Jan 1 1970.
Weba_Sebd_022_Google_Campaign_Extract	/SEBD/5.4 Jobs/Extract Google/ Adwords v2009	No	Weba_Sebd_022_Google_Campaign_Extract _Parameterized	Weba_Sebd_021 _Google_Acct _Extract	Reads in all CMPGN_XXXXXXXXXXXXX and appends and updates them to a new version of GOOGLE_CAMPAIGN. Note XXXXXXXXXXXXX is the number of milliseconds since Jan 1 1970.

Job Name	Folder	Optional?	Associated Job	Dependency	Description
Weba_Sebd_023_Google_Ad_Extract	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	Weba_Sebd_023_Google_Ad_Extract_Parameterized	Weba_Sebd_022_Google_Campaign_Extract	DQ rules implemented in Data Validation node. Reads in all AD_XXXXXXXXXXXXX and appends them to a new version of GOOGLE_AD with a process_dttm timestamp. Note XXXXXXXXXXXXX is the number of milliseconds since Jan 1 1970. Sorts GOOGLE_AD by natural keys and eliminates redundant data by keeping the most current process_dttm record. Note that the code for deleting duplicate data is a job post-process. (See the job properties.)
Weba_Sebd_024_Google_Search_Query_Extract	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	Weba_Sebd_024_Google_Search_Query_Extract_Parameterized	Weba_Sebd_023_Google_Ad_Extract	DQ rules implemented in Data Validation node. Reads in all SQ_XXXXXXXXXXXXX and appends them to a new version of GOOGLE_SQ with a process_dttm timestamp. Note XXXXXXXXXXXXX is the number of milliseconds since Jan 1 1970. Sorts GOOGLE_SQ by natural keys and eliminates redundant data by keeping the most current process_dttm record. Note that the code for deleting duplicate data is a job post-process. (See the job properties.)
Weba_Sebd_029_Google_Extract_Check	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	Yes	None	Weba_Sebd_024_Google_Search_Query_Extract	Verifies that all extract data sets have records. If one has 0 records, then the job will abend.
Weba_Sebd_041_Google_Load_Destination_Domain_Dim	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_029_google_extract_check	Loads PPCWORK.Destination_Domain_Dim with new records.
Weba_Sebd_042_Google_Load_Destination_Page_Dim	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_041_Google_Load_Destination_Domain_Dim	Loads ZZ_DESTINATION_PAGE_DIM with new records.

Job Name	Folder	Optional?	Associated Job	Dependency	Description
Weba_Sebd_043_Google_Load_Campaign_Dim	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_042_Google_Load_Destination_Page_Dim	Loads ZZ_CAMPAIGN_DIM with new records and updates existing records with current information.
Weba_Sebd_044_Google_Load_Adgroup	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_043_Google_Load_Campaign_Dim	Loads ZZ_AD_GROUP_DIM with new records and updates existing records with current information.
Weba_Sebd_045_Google_Load_Ad_Creative_Dim	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_044_Google_Load_AdGroup	Loads ZZ_AD_CREATIVE_DIM with new records and updates existing records with current information.
Weba_Sebd_046_Google_Load_Keyword_Dim	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_045_Google_Load_Ad_Creative_Dim	Loads ZZ_KEYWORD_DIM with new records and updates existing records with current information.
Weba_Sebd_047_Google_Load_Keyword_Metric_Fact	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_046_Google_Load_Keyword_Dim	Loads ZZ_KEYWORD_METRIC_FACT with new records and updates existing records with current information.
Weba_Sebd_048_Google_Load_Search_Query_Fact	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_047_Google_Load_Keyword_Metric_Fact	Loads ZZ_SEARCH_QUERY_FACT with new records and updates existing records with current information.
Weba_Sebd_070_Google_Signal_File_Update	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	No	None	Weba_Sebd_047_Google_Load_Keyword_Metric_Fact	Updates the ppc_signal_file status field to 'LOADED' for each extract file successfully loaded.
Weba_Sebd_071_Google_Cleanup_Extracts	/SEBD/5.4 Jobs/Extract Google/Adwords v2009	Yes	None	Weba_Sebd_070_Google_Signal_File_Update	Optional program and can be moved to the end of the ETL schedule. If an archive directory supplied the files processed in the extract, jobs will be copied to archive. All processed extract files are deleted from the extract area.
Weba_Sebd_099_Flip_Zz_Tables	/SEBD/5.4 Jobs/Common Jobs	No	Weba_Sebd_099_Flip_Zz_Tables_Parameterized	Weba_Sebd_070_Google_Signal_File_Update	Renames current SEBDMART tables to YY_*, and then drops the ZZ_* from all ZZ_* tables, making them the current SEBDMART tables. This program can run in a separate schedule if needed.

Job Name	Folder	Optional?	Associated Job	Dependency	Description
Weba_Sebd_161_Aggr_Keyword_Performance	/SEBD/5.4 Jobs/Aggregates	Yes	None	Weba_Sebd_099_Flip_Zz_Tables	Creates the aggregate and analytical tables - AGGR_SEBD_keyword_day and ANLY_SEBD_keyword_day. This aggregate table is used to populate the Keyword performance report and needs to be run after the SEBD data mart has been loaded.
Weba_Sebd_162_Aggr_Campaign	/SEBD/5.4 Jobs/Aggregates	Yes	None	Weba_Sebd_099_Flip_Zz_Tables	Creates the aggregate and analytical tables - AGGR_SEBD_keyword_day and ANLY_SEBD_keyword_day. This aggregate table is used to populate the Campaign report and can be run either consecutively or concurrently with the other aggregate jobs.
Weba_Sebd_163_Aggr_Traffic_Cmpn_Goal_Pages	/SEBD/5.4 Jobs/Aggregates	Yes	None	Weba_Sebd_099_Flip_Zz_Tables	Summary description: This job creates the AGGR_TRAFFIC_SEB_CMPGN_GOAL table. It reads all available Web marts and summarizes the goal page data by Web mart, page_desc, search_engine_domain, campaign_sk, and seb_campaign_id. Requirements: SEB campaign ID must be part of the destination page and SEB_CAMPAIGN_ID set within at least one (1) Web Mart.

Scheduling SEBD ETL Jobs

For information about scheduling jobs, see Chapter 2, “Setting Up a Web Mart.”

Google AdWords ETL does not complete until 3 p.m. PST. Google has indicated that search query summary data is not available before completion of its ETL. This means that before 3 p.m. PST, search query summary data is not available for the previous day.

Analyzing SEBD and Web Log Data

Overview of Web Log Data Analysis

Search engine bid data (SEBD) is a summarization of campaign responses for a specific time period. A search engine campaign can be designed to affect a single Web site or multiple Web sites. Therefore, the summary data that is provided by the search engine vendors (Google, Yahoo!, or MSN) can contain data for one or more Web sites. In other words, the data is site-independent. In addition, the only connection to Web log data is the destination URI's domain because search engine users might never click specific ads that contain a specific destination page.

To associate a Web log detail visit with a search engine bid (SEB) campaign:

1. Add the vendor campaign ID to the destination URI as seb_campaign=*vendor_campaignid*.
2. In SAS Data Surveyor Clickstream data, add SEB_CAMPAIGN to the CGI parameter parsing request.
3. In SAS Web Analytics, use the value of SEB_CAMPAIGN to populate the SEB_CAMPAIGN_ID field within the SESSION_FACT table.

Vendor-Specific Instructions for GoogleAdWords

1. Obtain a Google campaign ID - this number is available on the browser address line after a campaign is created (for example, `https://adwords.google.com/select/CampaignManagement?campaignId=11204460&mode=)`.
2. Add the campaign ID with a *G* prefix to the Destination URI (for example, ...?seb_campaign=**G**11204460).

Web Log Detail Campaign Aggregate

Combining Aggregates from All Web Marts

A WEBMARTS table is created from the preceding information and used to create a single AGGR_TRAFFIC_SEB_CMPGN_GOAL table. The WAMART libref is associated with the WEBMART that is referenced in the Webmart field. This table is created only if the AGGR_SEB_CAMPAIGN_DAY table is available in any Web mart that is listed within the WEBMARTS table.

Table: AGGR_TRAFFIC_SEB_CMPGN_GOAL

WAMART tables used:

❑ AGGR_SEB_CAMPAIGN_DAY

❑ PAGE_DIM

❑ SEARCH_ENGINE_DIM

SEBDMART tables used:

❑ CAMPAIGN_DIM

Table 5.2: AGGR_TRAFFIC_SEB_CMPGN_GOAL Table Fields

Field	Type	Length	Informat	Format	Table	Description
Webmart	Char	200			Not applicable	Populated with the GUID associated with a specific Web mart.
Search_Engine_Domain	Char	65			Not applicable	Search engine domain from Web mart - for later use when more than one vendor's data is available. (Current version only.) Google search visits are kept.
Seb_Campaign_Id	Char	32			WaMart.Aggr_Seb_ Campaign_Day	Vendor campaign ID (natural key).
Page_Desc	Char	2049			WaMart.Aggr_Seb_ Campaign_Day	A goal page is a special page that has significant meaning where a customer is trying to drive offline traffic to. A goal page is specified by using the goal page administrator that updates wacnfg.cnfg_wasebm.
Session_Dt	Num	8	DATE9.	DATE9.	WaMart.Aggr_Seb_ Campaign_Day	The date visits (sessions) occurred.
Goal_Visits	Num	12			WaMart.Aggr_Seb_ Campaign_Day	Number of visits where visitors entered the site via an SEB campaign and a goal page was viewed for a particular day.
Campaign_Sk	Num	8	8.		Sebdmart.Campaign_ Dim	SEBD campaign sk determined by matching the SEB_CAMPAIGN_ID to CAMPAIGN_ID in CAMPAIGN_DIM.

CHAPTER 6
Working with Oracle Databases

Overview

SAS Web Analytics supports Oracle as a data warehouse repository. The SAS Web Analytics ETL processes load Oracle tables, and reports are extracted from the same tables. SAS Web Analytics uses native Oracle functionality to operate efficiently on Oracle. The Oracle engine is used for queries, and load processes use separate techniques to load the Oracle tables cleanly and efficiently. These options can be set in numerous places. Here are some examples:

- ❑ Indexes can be added to tables, dropped, and re-created on a load.
- ❑ The Oracle UPSERT functionality can be used.
- ❑ The bulk-load facility can be turned off or on, or turned on programmatically, depending on data volume.

Oracle Client Connectivity

To ensure Oracle client connectivity, see “Setting UNIX Environment Variables for SAS/ACCESS” in the *SAS 9.2 Intelligence Platform: Data Administration Guide*, available at **http://support.sas.com/documentation/cdl/en/bidsag/61236/HTML/default/a003112863.htm**. The settings vary depending upon your operating system and configuration.

Note: If the Oracle System ID (SID) and the PATH= name differ in the tnsnames.ora file, then add an extended attribute for oracle_service_name so that the Web application can obtain the SID.

For more information, see the SAS/ACCESS for Oracle documentation, available at **http://support.sas.com/documentation**.

Initial Physical Modeling

An experienced Oracle database administrator (DBA) is a necessity. SAS Web Analytics supplies the logical data model for a Web mart, but makes no assumptions regarding a physical model. Therefore, DBAs need to calculate table sizes over time, and partition or finely tune tables as needs change. Indexes can be added on the Oracle side; SAS ETL will pick them up. For example, if a DBA adds index X to table BROWSER_DIM and has drop and re-create indexes enabled, then the ETL will find it, drop it, load the table, and re-create all indexes, including index X. SAS Web Analytics ships with a DDL that can be used to create a “shell” of a site. This DDL can be modified, or the site can be created by hand. Note that the table names and field names are required by the ETL process.

Warehouses can vary significantly from one application to another. Therefore, it is recommended that you try loading a small amount of data into a temporary site, calculating the growth rates for various tables over a time period, and taking these factors into account. The warehouse requires the same maintenance as any other database.

How Oracle Works with Web Analytics

SAS Web Analytics has several libraries per Web mart. Only the warehouse and aggregate tables (WaMart tables) are stored in Oracle. Ideally, one schema is used per site. This design enables the vast majority of the ETL to refrain from pulling Oracle tables so there is minimal impact to users who might be querying data. SAS Web Analytics implements this in the following ways:

- ❑ by ensuring that keying algorithms take place within SAS. Surrogate keys are generated for each dimension, and SAS keeps track of the converted values.
- ❑ by staging tables (WORKLIB tables) exist as SAS data sets.
- ❑ by tracking which records are to be updated, or appended, or both and which surrogate keys have been used.

In addition to these methods, SAS Web Analytics uses the Oracle engine wherever possible. Here are some examples:

- ❑ Queries. Most reports submit SQL passthrough code to the Oracle engine. The effect is the same as running a query through SQL-Plus because it uses SQL plans and indexes from Oracle.
- ❑ Loads. While records can be loaded using PROC APPEND, the bulk-load parameter can be manipulated. In addition to turning it on or off at a table level, the bulk-load parameter can be data-driven based on the number of input records that it needs to load.

  Note: If bulk loading is to be used from a UNIX environment, set the correct permissions. Write access to the /SASApp application server directory must be granted to all the user IDs that are used to run ETL processes. This directory is found under the *<your-server-installation-location>***`/Config/Lev1`** directory. Failure to grant these permissions results in errors on bulk-load attempts.
- ❑ Indexes. When dropping and re-creating indexes is requested, SAS uses the Oracle engine to tell it what the index definitions are. SAS then re-creates the indexes from these definitions after the load completes.

- Updates and inserts. In certain type 2 tables, administrators can opt for the UPSERT logic, which updates and appends in a single step. This is exclusively SQL passthrough code, so to a certain extent, administrators can tune Oracle performance within SAS.

Configuring SAS Web Analytics Middle Tier and Oracle

JDBC Requirements

The SAS Web Analytics middle tier requires the Java Database Connectivity (JDBC) Thin driver from Oracle (version 10 and later). The method for making the Java Archive (JAR) file available depends on the type of application server in use.

If the necessary Oracle JDBC JAR file is missing, and the Web application user attempts to access a Web mart stored in Oracle, the following error is displayed in the browser:

```
Error    oracle.jdbc.driver.OracleDriver
```

The following shows the contents of the application server log:

```
java.lang.ClassNotFoundException: oracle.jdbc.driver.OracleDriver
 at
org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoad
er.java:1358)
 at
org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoad
er.java:1204)
 at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:319)
 at java.lang.Class.forName0(Native Method)
 at java.lang.Class.forName(Class.java:164)
 at
com.sas.ci.webanalytics.data.ConnectionUrl.getConnection(ConnectionUrl
.java:621)
 at com.sas.ci.webanalytics.data.Data.getConnection(Data.java:4715)
 at com.sas.ci.webanalytics.data.Data.dosql(Data.java:7951)
 at com.sas.ci.webanalytics.data.Data.sql(Data.java:2036)
 at
com.sas.ci.webanalytics.services.WebMartService.getProfiles(WebMartSer
vice.java:456)
 at
com.sas.ci.webanalytics.servlet.WebAnalyticsServlet.process(WebAnalyti
csServlet.java:1162)
 at
com.sas.ci.webanalytics.servlet.WebAnalyticsServlet.doGet(WebAnalytics
Servlet.java:141)
```

Configuring the Application Server

JBoss

To make the JAR available, copy it to the appropriate server's library directory and restart the application server. Here is an example of this location:

```
${JBOSS_HOME}/server/SASServer1/lib
```

To find the correct server, locate the SAS Web Analytics Enterprise Archive (EAR) file (for example, sas.ci.webanalytics.ear).

WebLogic

The WebLogic 9.2 installation should already include compatible drivers in

```
${WL_HOME}/server/lib/ojdbc14.jar
```

If the file is missing, it will have to be added (and the server restarted). If it is the wrong version, it might have to be replaced.

WebSphere

To configure the location of the JDBC JAR file in WebSphere, use the administrative user interface. Select the **Resources ▸ JDBC ▸ JDBC Providers** configuration screen. Ensure that there is a provider for this JAR, and add one if there is not. You will need to provide the pathname to the JAR.

To add the provider:

1 Click **New** at the top of the JDBC provider table.

2 In the Step 1: Create Provider window, enter the field information as follows:

   Database Type = Oracle
   Provider Type = Oracle JDBC Driver
   Implementation Type = Connection Pool Data Source
   Name = Oracle JDBC Provider

3 In the Step 2: Enter Database Class Path Information window, enter the path to the location of the Oracle JAR in the **Directory** field.

CHAPTER 7

Using SAS Web Analytics with SAS Customer Intelligence

How Do SAS Web Analytics and SAS Customer Intelligence Work Together?

Introduction to SAS Web Analytics and SAS Customer Intelligence Campaigns

SAS Web Analytics analyzes the success of campaigns that are created and executed through SAS Customer Intelligence products. This section describes how the SAS Web Analytics and SAS Customer Intelligence products work together.

Marketing campaigns that are created in SAS Marketing Automation can be delivered to the customer by SAS Digital Marketing in an e-mail broadcast or by SAS Real-Time Decision Manager as an offer for single channel and multi-channel campaigns.

SAS Marketing Automation, SAS Digital Marketing, and SAS Real-Time Decision Manager send identifiers for customers (also called subjects) as subject IDs, and send identifiers for campaigns as response tracking codes and treatment tracking codes. These tracking codes identify SAS Marketing Automation cell_package, treatment, and business context values. SAS Digital Marketing adds these identifiers to e-mail broadcasts that contain links to target Web sites. SAS Real-Time Decision Manager adds these identifiers to Web site offers. The e-mails and offers are presented to customers as part of a campaign.

If a recipient opens the e-mail and clicks the link to a Web site (known as a click-through) or responds to an offer, the subject ID, response tracking code, and possibly the treatment tracking code (RTDM offers only) are sent to SAS Web Analytics through one of the following methods:

- SAS Data Surveyor for Clickstream Data, which processes the Web site's Web logs.
- If the SAS Tag Data Format is used for page tagging, a data collection server collects the output and writes the output to page tagging logs.

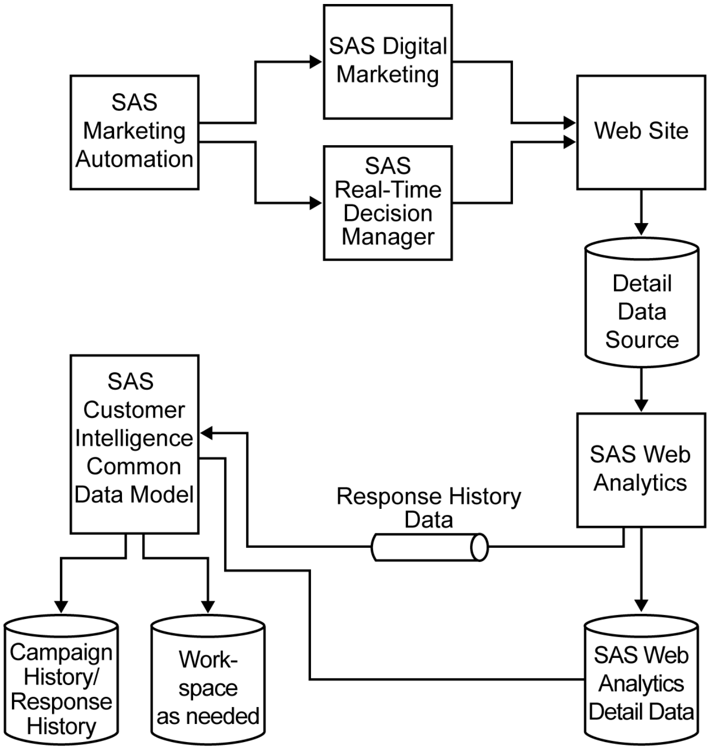
See "Stages in Template Jobs," "About the Customer Intelligence Template Jobs," and Collecting Campaign Information in a Customer Intelligence Job" in the *SAS Data Surveyor for Clickstream 2.2: User's Guide*.

For details about configuration, see Chapter 10, "Processing Campaign Data" in the *SAS Data Surveyor for Clickstream 2.2: User's Guide*.

Using the information from the Web log or page tagging log, SAS Web Analytics can track online behavior details for the campaign.

In SAS Web Analytics, campaign goals are created, identifying the pages in the Web site that are associated with a campaign. When it is detected that a customer visited one or more of these pages, SAS Web Analytics returns the subject ID, response tracking code, and treatment tracking code to the SAS Customer Intelligence Common Data Model to update the response history tables. There, the identifiers are matched to the original campaign, communication, and treatment or package.

Figure 7.1: Flow of Data from SAS Customer Intelligence to SAS Web Analytics


SAS Marketing Automation
SAS Digital Marketing
SAS Real-Time Decision Manager
Web Site
Detail Data Source
SAS Customer Intelligence Common Data Model
SAS Web Analytics
Response History Data
Campaign History/ Response History
Work-space as needed
SAS Web Analytics Detail Data

Campaign Features in SAS Web Analytics

In SAS Web Analytics, the Web analyst can create goals for a campaign and associate a SAS Customer Intelligence response code with the campaign goal. The Web analyst can analyze how SAS Customer Intelligence campaigns are driving traffic to Web sites or how well campaigns are driving actions within Web sites.

Prerequisites for Using SAS Web Analytics with SAS Customer Intelligence

1 Configure SAS Marketing Automation.

2 Configure SAS Digital Marketing, SAS Real-Time Decision Manager, or both.

For configuration information, see the *SAS Digital Marketing 5.4: Administrator's Guide,* the *SAS Marketing Automation 5.4: Administrator's Guide,* and the *SAS Real-Time Decision Manager 5.4: Administrator's Guide*.

Configure SAS Web Analytics to Monitor Campaign Goals

General Configuration Steps

1 Define campaign goals in the SAS Web Analytics Web application.

2 Define parameters for the job that updates customer response history in SAS Marketing Automation (Weba_2550_MA_Response_History_Update).

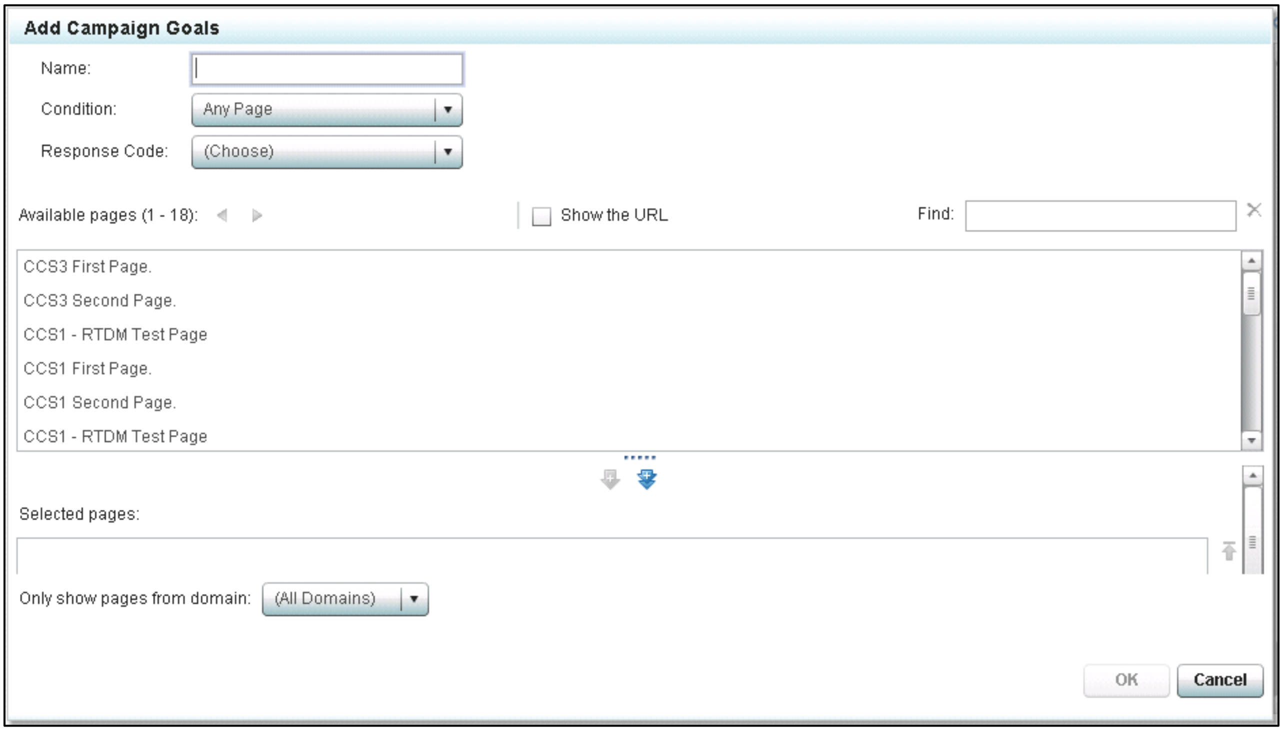
Define Campaign Goals

To define campaign goals in SAS Web Analytics, complete these steps. For descriptions of steps and fields, see the *SAS Web Analytics 5.4: User's Guide*.

1 Select **Campaign Goal Management** from the **Administration** tab.

2 Select the Web site for this campaign from the **Web site** drop-down menu.

3 Add campaign goals and select goal pages for the campaign.



Add Campaign Goals

Name:

Condition: Any Page

Response Code: (Choose)

Available pages (1 - 18):

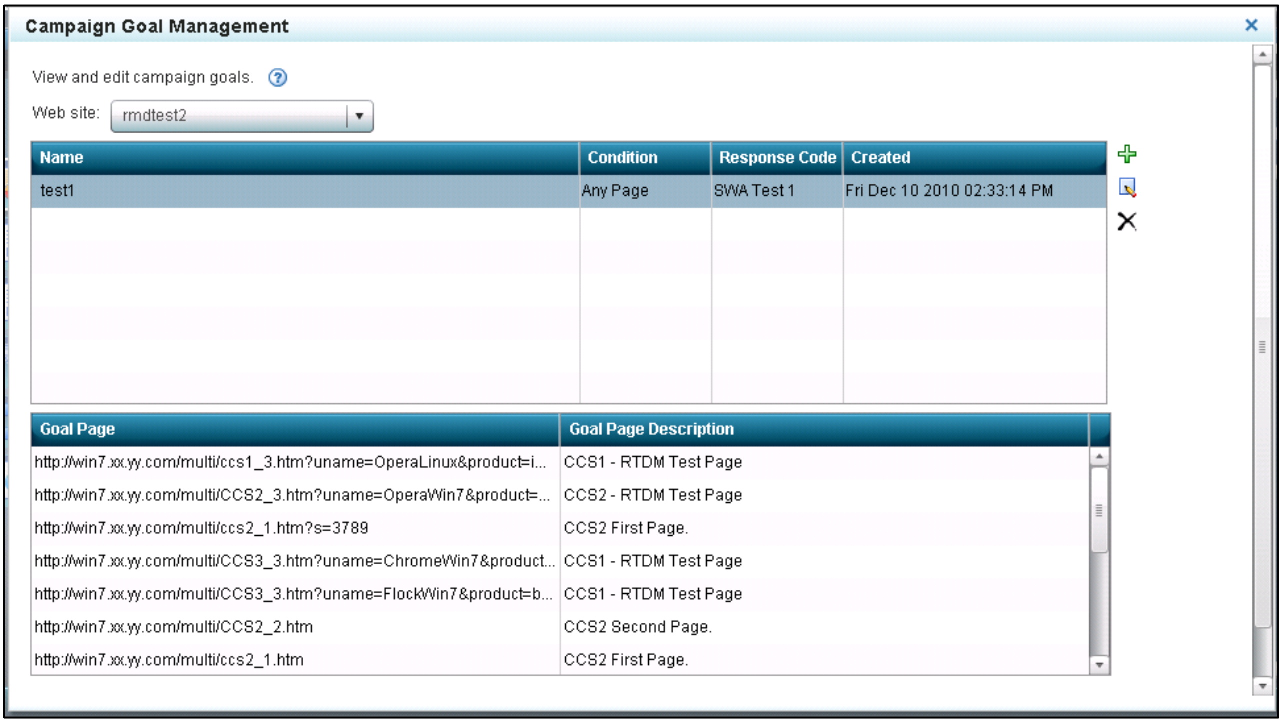
□ Show the URL

Find:

CCS3 First Page.
CCS3 Second Page.
CCS1 - RTDM Test Page
CCS1 First Page.
CCS1 Second Page.
CCS1 - RTDM Test Page

Selected pages:

Only show pages from domain: (All Domains)



Campaign Goal Management

View and edit campaign goals.

Web site: rmdtest2

Name	Condition	Response Code	Created
test1	Any Page	SWA Test 1	Fri Dec 10 2010 02:33:14 PM

Goal Page	Goal Page Description
http://win7.xx.yy.com/multi/ccs1_3.htm?uname=OperaLinux&product=i...	CCS1 - RTDM Test Page
http://win7.xx.yy.com/multi/CCS2_3.htm?uname=OperaWin7&product=...	CCS2 - RTDM Test Page
http://win7.xx.yy.com/multi/ccs2_1.htm?s=3789	CCS2 First Page.
http://win7.xx.yy.com/multi/CCS3_3.htm?uname=ChromeWin7&product...	CCS1 - RTDM Test Page
http://win7.xx.yy.com/multi/CCS3_3.htm?uname=FlockWin7&product=b...	CCS1 - RTDM Test Page
http://win7.xx.yy.com/multi/CCS2_2.htm	CCS2 Second Page.
http://win7.xx.yy.com/multi/ccs2_1.htm	CCS2 First Page.

Note that when the condition **Any Page** is selected, the visitor can view any of the pages selected in order for the goal to be reached. If **All Pages** is selected, then the visitor must view all the pages selected in order for the goal to be reached.

To change an existing campaign goal:

1. Select the goal and click the Edit icon next to the goal information.
2. Change the Name, Condition, and Response Code fields, or pages that complete the campaign goal.
3. Click **OK**.

The campaign goal is updated. A new version of it is created and its history is preserved for accurate tracking between SAS Marketing Automation and SAS Web Analytics.

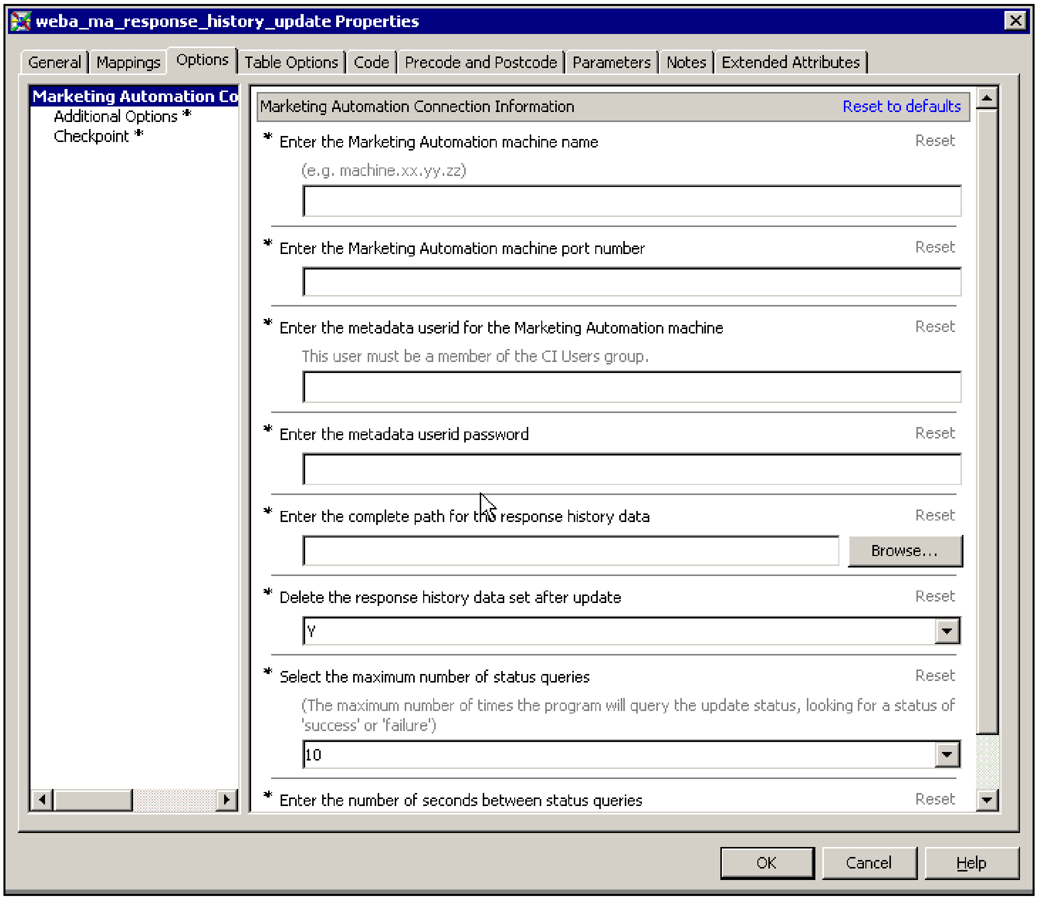
Define Parameters for the Job That Updates Response History

The SAS Web Analytics job Weba_2550_MA_Response_History_Update must be modified to supply the information that is necessary to complete the updates in SAS Marketing Automation.

1. Open SAS Data Integration Studio and navigate to **Shared Data/Web Analytics/**<*Web mart*>**/5.4 Jobs/Optional Jobs**.
2. Double-click the **weba_2550_ma_response_history_update** job.
3. Right-click the **weba_ma_response_history_update** node to open the

   Properties window. Click the **Options** tab and supply values for all the options.

   You need to supply the machine information, passwords, and locations for response history data so that SAS Web Analytics can update the response history data on the machine that is running SAS Marketing Automation. This machine is located on the middle tier for SAS Marketing Automation.


weba_ma_response_history_update Properties

* Enter the Marketing Automation machine name

(e.g. machine.xx.yy.zz)

* Enter the Marketing Automation machine port number

* Enter the metadata userid for the Marketing Automation machine

This user must be a member of the CI Users group.

* Enter the metadata userid password

* Enter the complete path for the response history data

* Delete the response history data set after update: Y

* Select the maximum number of status queries

(The maximum number of times the program will query the update status, looking for a status of 'success' or 'failure')

10

* Enter the number of seconds between status queries

Marketing Automation machine name

the name of the machine that runs the SAS Marketing Automation Web services, typically the middle-tier machine.

Marketing Automation machine port number

the port that is used by the SAS Marketing Automation Web services.

Metadata userid for the Marketing Automation machine and **Metadata userid password**

a user ID and password with permissions to update the SAS Marketing Automation response history data.

Complete path for the response history data

the path to which the temporary data that is used to update the SAS Marketing Automation response history tables will be written. The user ID that runs this job must have Write access to this directory.

Delete the response history data set after update

Default value is Y.

Note: When the value of this option is **Y**, the user ID that is used by the Stored Process Server (typically sassrv) must have Modify access to the path that is specified in the **Complete path for the response history data** field.

Maximum number of status queries

Default=10.

Number of seconds between status queries

Default=30.

For information about response history tables and SAS Customer Intelligence architecture, see the *SAS Marketing Automation 5.4: Administrator's Guide.*

Run the ETL Job to Update Response History Data

Steps for Running the Response History ETL Job

The ETL job that updates the response history data is Weba_2550_MA_Response_History_Update. The response history data is updated on the machine that is running SAS Marketing Automation.

Note: The job Weba_2550_MA_Response_History_Update is classified as a SAS Web Analytics optional job because you must have SAS Marketing Automation installed in order for the job to run properly. However, if you are running this job daily for your Web site, you should run it with your other required ETL jobs before the Weba_2600_ETL_Done job. For job dependency information, see Appendix 7.

Relationship of the Weba_1200_Warehouse_Staging_Tables Job to Response History Updates

Four columns that are related to response history data are stored in the SAS Web Analytics WaMart.Detail_Fact table. These columns are defined and populated during the execution of the Weba_1200_Warehouse_Staging_Tables job, which must execute successfully before the Weba_2550_MA_Response_History_Update job. If you want custom analysis of response history activities, this information can be combined with that of other tables by using the relevant keys (for example, Page_Sk can be used to retrieve information from the WaMart.Page_Dim table). Here are the columns:

Entry_Source_Id

SDM, for SAS Digital Marketing, or RTDM, for SAS Real-Time Decision Manager. These values are set during Clickstream data processing.

Entry_Action_Id

the response tracking code that is extracted from a Web log. This is the identifier that is captured when an e-mail recipient clicks on a link in an e-mail broadcast sent by SAS Digital Marketing.

Entry_Treatment_Id

the treatment tracking code that is extracted from a Web log. This identifier, along with a response tracking code, is captured when a customer clicks on an online offer generated by SAS Real-Time Decision Manager.

Subject_Id_Txt

the concatenation of all S1-S*n* values (subject ID information) from the Web log. This information is sent to the Web log along with the entry_action_id value and entry_treatment_id. Values are separated by an ampersand (&). The default length is $65 and holds at least two subject_id values with no truncation. This column can be expanded to accommodate additional subject_id values, if necessary.

In addition, five WaWork library data sets are created, which are used to supply the Weba_2550_MA_Response_History_Update job:

WaWork.RH_All_Goalsmet

WaWork.RH_Any_Goalsmet

WaWork.RH_Sessions_For_All_Goalsmet

WaWork.RH_Update_Activegoals_Processed

WaWork.RH.baddata

Columns the ETL Populates to Send to SAS Marketing Automation

The Weba_2550_MA_Response_History_Updates job creates a table in the directory that is supplied as a parameter to the job. This table is named RH_*nnnnnnnn*, where *nnnnnnnn* is a unique numeric identifier for the table. The data values that are supplied in this table are used by the SAS Marketing Automation RHUpdate Web service to update the associated response history tables. If the Delete response history data set after update option is set to Yes for the job, the table is deleted only after a successful update. If any invalid data was detected, it is stored in a data set named RH_*nnnnnnnn*_baddata in the same directory. This data set is not deleted automatically.

Columns Assigned by SAS Web Analytics

Resptracking_Cd (from Weblog, WaMart.Detail_Fact.Entry_Action_ID)
tmtTracking_Cd (from Weblog, WaMart.Detail_Fact.Entry_Treatment_ID)
Subject_ID1-Subjectid_N (from Weblog, WaMart.Detail_Fact.Subject_ID_Txt)
Response_Cd (from SAS Marketing Automation through
Wacnfg.Cnfg_WaCmpGoal.ResponseCode)
Response_Channel_Cd='_WC'
Response_Dttm (datetime goal was achieved)
External_Response_Info_ID1=WaMart.Detail_Fact.Detail_Sk
External_Response_Info_ID2=WaCnfg.Cnfg_WaCmpGoal.ID

Column Assigned by SAS Marketing Automation

Inferred_Response_Flg

Troubleshooting the Response History ETL Job

The Weba_2550_MA_Response_History_Update job performs two major functions:

- ❑ to capture the columns that are described in the preceding topic and correctly populate the RH_*nnnnnnnn* table.
- ❑ to request that SAS Marketing Automation process the data in the RH_*nnnnnnnn* table and report on the status of the operation.

If the Response History ETL job fails, examine the SAS log to determine which function experienced a problem. If the problem was in creating the RH_*nnnnnnnn* table, verify the following information:

- ❑ Campaign goals have been defined.
- ❑ The Weba_1200_Warehouse_Staging_Tables job completed successfully.
- ❑ The five WaWork data sets WaWork.RH_All_Goalsmet, WaWork.RH_Any_Goalsmet, WaWork.RH_Sessions_For_All_Goalsmet,

WaWork.rh_baddata, and WaWork.RH_Update_Activegoals_Processed were created successfully.

Note: The WaWork.rh_baddata might contain no observations.

- □ The user ID that is running the program has Write access to the directory supplied as the path for the response history data.
- □ If the value of the **Delete the response history data set after update** option is **Y**, be sure that the user ID that is used by the Stored Process Server (typically sassrv) has Modify access to the directory supplied as the path for the response history data.
- □ WaCnfg.Cnfg_Wacmpgoal was updated successfully.

If the preceding information has been verified and the RH_*nnnnnnnn* table was successfully created, then the problem is in the interaction between SAS Web Analytics and SAS Marketing Automation. Check these items to identify the problem.

- □ If you encounter the following error, the machine name, the port number, or both for the SAS Marketing Automation machine are incorrect, or the SAS Marketing Automation Web services are not running. Contact the SAS Marketing Automation administrator for more information.

```
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Error
encountered executing PROC SOAP.
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Please contact
the web analytics administrator for assistance.
```

- □ If the request to update the response history data is successfully received by SAS Marketing Automation, a message similar to this one should be found in the log:

```
NOTE: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Execution Ticket
RHBUPOP1253650928019 received successfully.
```

- □ An error subsequent to the preceding note typically indicates a problem in either the setup of the SAS Marketing Automation Web services, or actual problems with the data that was received from the Web logs. In the latter case, all the information that was received from the SAS Marketing Automation Web service is printed in the log, and might look something like this:

```
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Update of MA Response History
failed.

ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Please contact the web analytics
administrator for assistance.

ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Execution Failed.
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) StpResultsError
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) The results of the stored process
execution were not successful
ERROR: (WEBA:WEBA_MA_RESP_HISTORY_UPDATE) Execution failed.CCS Return Code:
[785] CCS Return Message: [Error processing History tables.  Check log and
MAMISC.CI_HISTORY_ERROR_REPORT for details] SAS Return Code: [100000] Node
name: [] Table name: [] Column name: []
```

If this error occurs, contact the SAS Marketing Automation administrator to determine the location of the **`MAMISC`** directory (typically on the SAS Marketing Automation middle tier or the SAS tier machine) and the stored process logs. The CI_HISTORY_ERROR_REPORT might contain information necessary to identify a data problem, such as unique constraint violations.

In the case of any type of error, the RH_*nnnnnnnn* table and its associated .xml file will remain in the response history target directory even if the Delete response history data set after update option is set to Yes.

CHAPTER 8

OLAP Setup and Configuration for SAS Web Analytics

Introduction to OLAP Setup and Configuration

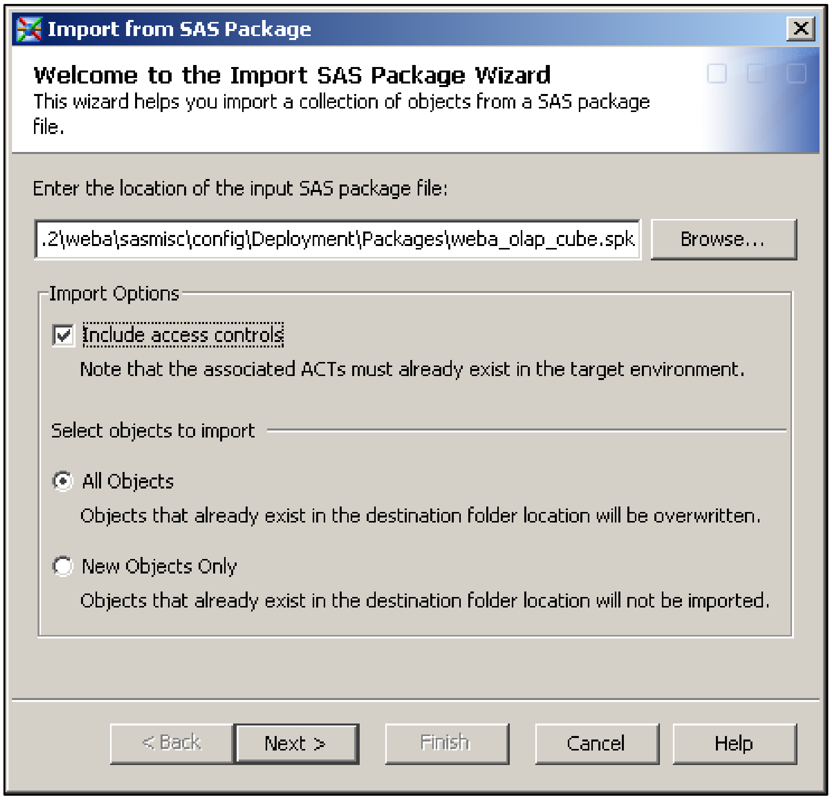
The instructions in this chapter help you create OLAP cubes, metadata, and jobs. Repeat these instructions for each Web mart for which OLAP cubes are required.

Import the Objects

1 In SAS Management Console, navigate to the **Folders** tab that will contain the OLAP metadata. The folder **`/Shared Data/Web Analytics/`**<*Web mart*> is strongly recommended, where *Web mart* is the Web site that will contain the OLAP objects. Right-click on the folder, and then select **Import ▸ SAS Package**.

2 In the Import from SAS Package window, select the **weba_olap_cube.spk** file from **`!sasroot\weba\sasmisc\config\Deployment\Packages`** (Windows) or **`!sasroot\misc\weba\config\Deployment\Packages`** (UNIX). The Import from SAS Package wizard appears.

3 Select **Include access controls** and **All Objects**. Click **Next**.



Import from SAS Package

Welcome to the Import SAS Package Wizard

This wizard helps you import a collection of objects from a SAS package file.

Enter the location of the input SAS package file:

.2\weba\sasmisc\config\Deployment\Packages\weba_olap_cube.spk

Import Options

☑ Include access controls

Note that the associated ACTs must already exist in the target environment.

Select objects to import

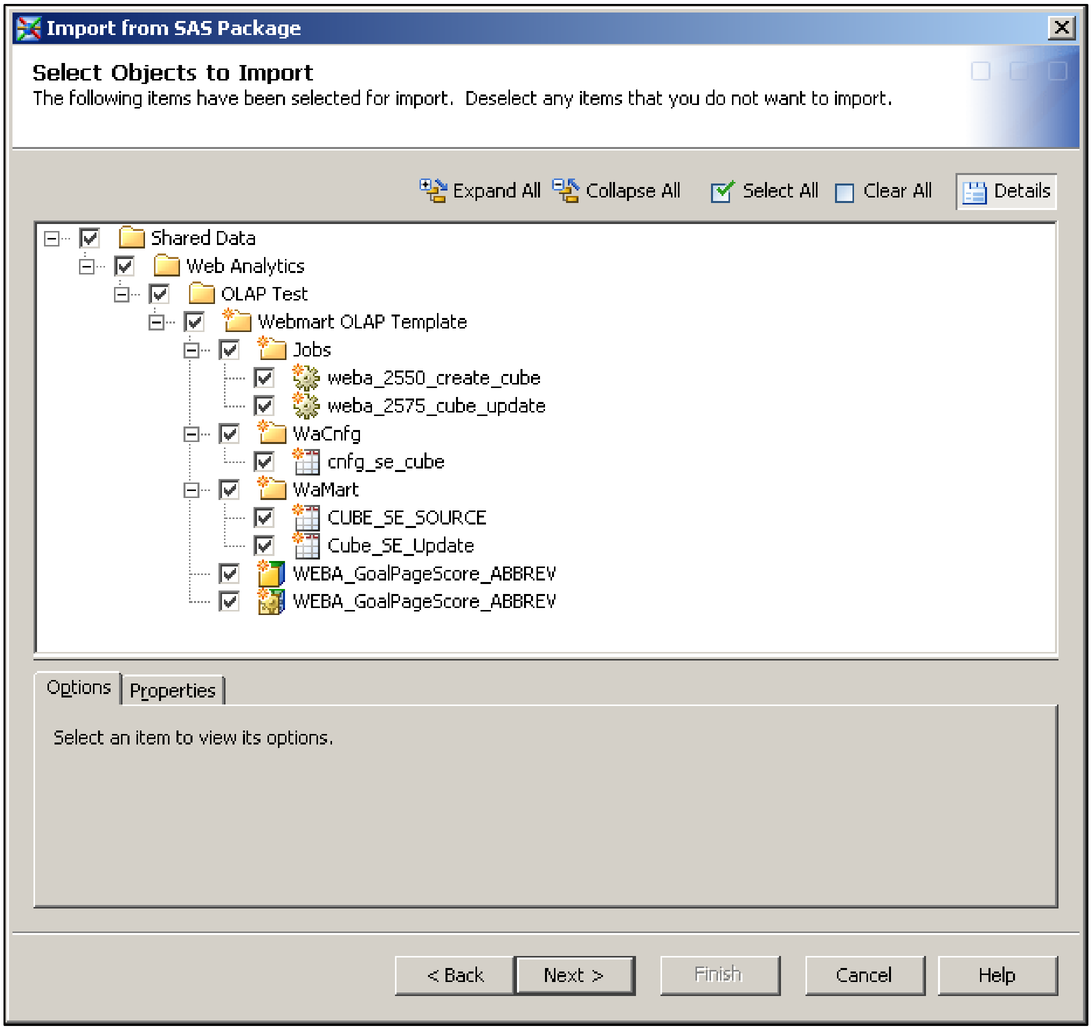
● All Objects

Objects that already exist in the destination folder location will be overwritten.

○ New Objects Only

Objects that already exist in the destination folder location will not be imported.

4 Select all objects for the import process. Click **Next**.



Select Objects to Import

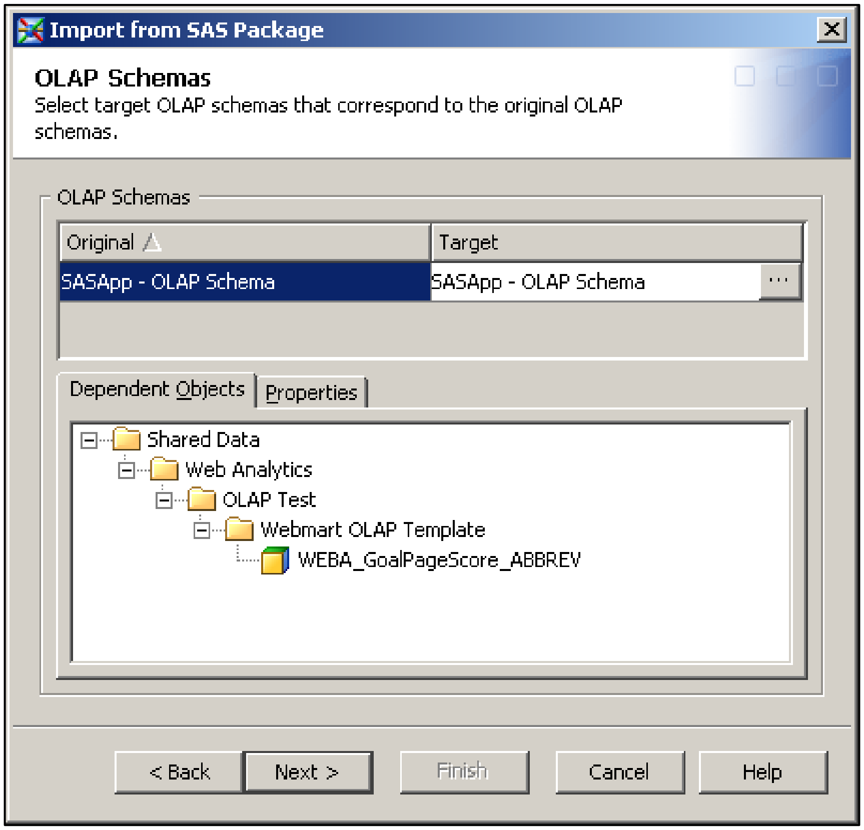
The following items have been selected for import. Deselect any items that you do not want to import.

- ☑ Shared Data
  - ☑ Web Analytics
    - ☑ OLAP Test
      - ☑ Webmart OLAP Template
        - ☑ Jobs
          - ☑ weba_2550_create_cube
          - ☑ weba_2575_cube_update
        - ☑ WaCnfg
          - ☑ cnfg_se_cube
        - ☑ WaMart
          - ☑ CUBE_SE_SOURCE
          - ☑ Cube_SE_Update
        - ☑ WEBA_GoalPageScore_ABBREV
        - ☑ WEBA_GoalPageScore_ABBREV

Select an item to view its options.

5 In the About Metadata Connections dialog box, click **Next**.

6 Select the cube’s schema and click **Next**.



Import from SAS Package

OLAP Schemas

Select target OLAP schemas that correspond to the original OLAP schemas.

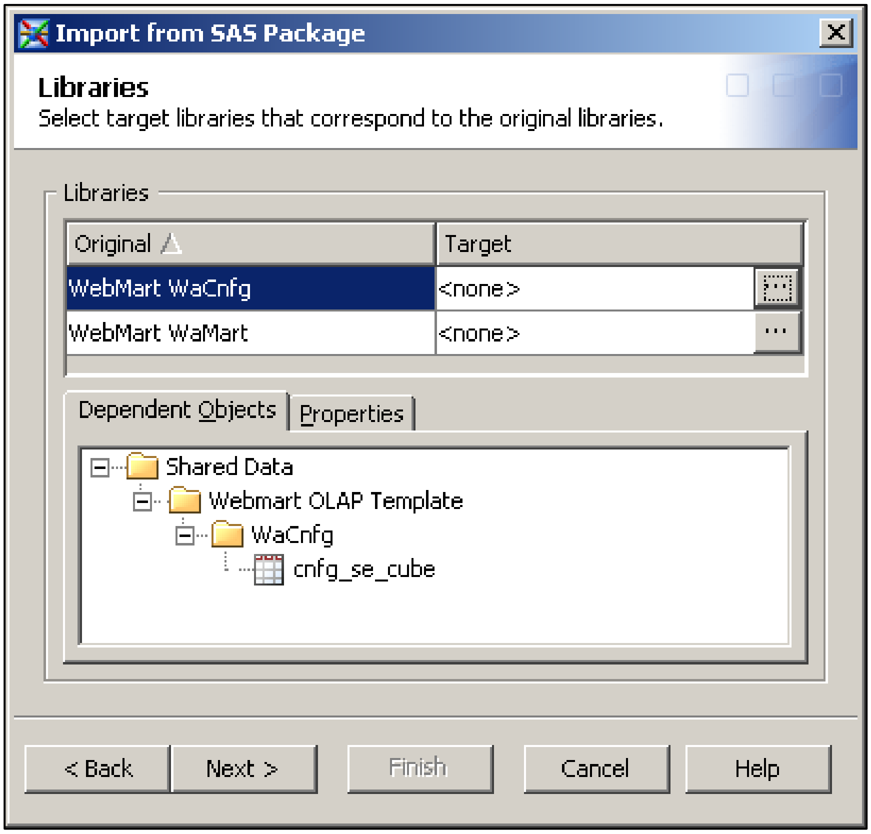
OLAP Schemas

Original	Target
SASApp - OLAP Schema	SASApp - OLAP Schema

7 Assign each of the source libraries in the import package to valid target libraries. Click the **...** button in the **Target** field to select a target library.

Caution:

Target libraries must be assigned correctly. If they are not assigned correctly, you might encounter errors that are difficult to correct.



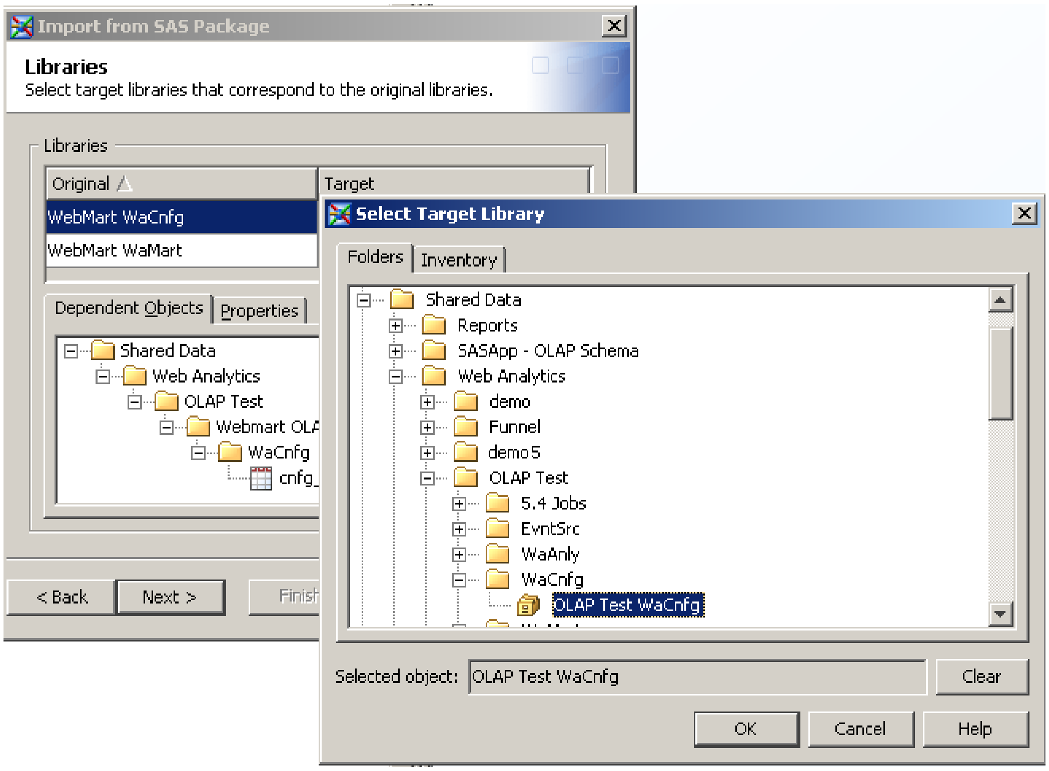
Import from SAS Package

Libraries

Select target libraries that correspond to the original libraries.

Original	Target
WebMart WaCnfg	<none>
WebMart WaMart	<none>

8 The Select Target Library dialog box appears. Select the same library as the source library.



Import from SAS Package

Libraries

Select target libraries that correspond to the original libraries.

Original	Target
WebMart WaCnfg	
WebMart WaMart	

Select Target Library

Selected object: OLAP Test WaCnfg

Map the source libraries as follows:

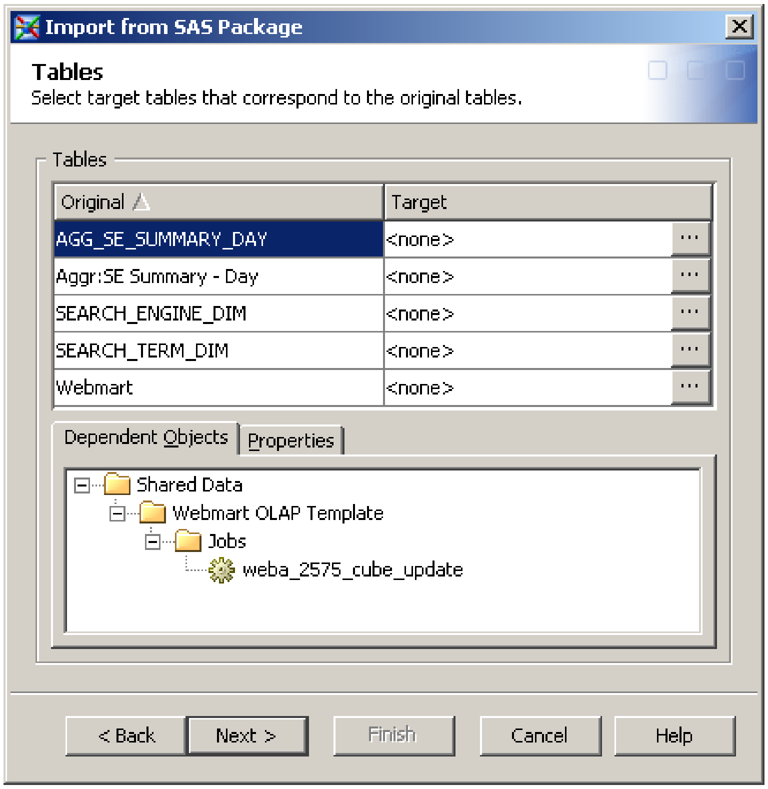
WebMart WaCnfg to <*Web mart*> **WaCnfg**
WebMart WaMart to <*Web mart*> **WaMart**

Click **Next**.

9 The import package contains five tables that must be mapped to a valid target location. Enter the target tables in the Tables dialog box in the wizard.

Caution:

Target tables must be assigned correctly. If they are not assigned correctly, you might encounter errors that are difficult to correct.



Import from SAS Package

Tables

Select target tables that correspond to the original tables.

Tables

Original	Target
AGG_SE_SUMMARY_DAY	<none>
Aggr:SE Summary - Day	<none>
SEARCH_ENGINE_DIM	<none>
SEARCH_TERM_DIM	<none>
Webmart	<none>

Dependent Objects | Properties

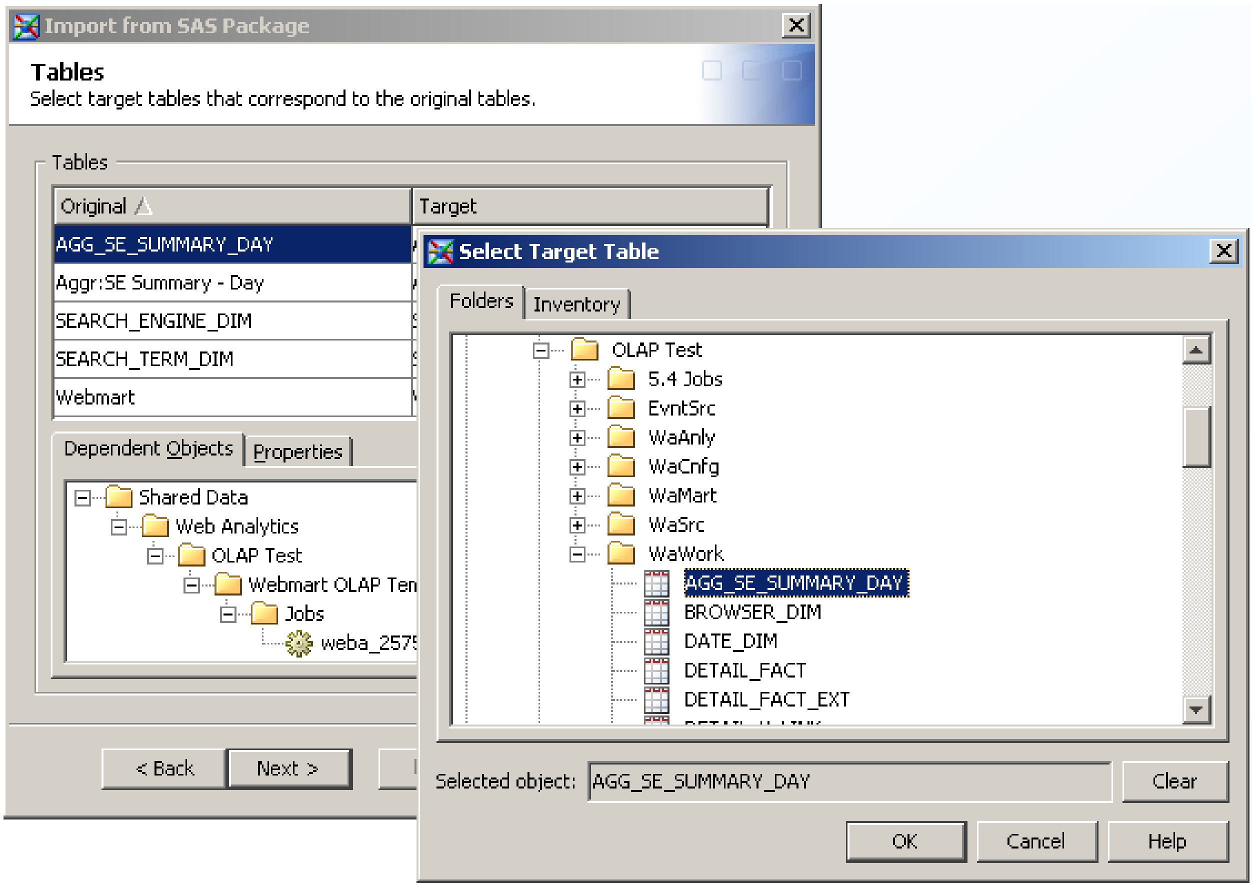
- Shared Data
  - Webmart OLAP Template
    - Jobs
      - weba_2575_cube_update

10 For each original table, click the **...** button in the **Target** field to select a target table. This action opens the Select Target Table dialog box. Select the corresponding table object.

Note: The default value for <*root folder path*> is **/Shared Data/Web Analytics**, and <*Web mart*> is the Web mart that will contain the OLAP objects.

Source Table	Target Library Folder	Target Table
Agg_SE_Summary_Day	<*Web mart root path*>/<*Web mart*>/WaWork	Agg_SE_Summary_Day
Aggr: SE_Summary - Day	< *Web mart root path*>/<*Web mart*>/WaMart	Aggr: SE_Summary - Day
Search_Engine_Dim	< *Web mart root path*>/<*Web mart*>/WaMart	Search_Engine_Dim
Search_Term_Dim	< *Web mart root path*>/<*Web mart*>/WaMart	Search_Term_Dim

Webmart	< *Web mart root path*>/<*Web mart*>/WaCnfg	Webmart



Import from SAS Package

Tables

Select target tables that correspond to the original tables.

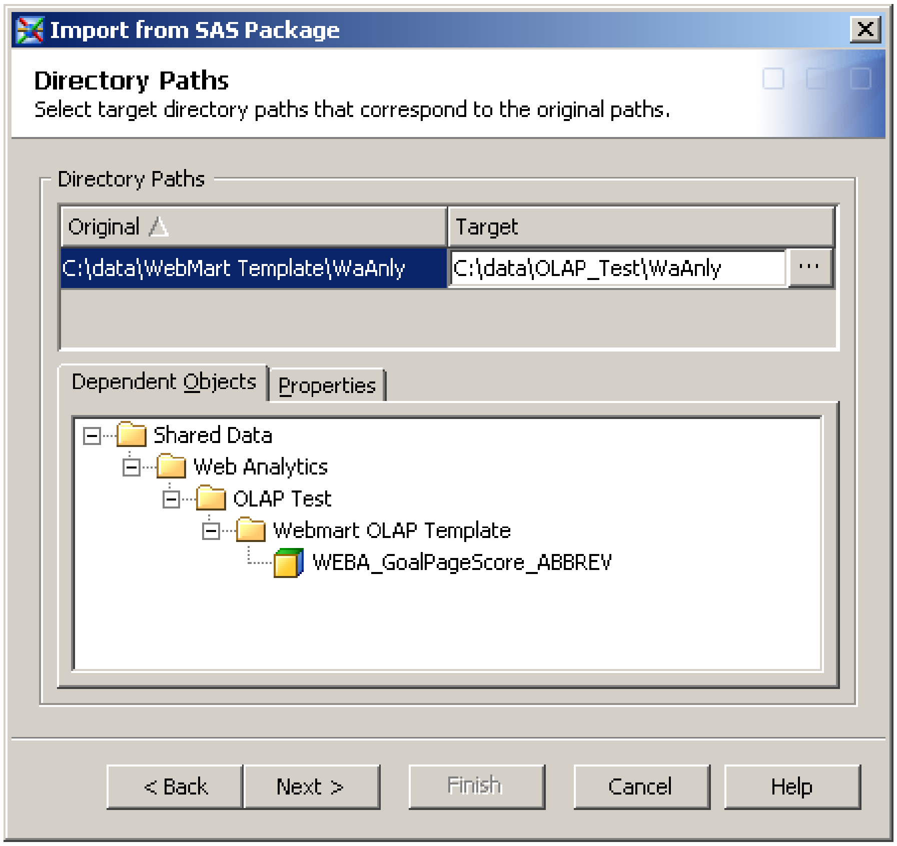
Original	Target
AGG_SE_SUMMARY_DAY	
Aggr:SE Summary - Day	
SEARCH_ENGINE_DIM	
SEARCH_TERM_DIM	
Webmart	

Select Target Table

Selected object: AGG_SE_SUMMARY_DAY

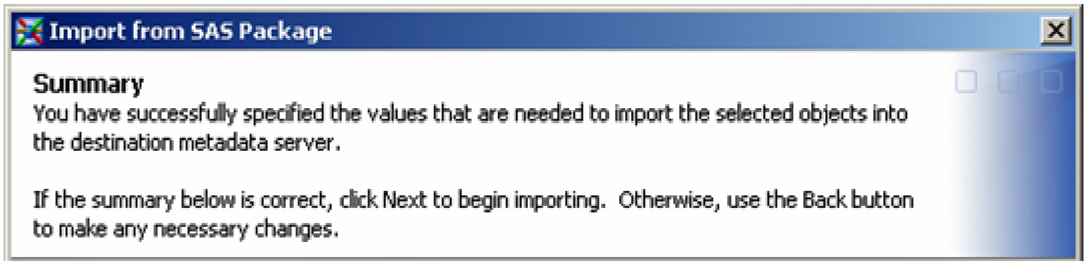
Click **Next**.

11 In the Directory Paths dialog box, select the **...** button in the **Target** field. Navigate to the location of the physical directory where the OLAP cubes will be stored. It is strongly recommended that you use the physical path to the WaAnly library, shown here:


Import from SAS Package
Directory Paths
Select target directory paths that correspond to the original paths.
Directory Paths

Original	Target
C:\data\WebMart Template\WaAnly	C:\data\OLAP_Test\WaAnly

Click **Next**. A summary dialog box appears where you can check your settings.



Summary

You have successfully specified the values that are needed to import the selected objects into the destination metadata server.

If the summary below is correct, click Next to begin importing. Otherwise, use the Back button to make any necessary changes.

If your settings are correct, click **Next** to complete the import process.

If the import is successful, a new folder called **Webmart OLAP Template** is created. It contains the OLAP cube and supporting objects.

Note: If the import is not successful, check the log for warnings or errors.

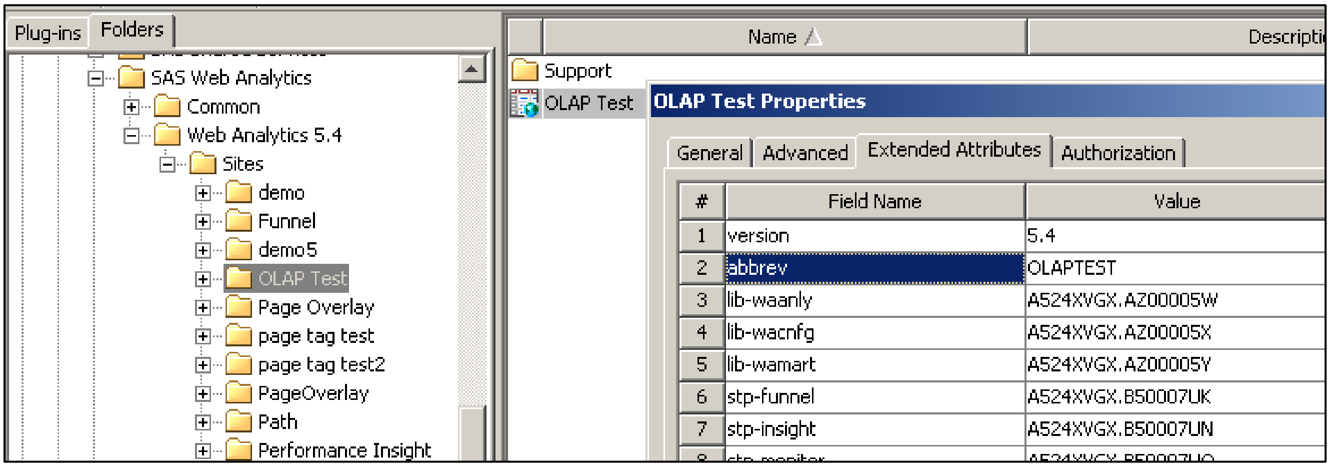
Customize the Cube

1 Open SAS Management Console. Click the **Folders** tab, and navigate to the **/System/Applications/SAS Web Analytics/SAS Web Analytics 5.4/ Site/**<*Web mart*> folder.

- ❑ In the right pane, select the <*Web mart*> object. Right-click on the Web mart object. Select **Properties**.
- ❑ Click the **Extended Attributes** tab. Note or copy the value of the **abbrev** field in row 2.

Caution:

Do not change any of the other field values on the Extended Attributes tab.



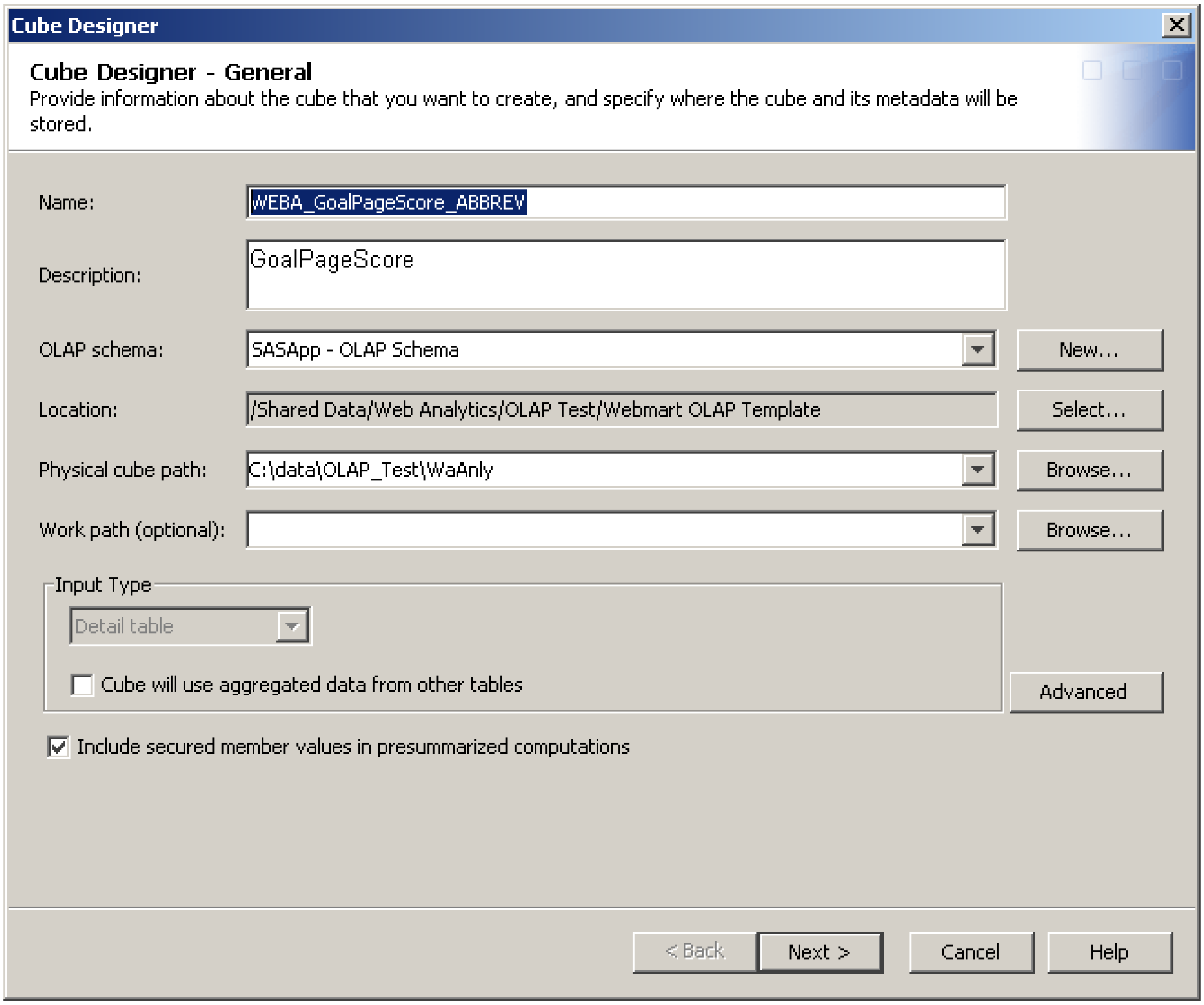
#	Field Name	Value
1	version	5.4
2	abbrev	OLAPTEST
3	lib-waanly	A524XVGX.AZ00005W
4	lib-wacnfg	A524XVGX.AZ00005X
5	lib-wamart	A524XVGX.AZ00005Y
6	stp-funnel	A524XVGX.B50007UK
7	stp-insight	A524XVGX.B50007UN
8	stp-monitor	A524XVGX.B50007UO

2 In SAS Data Integration Studio, navigate to the newly created **Webmart OLAP Template** folder. Right-click on the **WEBA_GoalPageScore_ABBREV** cube object and select **Edit Cube Structure**.

3 The **Cube Designer** dialog box appears. Change the cube name by replacing **ABBREV** in the cube name with the abbreviation that you noted in step 1.

Caution:

Do not change any other option on this tab.



Cube Designer

Cube Designer - General

Provide information about the cube that you want to create, and specify where the cube and its metadata will be stored.

Name: WEBA_GoalPageScore_ABBREV

Description: GoalPageScore

OLAP schema: SASApp - OLAP Schema

Location: /Shared Data/Web Analytics/OLAP Test/Webmart OLAP Template

Physical cube path: C:\data\OLAP_Test\WaAnly

Work path (optional):

Input Type

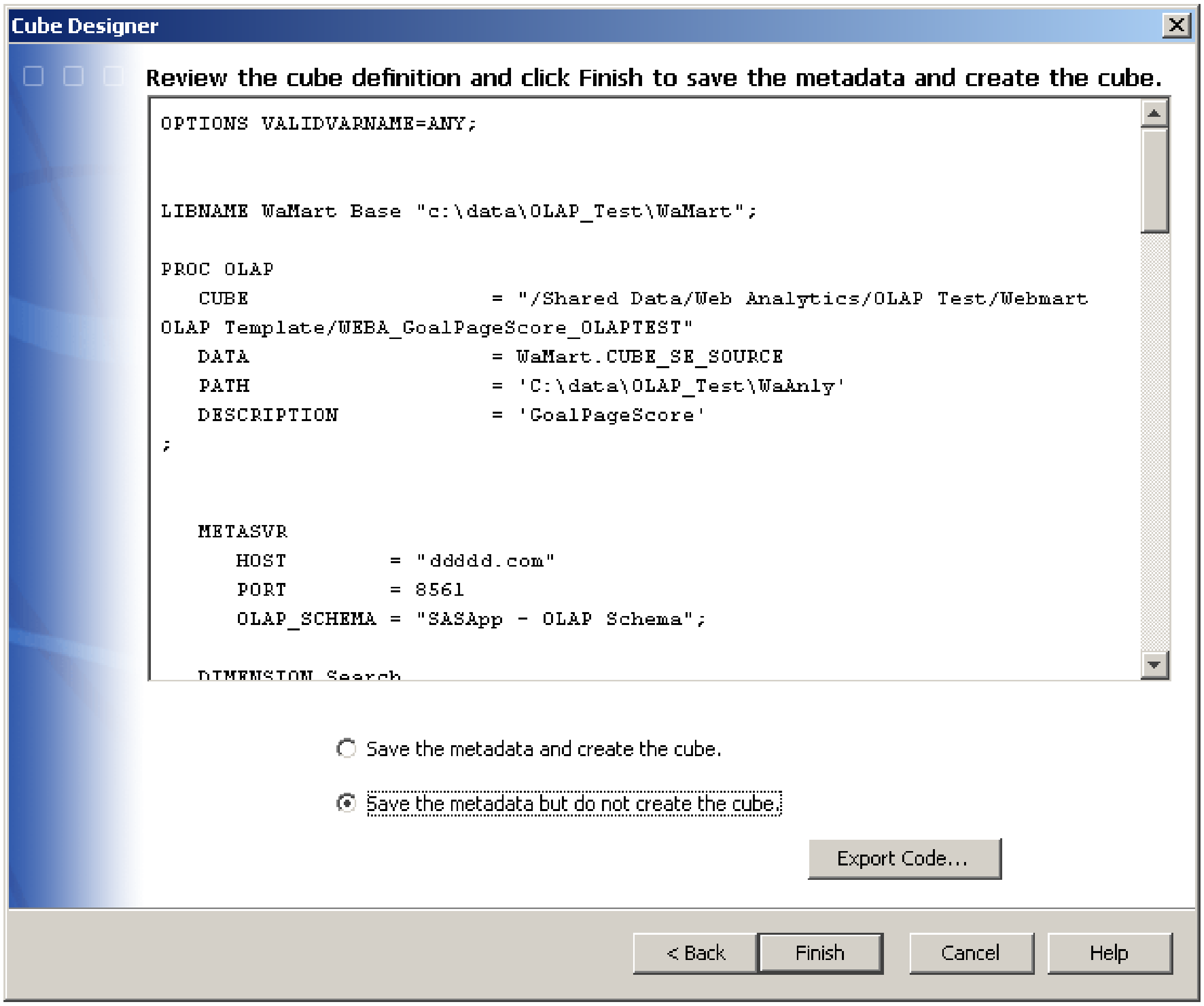
Detail table

□ Cube will use aggregated data from other tables

☑ Include secured member values in presummarized computations

Select **Next**.

4 Continue to select **Next** until the summary information is displayed. Select **Save the metadata but do not create the cube.** Click **Finish** to update the cube's metadata. The cube is now renamed and the OLAP cube and job objects have been re-registered within the metadata.



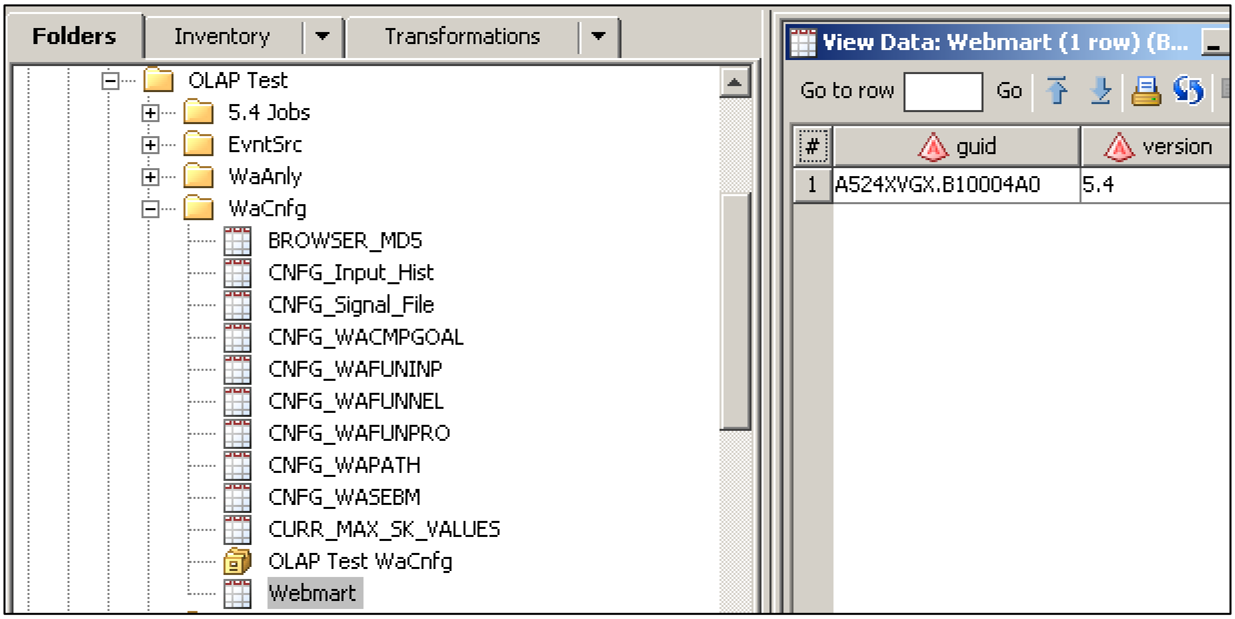
Review the cube definition and click Finish to save the metadata and create the cube.

```
OPTIONS VALIDVARNAME=ANY;


LIBNAME WaMart Base "c:\data\OLAP_Test\WaMart";

PROC OLAP
   CUBE                    = "/Shared Data/Web Analytics/OLAP Test/Webmart
OLAP Template/WEBA_GoalPageScore_OLAPTEST"
   DATA                    = WaMart.CUBE_SE_SOURCE
   PATH                    = 'C:\data\OLAP_Test\WaAnly'
   DESCRIPTION             = 'GoalPageScore'
;



   METASVR
      HOST        = "ddddd.com"
      PORT        = 8561
      OLAP_SCHEMA = "SASApp - OLAP Schema";

   DIMENSION Search
```

○ Save the metadata and create the cube.

● Save the metadata but do not create the cube.

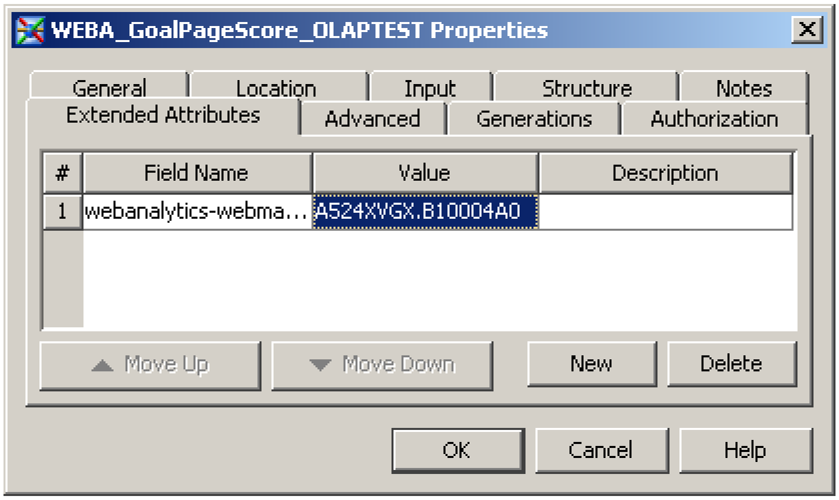
5 Record the GUID, a unique identifier for the Web mart. Navigate to <*Web mart root folder*>**/**<*Web mart*>**/WaCnfg**. Right-click on the **Webmart** table and select **Open**. Note the value in the GUID column.



- OLAP Test
  - 5.4 Jobs
  - EvntSrc
  - WaAnly
  - WaCnfg
    - BROWSER_MD5
    - CNFG_Input_Hist
    - CNFG_Signal_File
    - CNFG_WACMPGOAL
    - CNFG_WAFUNINP
    - CNFG_WAFUNNEL
    - CNFG_WAFUNPRO
    - CNFG_WAPATH
    - CNFG_WASEBM
    - CURR_MAX_SK_VALUES
    - OLAP Test WaCnfg
    - Webmart

#	guid	version
1	A524XVGX.B10004A0	5.4

6 Navigate to **WEBA_GoalPageScore**_<*abbrev*> cube object, where <*abbrev*> is the Web mart abbreviation. Right-click the cube and select **Properties**. Click the **Extended Attributes** tab. Enter the value of the **GUID** field (from step 5) in the **webanalytic-webmart-id** field.



#	Field Name	Value	Description
1	webanalytics-webma…	A524XVGX.B10004A0	

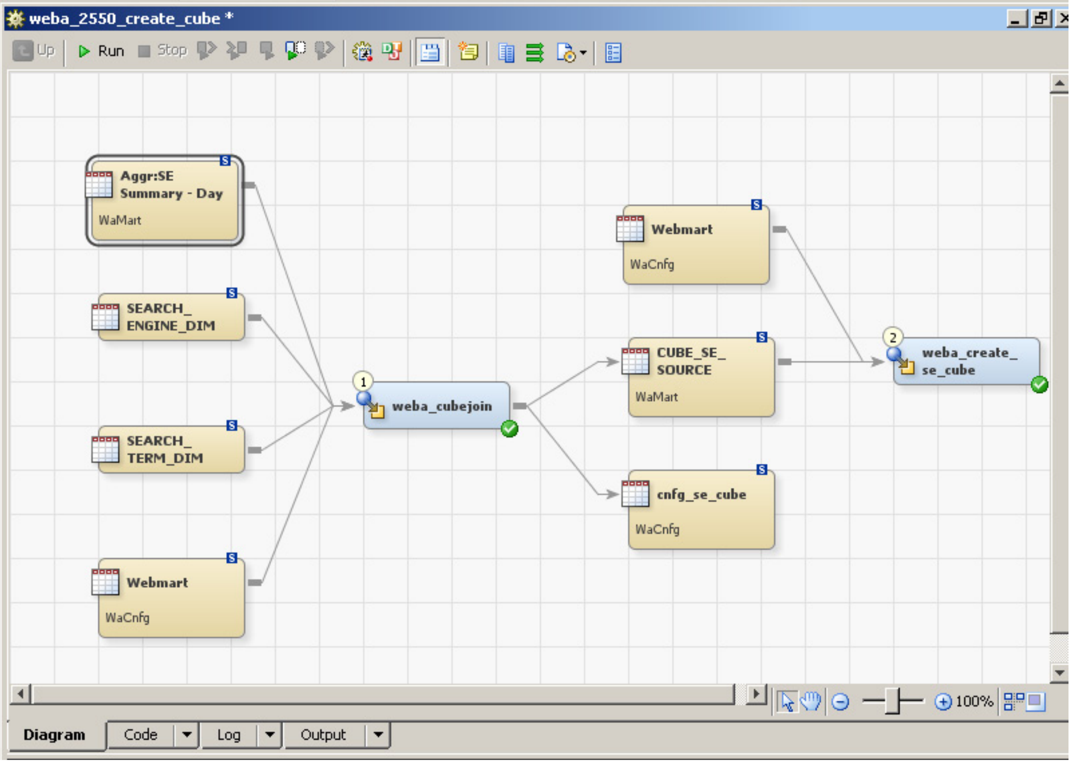
7 Navigate to the **WEBA_GoalPageScore**_<*abbrev*> cube object *job* object. Right-click the job and select **Properties**. Click the **Extended Attributes** tab. Enter the value of the **GUID** field (from step 5) in the **webanalytic-webmart-id** field.

8 Ensure that the source data requirements for the cube have been satisfied. In order for the cube to contain data, goal pages must have been selected with the Search Term Goals administrative interface within the SAS Web Analytics Web application. For more information, see the *SAS Web Analytics: User's Guide*.

Also, at least one day of data must have been loaded into the WaMart Aggr:SE Summary – Day aggregate table with the Weba_2100_Agg_Sebm job. For more information, see “Appendix 5, How the Aggregate Jobs Work.”

9 The Weba_2550_Create_Cube populates the **WEBA_GoalPageScore_**<*abbrev*> cube by using the WaMart Aggr:SE Summary – Day aggregate table.

Configure the Weba_2550_Create_Cube job. Open the job, and then right-click on the **weba_create_se_cube** transformation.



10 Run the job. If errors are encountered, perform these tasks:

a. Select **Properties**. Click the **Options** tab, and then click the **Metadata Rebuild** tab.

b. Enter the cube metadata path, and select the OLAP Server. Run the job. The cube now contains data.

c. Exit the job without saving.

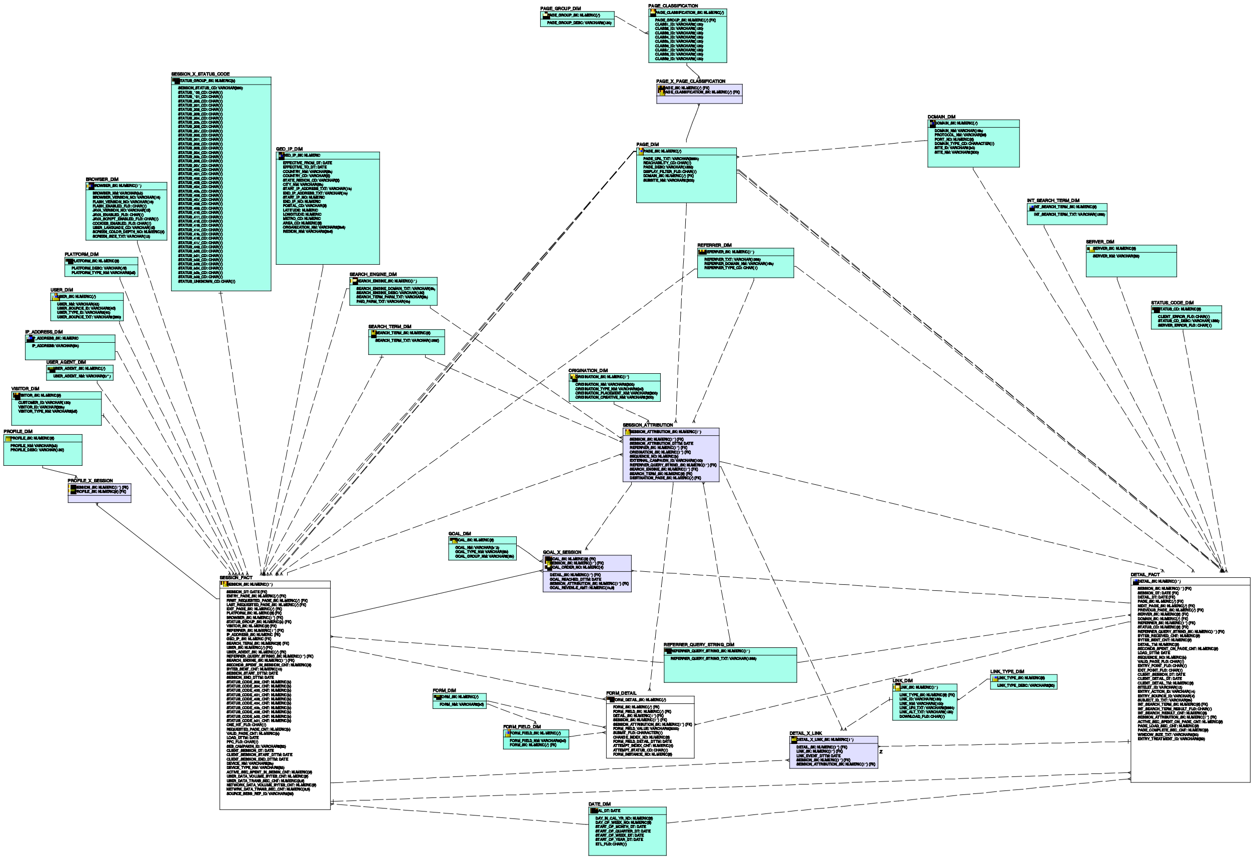
11 Schedule the 2575 job to run with each subsequent ETL.

APPENDIX 1

SAS Web Analytics Web Mart Aggregates

Web Mart Data Model

Figure A1.1: SAS Web Analytics Web Mart Data Model



Aggregates

Table A1.1: AGGR_ALL_PAGE

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
session_count	Num	8		
page_count	Num	8		
one_hit_session_count	Num	8		
entry_page_count	Num	8		
exit_page_count	Num	8		
prosp_bounce	Num	8		.
session_total	Num	8		.
page_total	Num	8		
page_sk	Num	7		
status_cd	Num	3		

Table A1.2: AGGR_BOUNCE

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
one_hit_session_count	Num	8		
prosp_bounce	Num	8		
session_count	Num	8		
page_sk	Num	7		

Table A1.3: AGGR_BROWSER

session_dt	day	Num	8	DATE9.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	
page_count		Num	8	NLNUM15.	.
session_sum		Num	8		
page_sum		Num	8		
browser_nm		Char	52		
browser_version_no		Char	16		

Table A1.4: AGGR_DAILY_TOTAL

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8	
repeat_visitors		Num	8	8	
page_view_med_session_cnt		Num	8	8	
page_view_low_session_cnt		Num	8	8	

Field	Label	Type	Length	Format	Informat
page_view_high_session_cnt		Num	8	8	
session_dt	day	Num	8	DATE9.	DATE9.
session_count		Num	8		
page_count		Num	8		
duration	duration	Num	8		
file_count		Num	8		
one_hit_session_count		Num	8		
prosp_bounce		Num	8		
day_of_week		Num	8		

Table A1.5: AGGR_ENTRY_PG

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
entry_page_count	Num	8		
session_count	Num	8		
page_count	Num	8		
one_hit_session_count	Num	8		
prosp_bounce	Num	8		
referrer_sk	Num	8		
page_sk	Num	7		

Table A1.6: AGGR_EXIT_PG

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
exit_page_count	Num	8		
page_sk	Num	7		

Table A1.7: AGGR_HOURLY_STATUS

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE9.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	
page_count		Num	8	NLNUM15.	
file_count	Hit Count	Num	8	NLNUM15.	
daily_sessions		Num	8		
hour		Num	8		
status_cd		Num	3		

Table A1.8: AGGR_HOURLY_TOTAL

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE9.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	NLNUM15
page_count		Num	8	NLNUM15.	NLNUM15
duration		Num	8		
file_count	Hit Count	Num	8	NLNUM15	NLNUM15
hour		Num	8		

Table A1.9: AGGR_INSIGHT

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8	
repeat_visitors		Num	8	8	
page_view_med_session_cnt		Num	8	8	
page_view_low_session_cnt		Num	8	8	
page_view_high_session_cnt		Num	8	8	
session_dt	day	Num	8	DATE9.	DATE9.
session_count		Num	8		
page_count		Num	8		
status_code_302_count	status_code_302_count	Num	8		
status_code_304_count	status_code_304_count	Num	8		
status_code_400_count	status_code_400_count	Num	8		
status_code_401_count	status_code_401_count	Num	8		
status_code_403_count	status_code_403_count	Num	8		
status_code_404_count	status_code_404_count	Num	8		
status_code_405_count	status_code_405_count	Num	8		
status_code_408_count	status_code_408_count	Num	8		
status_code_500_count	status_code_500_count	Num	8		
status_code_501_count	status_code_501_count	Num	8		
duration		Num	8		
file_count		Num	8		
one_hit_session_count		Num	8		
total_bytes_sent		Num	8		
daily_sessions		Num	8		

Table A1.10: AGGR_INSIGHT_ST

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE9.	DATE9.
session_count		Num	8		
session_total		Num	8		
search_term_sk		Num	8		
search_engine_sk		Num	8		
referrer_sk		Num	8		

Table A1.11: AGGR_INT_SEARCH

Field	Label	Type	Length	Format	Informat
session_dt	session_dt	Num	8	DATE9.	
page_count	nPAGE_COUNT	Num	8		
exit_count	nEXIT_COUNT	Num	8		
total_search_cnt	nTOTAL_SEARCH_CNT	Num	8		
conv_cnt	nCONV_CNT	Num	8		
unique_cnt	nUNIQUE_CNT	Num	8		
refine_cnt	nREFINE_CNT	Num	8		
int_search_term_sk		Num	8		

Table A1.12: AGGR_PLATFORM

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE9.	DATE9.
session_count		Num	8	NLNUM15.	
page_count		Num	8	NLNUM15.	
session_sum		Num	8		
page_sum		Num	8		
platform_desc		Char	78	.	

Table A1.13: AGGR_SEBM

Field	Type	Length	Format	Informat
pg_st_sessions	Num	8		
goal	Num	8		
ref_st_cnt	Num	8		
session_count	Num	8		
session_dt	Num	8	DATE9.	DATE9.
page_sk	Num	7		
search_term_sk	Num	6		
ppc_flg	Char	1		

Table A1.14: AGGR_SEB_CAMPAIGN

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
goal_visits	Num	8		
search_engine_sk	Num	8		
seb_campaign_id	Char	32		
page_sk	Num	8		

Table A1.15: AGGR_SUMMARY

Field	Label	Type	Length	Format	Informat
pg_st_sessions		Num	8		
ref_st_cnt		Num	8		
cs_gv		Num	8		
tv_gp		Num	8		
session_count		Num	8		
session_dt	Day	Num	8	DATE9.	DATE9.
search_engine_sk		Num	8		
ppc_flg		Char	1		
search_term_sk		Num	6		

Table A1.16: AGGR_STATUS

Field	Type	Length	Format	Informat
session_dt	Num	8	DATE9.	DATE9.
session_count	Num	8		
page_count	Num	8		
session_total	Num	8		
status_cd	Num	3		
referrer_sk	Num	8		
page_sk	Num	7		

Table A1.17: AGGR_TERM

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE9.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	
session_total		Num	8	.	
search_term_sk		Num	8		
search_engine_sk		Num	8		

Table A1.18: AGGR_VISITOR

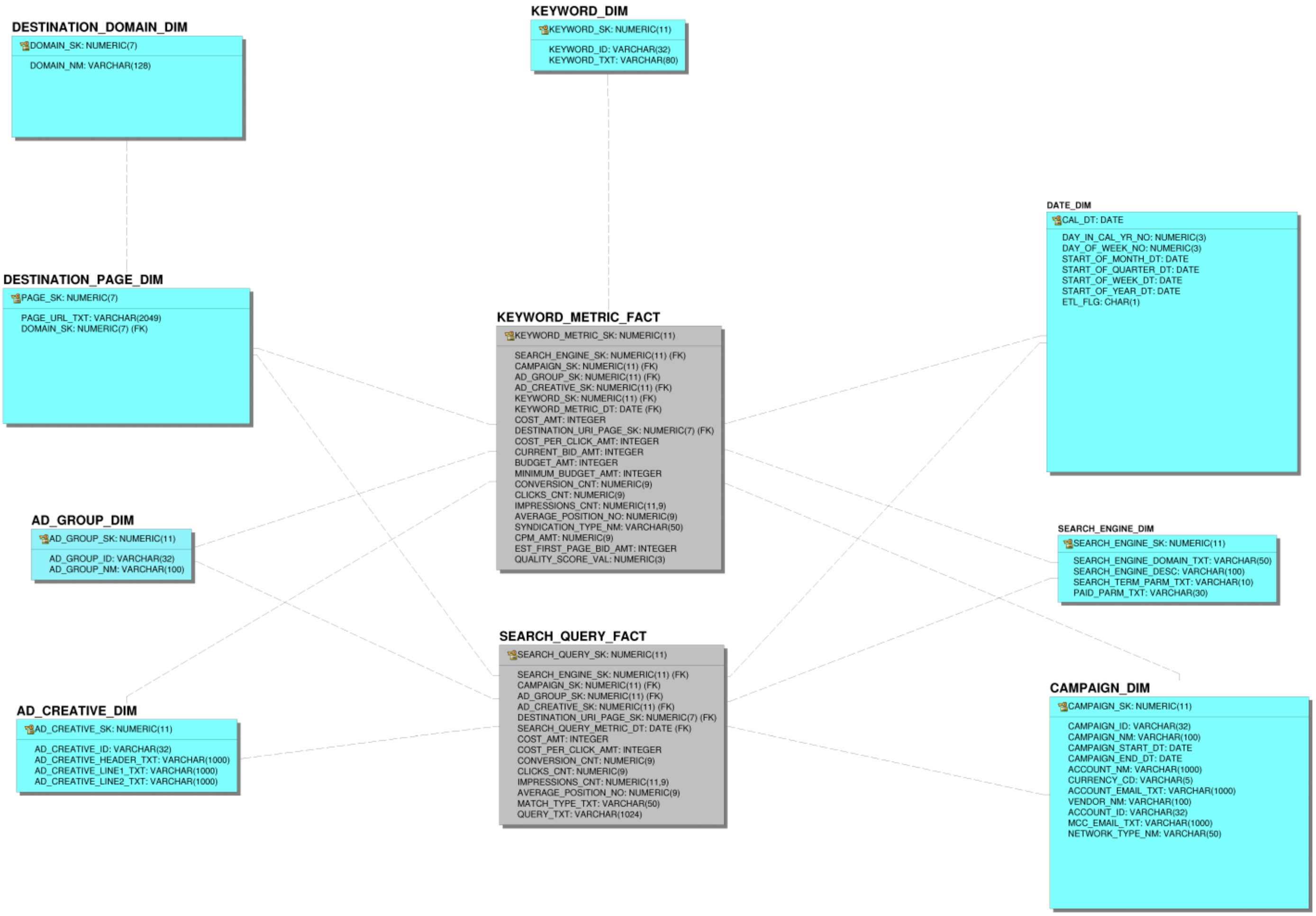
Field	Type	Length	Format	Informat
session_dt	Num	8	DATE7.	DATE9.
session_count	Num	8	NLNUM15.	
page_count	Num	8	NLNUM15.	
duration	Num	8	NLNUM15.	
page_view_los_session_cnt	Num	8	NLNUM15.	
page_view_med_session_cnt	Num	8	NLNUM15.	
page_view_high_session_cnt	Num	8	NLNUM15.	
visitor_sk	Num	8		
start_dttm	Num	8	DATETIME20.	
end_dttm	Num	8	DATETIME20.	NLDATA21.
day_of_week	Num	8	DATETIME20.	NLDATA21.

APPENDIX 2

SAS Web Analytics SEBD Mart

SEBD Mart Data Model

Figure A2.1: SAS Web Analytics SEBD Mart Data Model


DESTINATION_DOMAIN_DIM
DOMAIN_SK: NUMERIC(7)
DOMAIN_NM: VARCHAR(128)
KEYWORD_DIM
KEYWORD_SK: NUMERIC(11)
KEYWORD_ID: VARCHAR(32)
KEYWORD_TXT: VARCHAR(80)
DATE_DIM
CAL_DT: DATE
DAY_IN_CAL_YR_NO: NUMERIC(3)
DAY_OF_WEEK_NO: NUMERIC(3)
START_OF_MONTH_DT: DATE
START_OF_QUARTER_DT: DATE
START_OF_WEEK_DT: DATE
START_OF_YEAR_DT: DATE
ETL_FLG: CHAR(1)
DESTINATION_PAGE_DIM
PAGE_SK: NUMERIC(7)
PAGE_URL_TXT: VARCHAR(2049)
DOMAIN_SK: NUMERIC(7) (FK)
KEYWORD_METRIC_FACT
KEYWORD_METRIC_SK: NUMERIC(11)
SEARCH_ENGINE_SK: NUMERIC(11) (FK)
CAMPAIGN_SK: NUMERIC(11) (FK)
AD_GROUP_SK: NUMERIC(11) (FK)
AD_CREATIVE_SK: NUMERIC(11) (FK)
KEYWORD_SK: NUMERIC(11) (FK)
KEYWORD_METRIC_DT: DATE (FK)
COST_AMT: INTEGER
DESTINATION_URI_PAGE_SK: NUMERIC(7) (FK)
COST_PER_CLICK_AMT: INTEGER
CURRENT_BID_AMT: INTEGER
BUDGET_AMT: INTEGER
MINIMUM_BUDGET_AMT: INTEGER
CONVERSION_CNT: NUMERIC(9)
CLICKS_CNT: NUMERIC(9)
IMPRESSIONS_CNT: NUMERIC(11,9)
AVERAGE_POSITION_NO: NUMERIC(9)
SYNDICATION_TYPE_NM: VARCHAR(50)
CPM_AMT: NUMERIC(9)
EST_FIRST_PAGE_BID_AMT: INTEGER
QUALITY_SCORE_VAL: NUMERIC(3)
AD_GROUP_DIM
AD_GROUP_SK: NUMERIC(11)
AD_GROUP_ID: VARCHAR(32)
AD_GROUP_NM: VARCHAR(100)
SEARCH_ENGINE_DIM
SEARCH_ENGINE_SK: NUMERIC(11)
SEARCH_ENGINE_DOMAIN_TXT: VARCHAR(50)
SEARCH_ENGINE_DESC: VARCHAR(100)
SEARCH_TERM_PARM_TXT: VARCHAR(10)
PAID_PARM_TXT: VARCHAR(30)
SEARCH_QUERY_FACT
SEARCH_QUERY_SK: NUMERIC(11)
SEARCH_ENGINE_SK: NUMERIC(11) (FK)
CAMPAIGN_SK: NUMERIC(11) (FK)
AD_GROUP_SK: NUMERIC(11) (FK)
AD_CREATIVE_SK: NUMERIC(11) (FK)
DESTINATION_URI_PAGE_SK: NUMERIC(7) (FK)
SEARCH_QUERY_METRIC_DT: DATE (FK)
COST_AMT: INTEGER
COST_PER_CLICK_AMT: INTEGER
CONVERSION_CNT: NUMERIC(9)
CLICKS_CNT: NUMERIC(9)
IMPRESSIONS_CNT: NUMERIC(11,9)
AVERAGE_POSITION_NO: NUMERIC(9)
MATCH_TYPE_TXT: VARCHAR(50)
QUERY_TXT: VARCHAR(1024)
CAMPAIGN_DIM
CAMPAIGN_SK: NUMERIC(11)
CAMPAIGN_ID: VARCHAR(32)
CAMPAIGN_NM: VARCHAR(100)
CAMPAIGN_START_DT: DATE
CAMPAIGN_END_DT: DATE
ACCOUNT_NM: VARCHAR(1000)
CURRENCY_CD: VARCHAR(5)
ACCOUNT_EMAIL_TXT: VARCHAR(1000)
VENDOR_NM: VARCHAR(100)
ACCOUNT_ID: VARCHAR(32)
MCC_EMAIL_TXT: VARCHAR(1000)
NETWORK_TYPE_NM: VARCHAR(50)
AD_CREATIVE_DIM
AD_CREATIVE_SK: NUMERIC(11)
AD_CREATIVE_ID: VARCHAR(32)
AD_CREATIVE_HEADER_TXT: VARCHAR(1000)
AD_CREATIVE_LINE1_TXT: VARCHAR(1000)
AD_CREATIVE_LINE2_TXT: VARCHAR(1000)

SEBD Aggregates

AGGR_SEBD_KEYWORD_DAY Table

SEBDMART tables used

- ❑ SEBDMART.KEWORD_METRIC_FACT
- ❑ SEBDMART.CAMPAIGN_DIM

Fields

Table A2.1: AGGR_SEBD_KEYWORD_DAY Table Fields

Field	Type	Length	Informat	Format	Description
campaign_sk	Num	8	12.	BEST12.	campaign surrogate key; use to join to the CAMPAIGN_DIM table
keyword_sk	Num	8	12.	BEST12.	keyword surrogate key; use to join to the KEYWORD_DIM table
currency_cd	Char	5	$5.	$5.	ISO currency code associated with a specific campaign
keyword_metric_dt	Num	8	DATE9.	DATE9.	response date for campaign and keyword
impressions_cnt	Num	8			number of impressions displayed during a day
clicks_cnt	Num	8	12.	BEST12.	number of click-throughs during a day
cost_amt	Num	8	12.	BEST12.	amount that was paid for the click-throughs on a given day in the currency specified by the currency code
budget_amt	Num	8	12.	BEST12.	maximum budgeted bid amount for a keyword within a campaign on a day
current_bid_amt	Num	8	12.	BEST12.	maximum amount bid for a keyword within a campaign on a day
average_position_no	Num	8	12.	BEST12.	average position of all ads for a keyword and campaign combination for a day

ANLY_SEBD_KEYWORD_DAY Table

SEBDMART tables used

- ❑ SEBDMART.KEWORD_METRIC_FACT
- ❑ SEBDMART.CAMPAIGN_DIM
- ❑ SEBDMART.KEYWORD_DIM

Summary level

campaign_nm
keyword_txt currency_cd
keyword_metric_dt

Fields

Table A2.2: ANLY_SEBD_KEYWORD_DAY Table Fields

Field	Type	Length	Informat	Format	Description
campaign_nm	Char	130	$130	$130	campaign name
keyword_txt	Char	104	$104	$104	keyword text
currency_cd	Char	5	$5	$5	ISO currency code associated with a specific campaign
keyword_metric_dt	Num	8	DATE9.	DATE9.	response date for campaign and keyword
impressions_cnt	Num	8	NLNUM11.9	NLNUM11.9	number of impressions displayed during a day
clicks_cnt	Num	8	12.	BEST12.	number of click-throughs during a day
cost_amt	Num	8	12.	BEST12.	amount that was paid for the click-throughs on a given day in the currency specified by the currency code
budget_amt	Num	8	12.	BEST12.	maximum budgeted bid amount for a keyword within a campaign on a day
current_bid_amt	Num	8	12.	BEST12.	maximum amount bid for a keyword within a campaign on a day
average_position_no	Num	8	12.	BEST12.	average position of all ads for a keyword and campaign combination for a day
cum_impressions	Num	8	12.	BEST12.	cumulative impressions for a campaign and keyword
cum_clicks	Num	8	12.	BEST12.	cumulative click-throughs for a campaign and keyword
ctr	Num	8	12.	BEST12.	Click-through rate = clicks_cnt / impressions_cnt
cum_ctr	Num	8	12.	BEST12.	cumulative click-through rate = cum_clicks / cum_impressions

AGGR_SEBD_CAMPAIGN_AD_SUMMARY Table

SEBDMART tables used

❑ SEBDMART.KEYWORD_METRIC_FACT

❑ SEBDMART.CAMPAIGN_DIM

Fields

Table A2.3: AGGR_SEBD_CAMPAIGN_AD_SUMMARY Table Fields

Field	Type	Length	Informat	Format	Description
campaign_sk	Num	8	12.	BEST12.	campaign surrogate key
ad_creative_sk	Num	8	12.	BEST12.	ad creative surrogate key
currency_cd	Char	8	$8	$8	ISO currency code
month	Num	8	NLDATE20.	NLDAT20.	first day of month
cost_amt	Num	8	12.	BEST12.	cost
conversion_cnt	Num	8	12.	BEST12.	conversions
clicks_cnt	Num	8	12.	BEST12.	number of click-throughs
impressions_cnt	Num	8	NLNUM11.9	NLNUM11.9	impressions
cpm_amt	Num	8	12.	BEST12.	cost per thousand impressions

AGGR_SEB_CAMPAIGN_DAY Table

Tables used from the WAMART data mart

- WAMART.SESSION_FACT
- WAMART.DETAIL_FACT

Fields

Table A2.4: AGGR_SEB_CAMPAIGN_DAY Table Fields

Field	Type	Length	Informat	Format	Table	Description
seb_campaign_id	Char	32	$32.	$32.	worklib.session_fact	Vendor campaign ID (natural key).
search_engine_sk	Num	8	12.	BEST12.	worklib.session_fact	Search engine associated with the seb_campaign_id (Vendor).
page_sk	Num	8	12.	BEST12.	worklib.detail_fact, wacnfg.cnfg_wasebm	Goal page - a special page that has significant meaning where a customer is trying to drive offline traffic to. A goal page is specified using the goal page administrator, which updates wacnfg.cnfg_wasebm.
session_dt	Num	8	DATE9.	DATE9.	worklib.session_fact	The date visits (sessions) occurred.
goal_visits	Num	8	12.	BEST12.	worklib.session_fact, worklib.detail_fact, and wacnfg.cnfg_wasebm	Number of visits where visitors entered the site through an SEB campaign (ppc flg=1) and a goal page was viewed for a particular day.

Notes:

- If there are no goal pages, the aggregate will be empty.
- If goal pages are present, SEB_CAMPAIGN_ID will be summarized, including missing values.

APPENDIX 3

Google Data Source for Field Mappings

Extract Tables

Table A3.1: GOOGLE_ACCT

Source Field	Type	Length	Used?	Transformation
ClientEmail	Char	255	Y	
CurrencyCode	Char	255	Y	
CustomerId	Char	255	Y	Add a 'G' as a prefix.
DescriptiveName	Char	255	Y	
Email	Char	255	Y	

Table A3.2: GOOGLE_CAMPAIGN

Source Field	Type	Length	Used?	Transformations
Campaign	Char	255	Y	
CampaignId	Char	32	Y	Add a 'G' as a prefix.
CampaignStatus	Char	255	N	
CustomerId	Char	255	Y	Add a 'G' as a prefix.
EndDay	Num	8	Y	
NetworkTypes	Char	255	Y	
StartDay	Num	8	Y	

Table A3.3: GOOGLE_AD

Source Field	Type	Length	Used?	Notes
AdGroup	Char	255	Y	
AdGroupId	Char	32	Y	Add a 'G' as a prefix.
AdWordsType	Char	255	Y	
AveragePosition	Num	8	Y	Converted from character to numeric.
CPC	Num	8	Y	Converted from character to numeric.
CPM	Num	8	Y	Check for presence in extract data sets. If it is not present, then set it to missing. Otherwise, convert it to numeric.
Campaign	Char	255	N	
CampaignId	Char	32	Y	Add a 'G' as a prefix.
Clicks	Num	8	Y	Converted from character to numeric.
ConversionRate	Num	8	N	Converted from character to numeric.
Conversions	Num	8	N	Converted from character to numeric.
Cost	Num	8	Y	Converted from character to numeric.
CostPerConversion	Num	8	N	Converted from character to numeric.
CostPerTransaction	Num	8	N	Converted from character to numeric.
CreativeId	Char	32	Y	Add a 'G' as a prefix.
DailyBudget	Num	8	Y	Converted from character to numeric.
DescriptionLine1	Char	255	Y	
DescriptionLine2	Char	255	Y	
DescriptionLine3	Char	255	Y	
Dest_domain	Char	128	Y	Created from DestinationURL. Domain is the first string between http(s):// and / within the URL. An example of DestinationURL is http://www.sas.com/apps/sim/redirect.jsp?detail= TR1044, where the domain is www.sas.com.
Dest_page	Char	2049	Y	Created from DestinationURL. The http(s)://domain is stripped from the string. An example of DestinationURL is http://www.sas.com/apps/sim/redirect.jsp?detail= TR1044, where the page is /apps/sim/redirect.jsp?detail=TR1044.
Impressions	Num	8	Y	Converted from character to numeric.
KeywordId	Char	32	Y	Add a 'G' as a prefix.
MaxContentCPC	Num	8	N	Converted from character to numeric.
MaximumCPC	Num	8	Y	Converted from character to numeric.
MaximumCPM	Num	8	Y	Check for presence in extract data sets. If it is not present, then set it to missing. Otherwise, convert it to numeric.
keyword	Char	255	Y	
process_dttm	Num	8	Y	Datetime() when the row is processed into google_ad. Used to eliminate duplicate data when the same file is read in twice.

Source Field	Type	Length	Used?	Notes
responsedate	Num	8	Y	Date of an impression and click for a given campaign, adgroup, and ad and keyword combination.
search_engine	Char	8	Y	Created static value of 'google'.
FirstPageCpc	Num	8	Y	Added in V13.
QualityScore	Num	3	Y	Added in V13.

Table A3.4: GOOGLE_SQ

Source Field	Type	Length	Used?	Notes
creativeid	Char	2048	Y	Add a 'G' as a prefix.
Query	Char	1024	Y	Search queries with Google that triggered the creative.
AdGroupId	Char	255	Y	Add a 'G' as a prefix.
AveragePosition	Num	8	Y	Converted from character to numeric.
CPC	Num	8	Y	Converted from character to numeric.
CTR	Num	8	Y	Converted from character to numeric.
CampaignId	Char	255	Y	Add a 'G' as a prefix.
Clicks	Num	8	Y	Converted from character to numeric.
ConversionRate	Num	8	N	Converted from character to numeric.
Conversions	Num	8	Y	Converted from character to numeric.
Cost	Num	8	Y	Converted from character to numeric.
DestDomain	Char	128	Y	Created from DestinationURL. Domain is the first string between http(s):// and / within the URL. An example of DestinationURL is http://www.sas.com/apps/sim/redirect.jsp?detail=TR1044, where the domain is www.sas.com.
Dest_page	Char	2049	Y	Created from DestinationURL. The http(s)://domain is stripped from the string. An example of DestinationURL is http://www.sas.com/apps/sim/redirect.jsp?detail=TR1044, where the page is /apps/sim/redirect.jsp?detail=TR1044.
Impressions	Num	8	Y	Converted from character to numeric.
MatchType	Char	255	Y	
ResponseDate	Num	8	Y	
Search_engine	Char	8	Y	
Process_dttm	Num	8	N	
AdWordsType	Char	255	Y	

Table Mappings

Table A3.5: Destination_Domain_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Domain_ SK	Num	7	Surrogate key. Value is a sequential number.
SEBDWORK.GOOGLE_AD	Dest_ Domain	Char	128	Extracted from DestinationURL. Use unique values of Dest_Domain to populate the field.	Domain_ Nm	Varchar	128	Natural key. Domain name is normalized by lowercasing and compressing blanks.

Table A3.6: Destination_Page_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Page_SK	Num	7	Surrogate key. Value is a sequential number.
SEBDWORK. GOOGLE_AD	Dest_ Page	Char	2049	Extracted from DestinationURL. Use unique values of Dest_Page to populate field.	Page_ URL_TXT	Varchar	2049	Natural key. Normalized page string stored (lowercased and compbl).
Destination_ Domain_Dim	Domain_ Sk	Num	7	Uses Dest_Domain in SEBDWORK. GOOGLE_AD. Value needs to be normalized - lowercased and extra spaces compressed (compbl).	Domain_ sk	Num	7	Foreign key. Domain_sk is determined by matching Dest_Domain with Domain_Nm within Destination_Domain_ Dim.

Table A3.7: Campaign_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Campaign_ SK	Num	8	Surrogate key. Value is a sequential number.
SEBDWORK. GOOGLE_ CAMPAIGN	CampaignId	Char	32	No transformation.	Campaign_ Id	Varchar	32	Natural key. Google Adwords campaign ID.
SEBDWORK. GOOGLE_ CAMPAIGN	Campaign	Char	255	Value is stripped of leading and trailing blanks.	Campaign_ nm	Varchar	100	Should be identical to the name that is associated with the campaign ID in ad report. Field is updated with the value in GOOGLE_CAMPAIGN.
SEBDWORK. GOOGLE_ CAMPAIGN	StartDay	Date	8	Google send date is in yyyy-mm-dd (2004-04-02) form when extract XML is sent. Extract application converts the value to a SAS date format.	Campaign_ Start_Dt	Num	8	Date that the campaign started. Field is updated with the value in GOOGLE_CAMPAIGN.
SEBDWORK. GOOGLE_ CAMPAIGN	EndDay	Date	8	Google send date is in yyyy-mm-dd (2004-04-02) form when extract XML is sent. Extract application converts the value to a SAS date format.	Campaign_ End_Dt	Num	8	Date that the campaign either ended or will end. Field is updated with the value in GOOGLE_CAMPAIGN.
SEBDWORK. GOOGLE_ ACCT	Descriptive Name	Char	255	Name of Google account. Account name is associated with a campaign via the customerid in the campaign and account extract tables.	Account_ Nm	Varchar	1000	Field is updated with the value in GOOGLE_ACCT.

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
SEBDWORK. GOOGLE_ ACCT	CurrencyCode	Char	255	Value is compressed to remove spaces. Attribute of the account and not of the campaign.	Currency_ CD	Varchar	5	Google notes. The currency in which this account operates and the value if from a subset of the currency codes derived from the ISO 4217 standard. See Currency Codes for the currently supported currencies. Field is updated with the value in GOOGLE_ACCT.
SEBDWORK. GOOGLE_ ACCT	ClientEmail	Char	255	Used as is.	Account_ Email_ Txt	Varchar	1000	User ID for the account when logging in to the API. Field is updated with value in GOOGLE_ACCT.
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Vendor_ Nm	Varchar	100	Static value - 'GOOGLE_ADWORD S'
SEBDWORK. GOOGLE_ ACCT	CustomerId	Char	255	Used as is.	MCC_ Email_Txt	Varchar	1000	MCC user ID for the account when logging on to the API. Field is updated with value in GOOGLE_ACCT.
SEBDWORK. GOOGLE_ ACCT	Account_Id	Char	255	Stripped of blanks.	Account_ Id	Varchar	32	Google customer ID that is associated with the client e-mail. Field is updated with the value in GOOGLE_ACCT.
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Network_T ype_Nm	Char	50	

Table A3.8: Ad_Group_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Ad_Group_SK	Num	8	Surrogate key. Value is a sequential number.
SEBDWORK.GOOGLE_AD	AdGroup Id	Char	255	Use unique values of compressed field.	Ad_Group_Id	Varchar	32	Natural key.
SEBDWORK.GOOGLE_AD	AdGroup	Char	255	Use unique values of stripped field.	Ad_Group_Nm	Varchar	100	Value in GOOGLE_AD is used to update value within Ad_Group_Dim

Table A3.9: Ad_Creative_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Ad_Creative_SK	Num	8	Surrogate key. Value is a sequential number.
SEBDWORK.GOOGLE_AD	CreativeId	Char	255	Use unique values of compressed field.	Ad_Creative_Id	Varchar	32	Natural key.
SEBDWORK.GOOGLE_AD	Description Line1	Char	255	Use as is.	Ad_Creative_Header_Txt	Varchar	1000	Value in GOOGLE_AD is used to update the value in Ad_Creative_Dim.
SEBDWORK.GOOGLE_AD	Description Line2	Char	255	Use as is.	Ad_Creative_Line1_Txt	Varchar	1000	Value in GOOGLE_AD is used to update the value in Ad_Creative_Dim.
SEBDWORK.GOOGLE_AD	Description Line3	Char	255	Use as is.	Ad_Creative_Line2_Txt	Varchar	1000	Value in GOOGLE_AD is used to update the value in Ad_Creative_Dim.

Table A3.10: Keyword_Dim

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Keyword_SK	Num	8	Surrogate key. Value is a sequential number.
SEBDWORK.GOOGLE_AD	KeywordId	Char	255	Use unique values of compressed field.	Keyword_Id	Varchar	32	Natural key.
SEBDWORK.GOOGLE_AD	Keyword	Char	255	Use unique values of stripped field.	Keyword_Txt	Varchar	80	Value in GOOGLE_AD is used to update the value in Keyword_Dim.

Table A3.11: Keyword_Metric_Fact

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	Keyword_ metric_sk	Num	8	Surrogate	Value is a sequential number.
MINIMUM_ BUDGET_AMT	Budget_ Amt	Num	8	N/A	Minimun_ Budget_Amt	Num	8		
SEBDMART. SEARCH_ ENGINE_DIM	Search_ engine_sk	Num	8	Natural key = search_engine.	Search_ engine_sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table.
SEBDMART. CAMPAIGN_ DIM	Campaign_ sk	Num	8	Natural key = campaignid.	Campaign_ sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table.
SEBDMART. AD_GROUP_ DIM	Ad_group_ sk	Num	8	Natural key = adgroupid.	Ad_group_sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table.
SEBDMART. AD_CREATIVE_ DIM	Ad_ creative_ sk	Num	8	Natural key = creativeid.	Ad_creative_sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table.
SEBDMART. KEYWORD_ DIM	Keyword_ sk	Num	8	Natural key = keywordid.	Keyword_ sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table.
SEBDMART. DESTINATION_ PAGE_DIM	Destina tion_uri_ page_sk	Num	7	Natural key = dest_page and dest_domain.	Destination_uri_ page_sk	Num	7	Foreign	Natural keys used to join SEBDWORK. GOOGLE_AD to dimensional table. Need to join dest_domain with Destination_ Domain_Dim to get Domain_sk.

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
SEBDWORK. GOOGLE_AD	Response date	DATE	8	Google send date in yyyy-mm-dd (2004-04-02) form when extract XML is sent. Extract application converts the value to SAS date format.	Keyword_ metric_dt	Num	8	Foreign	Responsedate is the date when impressions occurred for a keyword or content.
SEBDWORK. GOOGLE_AD	Cost	Num	8	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Cost_amt	Num	8		
SEBDWORK. GOOGLE_AD	CPC	Char	255	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Cost_per_ click_amt	Num	8		
SEBDWORK. GOOGLE_AD	Maximum CPC or Maximum CPM	Char	255	If MaximumCPC is missing, then use MaximumCPM. Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Current_bid_amt	Num	8		
SEBDWORK. GOOGLE_AD	Daily Budget	Num	8	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Budget_amt	Num	8		
SEBDWORK. GOOGLE_AD	Conver sions	Num	8		Conversion_cnt	Num	8		
SEBDWORK. GOOGLE_AD	Clicks	Num	8		Clicks_cnt	Num	8		

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
SEBDWORK. GOOGLE_AD	Impressions	Num	8		Impressions_cnt	Num	8		
SEBDWORK. GOOGLE_AD	Average position	Num	8		Average_ position_no	Num	8		
SEBDWORK. GOOGLE_AD	Adwords Type	Char	255	Strip blanks.	Syndication_ type_nm	Varch ar	50		
SEBDWORK. GOOGLE_AD	CPM	Num	8	Cost per thousand impressions. Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Cpm_amt	Num	8		
SEBDWORK. GOOGLE_AD	FirstPage CPC	Num	8	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	EST_FIRST_ PAGE_BID_AMT	Num	8		Introduced in Google Adwords V13.
SEBDWORK. GOOGLE_AD	Quality Score	Num	8		QUALITY_ SCORE_VAL	Num	3		Introduced in Google Adwords V13.

Table A3.12: Search_Query_Fact

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
Not applicable	Not applicable	Not applicable	Not applicable	Not applicable	search_ query_sk	Num	8	Surrogate	Value is a sequential number.
SEBDMART. SEARCH_ ENGINE_ DIM	Search_ engine_sk	Num	8	Natural key = search_engine.	Search_ engine_sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_SQ to dimensional table.
SEBD.CAM PAIGN_DIM	Campaign_ sk	Num	8	Natural key = campaignid.	Campaign_ sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_SQ to dimensional table.

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
SEBDMART. AD_GROUP_ DIM	Ad_ group_sk	Num	8	Natural key = adgroupid.	Ad_group_ sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_SQ to dimensional table.
SEBDMART. AD_CREATIVE _DIM	Ad_ creative _sk	Num	8	Natural key = creativeid.	Ad_ creative_sk	Num	8	Foreign	Natural keys used to join SEBDWORK. GOOGLE_SQ to dimensional table.
SEBDMART. DESTINATION _PAGE_DIM	Destination _uri_ page_sk	Num	7	Natural key = dest_page and dest_domain.	Destination _uri_page_ sk	Num	7	Foreign	Natural keys used to join SEBDWORK.GOOGLE_SQ to dimensional table. Need to join dest_domain with Destination_Domain_Dim to get Domain_sk.
SEBDWORK. GOOGLE_SQ	Response date	Date	8	Google send date in yyyy-mm-dd (2004-04-02) form when extract XML is sent. Extract application converts the value to SAS date format.	search_ query_ metric_dt	Num	8	Foreign	Responsedate is the date when impressions occurred for a keyword or content.
SEBDWORK. GOOGLE_SQ	Cost	Num	8	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Cost_amt	Num	8		
SEBDWORK. GOOGLE_SQ	CPC	Char	255	Divide by 1000000 to get monetary value. See Google Adwords (WA) for a description of AD report fields. Units depend on account currency code.	Cost_ per_click_ amt	Num	8		
SEBDWORK. GOOGLE_SQ	Conversions	Num	8		Conversion _cnt	Num	8		

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
SEBDWORK. GOOGLE_SQ	Clicks	Num	8		Clicks_cnt	Num	8		
SEBDWORK. GOOGLE_SQ	Impressions	Num	8		Impressions _cnt	Num	8		
SEBDWORK. GOOGLE_SQ	Averagepos ition	Num	8		Average_ position_no	Num	8		
SEBDWORK. GOOGLE_SQ	MatchType	Char	255		Match_Type _Txt	Char	50		This field is stripped of leading and trailing blanks.
SEBDWORK. GOOGLE_SQ	Query	Char	1024		Query_Txt	Char	1024		

APPENDIX 4

Tips and Techniques

Creating a Web Mart

Before you begin, in the appropriate sasv9.cfg file, specify the argument **-Xmx256m** in the SAS option JREOPTIONS. For more information, see “Setting the SAS Option JREOPTIONS” on page 170.

To create a Web mart:

1. Run the Site Initialization job.
2. Add the data files to the `WaSrc` directory (for one day).
3. Run the Weba_0000_Prime_Tables job.
4. Run the remaining ETL jobs for the new Web mart. You can use the weba_run_etl script to run the jobs. For more information, see Appendix 7, “The ETL Jobs.”
5. Create goal pages from the Web application. Add funnels, clickstream analysis, and so on, from the Web application as well, in case you test path analysis, funnels, and other features that are active.
6. Clean out the `WaSrc` directory.
7. Add the new data that you want to run through to the `WaSrc` directory.
8. Start with the Weba_1100_ Data_Health_Check job to catch any abnormalities in the data that you are about to load.
9. If the health check passes with no errors, then you can run the rest of the jobs.

Note: You do not actually run the _parameterized job because that is a part of the Weba_1410_Profile_Search_Engines job.

Setting the SAS Option JREOPTIONS

In JREOPTIONS, replace the minimum value in the argument **–Xmx128m** with **256** in the appropriate sasv9.cfg file. If the value is set too low, the site initialization fails and returns **ERROR: Insufficient memory available for Java**.

To check the setting of JREOPTIONS, run the following SAS code that uses the same SAS executable command and environment that is used for the site initialization program:

```
proc options option=jreoptions;
run;
```

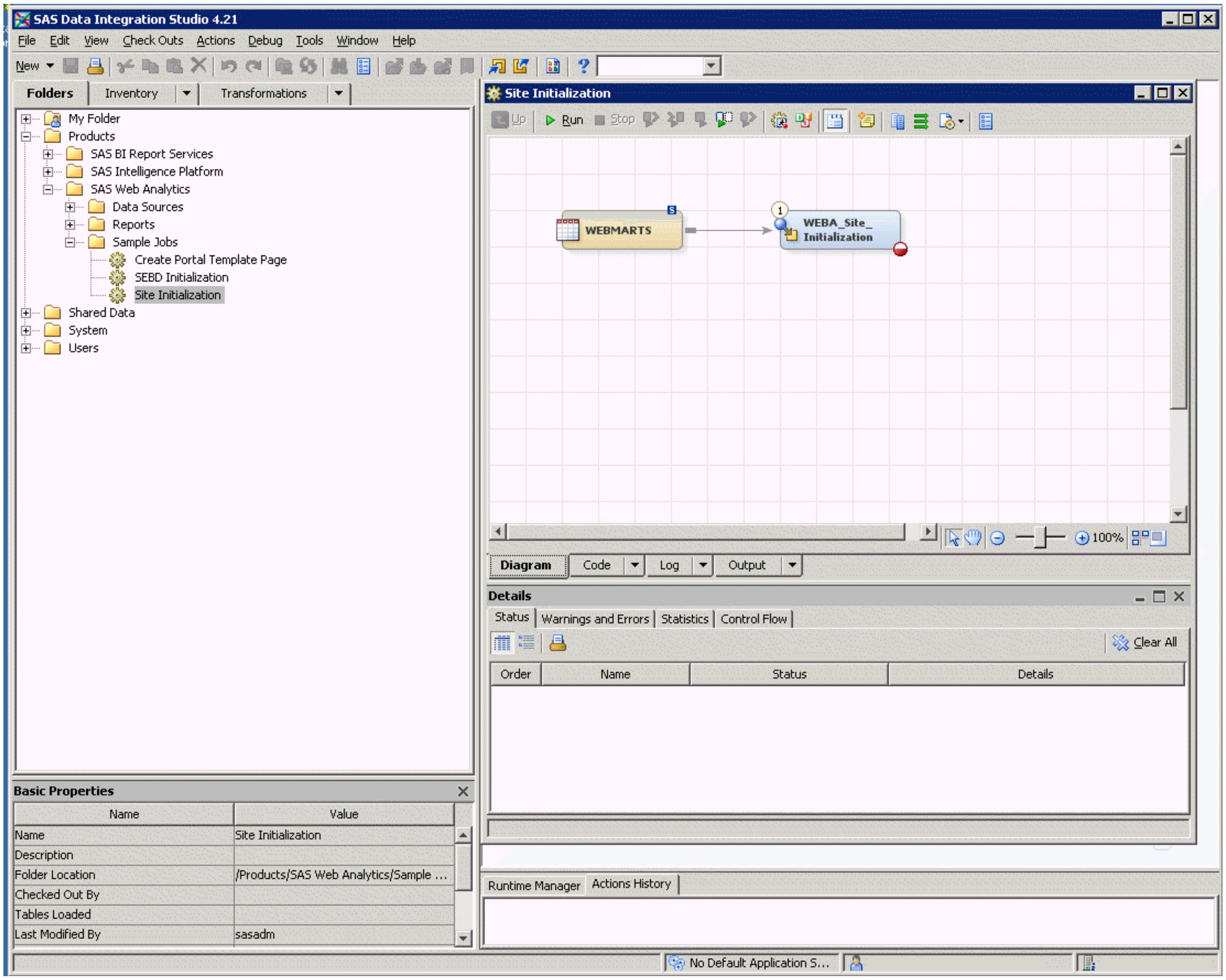
Re-creating Metadata for an Existing Web Site

To re-create metadata for an existing Web site:

1 Open SAS Data Integration Studio and navigate to **`/Products/SAS Web Analytics/Sample Jobs`**.

2 Copy the Site Initialization job and use Paste Special to paste the job into another folder. For example, use **`/My Folder`** or **`/Shared Data/My Folder`**.

3 Open the Site Initialization job by double-clicking the job.


SAS Data Integration Studio 4.21
File Edit View Check Outs Actions Debug Tools Window Help
New
Folders Inventory Transformations
My Folder
Products
SAS BI Report Services
SAS Intelligence Platform
SAS Web Analytics
Data Sources
Reports
Sample Jobs
Create Portal Template Page
SEBD Initialization
Site Initialization
Shared Data
System
Users
Site Initialization
Up Run Stop
WEBMARTS
WEBA_Site_ Initialization
100%
Diagram Code Log Output
Details
Status Warnings and Errors Statistics Control Flow
Clear All
Order Name Status Details
Runtime Manager Actions History
No Default Application S...

Basic Properties

Name	Value
Name	Site Initialization
Description	
Folder Location	/Products/SAS Web Analytics/Sample ...
Checked Out By	
Tables Loaded	
Last Modified By	sasadm

4 Double-click the **WEBA_Site_Initialization** transformation within the selected job, and then select the **Options** tab in the Properties dialog box. Complete the required fields. In **Web mart creation action,** you select **Recreate the metadata for an existing Web mart**.

Accessing SAS Web Analytics Macros

SAS Web Analytics ETL jobs and report stored processes use solution-specific macros. These macros are automatically available to a SAS session that is started by one of the following:

- stored process server
- workspace server
- SAS Batch server

In order to automatically have access to the SAS Web Analytic macros within an interactive SAS session, complete one of these tasks:

- Run the script file that is located in <*your-server-installation-location*>/**`config/Lev1/SASApp`**.

-Windows example – **`C:\SAS\config\Lev1\SASApp\sas.bat`**

-UNIX example –- **`/usr/local/config/Lev1/SASApp/sas.sh`**

□ Open an interactive SAS session and run the following statements from the editor within the Display Manager:

-Windows

```
Options sasautos=(SASAUTOS “<sasroot directory>\SASFoundation\9.2
\weba\ucmacros”);
```

```
%include ““<sasroot directory>\SASFoundation\9.2
\weba\sasmisc\weba_autoexec.sas”;
```

-UNIX

```
Options sasautos=(SASAUTOS “<sasroot directory>/SASFoundation/9.2
/ucmacros/weba”);
```

```
%include ““<sasroot directory>/SASFoundation/9.2
/misc/weba/weba_autoexec.sas”;
```

Adding Search Engines to an Existing Search_Engine_Dim Table

To add search sites, you first need to gather information.

1 Determine whether your site of interest is one of the following:

search engine uses its own proprietary search algorithm.

search site uses a search engine provided by other companies to return search results. These sites also provide other content.

Aggregator searches multiple search engines and returns a results page that indicates which search engine returned each link. If it is an aggregator, it is not an independent search site.

2 Determine what the search site referrer will look like. For example, if the search site is www.mysearches.com, then the referrer should be http://www.mysearches.com/searches?query=my+searches. The referrer should show the query string that contains the search terms.

3 You can use these plug-ins to help you determine the page referrer:

□ Firefox - HTTP Live Headers

□ Internet Explorer ieHTTPHeaders

4 For each search engine that will be added to the Search_Engine_Dim table, gather the following information for the fields indicated:

Table A4.1: SEARCH_ENGINE_DIM Table Fields for Adding Search Engines

Field	Description
SEARCH_ENGINE_DOMAIN_TXT	The portion of the search site domain that uniquely identifies a search site. For example, **google** is the unique portion of these domains: **`www.google.com`**, **`www.google.fr`** and **`www.google.cn`**. Verify that only the search engine is identified by the portion of the domain that is entered into this field.
SEARCH_ENGINE_DESC	The search engine name. Typically this is the name that is found on the search engine site.
SEARCH_TERM_PARM_TXT	The query string parameter used by the search engine to contain the search terms entered by the search engine user. To determine which query string parameter is being used, look for the parameter that contains the search terms that you typed into the search box. Typically, spaces appear as **+** and other characters (**<>**,**'**, and **"**) are HTML-encoded. This helps you identify the search terms parameter.

After you have gathered all the appropriate information, you can update the Search_Engine_Dim table by using one of these methods:

- ❑ Manually – You can update the Weba_Search_Engine_Dim table, which is located in the Sashelp library, by using the table view within an interactive SAS session or within SAS Enterprise Guide. For information about using the table view, see SAS Help and Documentation or the online Help in SAS Enterprise Guide.
- ❑ Programmatically – You can run the following code in a SAS interactive session to update the Sashelp.Weba_Search_Engine_Dim:

```
data new_search_engines;
 infile datalines dsd delimiter=',';
 attrib SEARCH_ENGINE_DOMAIN_TXT    length=$65  format=$65.
        SEARCH_ENGINE_DESC          length=$130 format=$130.
        SEARCH_TERM_PARM_TXT        length=$25  format=$25.
        PAID_PARM_TXT               length=$45  format=$45.;
 informat search_engine_domain_txt $65. search_engine_desc $130.
          search_term_parm_txt $15. paid_parm_txt $45.;
 input search_engine_domain_txt $
       search_engine_desc $
       search_term_parm_txt $;
 paid_parm_txt=;
 datalines;
 /* put new search engine information here - here is an example line
bing       ,Bing                              ,q
 */
run;
```

```
libname sh "<sashelp path>";
data sh.weba_search_engine_dim;
 attrib SEARCH_ENGINE_DOMAIN_TXT    length=$65  format=$65.
        SEARCH_ENGINE_DESC          length=$130 format=$130.
        SEARCH_TERM_PARM_TXT        length=$25  format=$25.
        PAID_PARM_TXT               length=$45  format=$45.;
 set sh.weba_search_engine_dim
     new_search_engines;
run;
```

After the Sashelp.Weba_Search_Engine_Dim table is updated, run the <*Web mart root folder*>**/5.4 Jobs/Optional Jobs/weba_1101_load_search_engine_dim** job for each Web mart to update the Search_Engine_Dim table.

Note: It is not recommended that this job be part of the ETL because new search sites will occasionally be added.

Making a Web Mart Available for Reports in a Production Environment

If you are moving a Web mart from a test environment to production, the last step is to make the Web mart available in drop-down lists for SAS Web Report Studio and SAS Web Analytics reports.

Use the method in this section only if all of the following criteria are met. Otherwise, reporting errors will occur.

❑ You are moving a Web mart from a test machine to a production machine.

❑ Site initialization is complete on the production machine.

❑ Data has been moved from the test machine to the production machine.

To make the Web mart available for reports:

1 Open an interactive SAS session and enter **`metabrowse`** on the command bar.

2 Log on to the appropriate metadata server. Use sasadm@saspw as the user ID.

3 In the Metadata Browser window, select **Tools** ▸ **Options** ▸ **Explore**.

4 In Explore Options window, click the **General** tab. In the Miscellaneous area, clear **Metadata Browse mode** and click **OK**.

5 In the Metadata Browser pane, expand **Foundation**.

6 Scroll down within **Foundation**. Expand, and then select **Transformation**.

7 In the right pane, double-click the Web mart object. (The transformations can be sorted by name and type.)

8 In the far right pane there is a list of attributes for the Web mart object. Right-click on the **IsActive** attribute, and then click **Modify**.

9 In the **Modify Value** box, change the value from `0` to `1.` Click **OK**.

The Web mart is now visible in the Weba.Webmarts data set, which is used for the reports drop-down lists.

Removing a Site

You can remove a Web site either by running a job to delete its metadata and tables, or by deleting its metadata manually.

To run the job:

1 In SAS Data Integration Studio, navigate to the **`/Products/SAS Web Analytics/Sample Jobs`** folder.

2 Open the Site Metadata Cleanup job.

3 Read the Read Me First note and follow the recommendations in the note.

To remove a Web site's metadata manually:

4 Open SAS Management Console and log on as the SAS Administrator or as a SAS Web Analytics Administrator.

5 Click the **Folders** tab.

6 Navigate to **`/System/Applications/SAS Web Analytics/Web Analytics 5.4/Sites`**.

Delete the folder with the name of the site that you want to remove. This deletes the Web mart object that is assigned to the site and the stored processes that are associated with the SAS Web Analytics application reports.

7 Navigate to the folder where the site jobs were installed. For example, if the site was initialized under **`/Shared Data/Web Analytics`**, then navigate to that folder. Delete the folder with the name of the site. Deleting the folder removes the following:

jobs
tables
libraries

8 Exit SAS Management Console.

To complete the removal, delete the data mart that is associated with the site.

9 Navigate to the directory where the site's data mart exists.

10 Determine whether archiving the data is necessary.

If archiving is necessary, then archive the data according to your standard practices. After the archival is complete, delete the site's data mart directory and subdirectories. If archiving is not necessary, then delete the site's directory and subdirectories.

If the Web mart data is Oracle, then contact the Oracle database administrator about the Oracle data repository.

Communicate with your database administrator to determine how the database tables associated with the site should be handled.

Loading MaxMind Data

If you are loading MaxMind data with The Weba_9999_Maxmind_Load job, you must supply the location of the CSV files inside the MaxMind job in SAS Data Integration Studio. Specify the location on the **File Location** tab of the properties for the City Blocks node and the Geographic City nodes.

If the CSV file location is not supplied, an error is displayed in the log:

```
Apparent symbolic reference NOT_CONNECTED not resolved
```

Note: If you are running the MaxMind job on UNIX, you must deploy the job the same way the Site Initialization job was deployed. For more information, see the topic “Initialize the Site” in Chapter 2, “Setting Up a Web Mart.”

Setting y-axis Increments for Monitor and Insight Reports with Custom Metrics

If you created custom metrics for your Monitor and Insight reports, you must format the metrics within the WaAnly.Anly_<*interval*>_metrics_vw table objects by using SAS Data Integration Studio. If you do not format the metrics, then the values for the y-axis increments (tick marks) will be based on unformatted values and therefore will not display properly.

To format the metrics, navigate to the <*Web mart*>**\WaAnly** folder and edit the Anly_<*interval*>_metrics_vw table objects as follows:

1. Double-click on one of the **Anly_**<*interval*>**_metrics_vw** table objects.
2. Click the **Columns** tab.
3. Add a format for the metric.
4. Save any changes.
5. Redeploy the <*Web mart*>**\5.4 Jobs\ETL Jobs\Webmart\weba_2501_prep_insight_monitor** job.
6. To update the analytic views before the next scheduled ETL run, select **Exclude from Run** in nodes 1 and 2 in the Weba_2501_Prep_Insight_Monitor job, and then run the job.

Specifying an Alternate Directory for ORA_LOGDIR

The ORA_LOGDIR option is a used within SAS Web Analytics applications that have an Oracle repository. When Bulkload is selected as a loading option, either explicitly or conditionally, SAS saves the Oracle bulkload log separately from the SAS log. It attempts to place the log in a default location, which sometimes causes an error similar to the following:

```
ERROR: Insufficient authorization to access <remote
directory>/BL_JOB_0.ctl
```

This error is triggered when the user ID that is running the ETL job does not have write access to the appropriate default directory.

Rather than requiring write access to the default directory, SAS Web Analytics enables you to specify an alternate directory. To set the ORA_LOGDIR option:

1. Open SAS Data Integration Studio.
2. Navigate to **`/Shared Data/Web Analytics/`**<*Web mart*>**`/WaCnfg`**, or to the folder where the Web mart ETL folders are located.
3. Right-click the **Webmart** table and select **Update Metadata**. This updates the table metadata to include the necessary column.
4. Right-click the **Webmart** table and select **Open**.
5. Enter the new path in the ora_logdir column. All ETL jobs for the Web mart will now write any Oracle bulkload logs to this directory.
6. Save the changes.

For more information about the Oracle bulkload option that ORA_LOGDIR uses, see the BL_DEFAULT_DIR= data set option in the *SAS/ACCESS(R) 9.2 for Relational Databases: Reference*, available at **`http://support.sas.com/documentation/cdl/en/acreldb/63647/HTML/default/viewer.htm#a003359598.htm`**.

Tracking Internal Searches

For information about tracking internal searches for content within a site, see the *SAS Data Surveyor for Clickstream Data 2.2: User's Guide*.

APPENDIX 5

How the Aggregate Jobs Work

Regenerate Aggregates

Introduction

You can edit the aggregate jobs to reload aggregate tables with all of the history that is currently stored in the warehouse tables. The process varies depending on the type of data that is in the tables.

SAS Data

1. Within a Web mart, back up the physical WaMart aggregate tables (tables with the prefix aggr_) that are to be reloaded. Ensure that the ETL is not running, and that the ETL is not scheduled to start.
2. Delete the same physical aggregate tables from the WaMart library. Start a Base SAS session, assign a library to the WaMart path, and delete the appropriate tables.
3. In SAS Data Integration Studio, navigate to the <*Web mart*>/**WaWork** folder. Open the Properties window of the Web mart's WaWork library object. Change the Path Specification on the **Options** tab to the same path that the WaMart library object path specifies. For example, WaMart points to <*path*>/**wamart**, and WaWork points to <*path*>/**wawork.** Change the WaWork library object to point to <*path*>/**wamart**.
4. The aggregate jobs contain the numbers 1500 through 2150 inclusively, with _agg after the number. Run the job for each aggregate to be reloaded and ensure that the job executed successfully. The jobs might run for a long time because they are reaggregating an entire warehouse of data. Address any issues that you encounter, and then repeat steps 2–4 for each job.
5. Change the WaWork library object path back to its original value. For example, change the library path from <*path*>/**wamart** to <*path*>/**wawork**.
6. Regenerate the job code and reschedule the ETL.

Oracle Data

Step 1: Back up physical WaMart aggregate tables.

Within a Web mart, back up the physical WaMart aggregate tables (tables with the prefix aggr_) that are to be reloaded. Ensure that the ETL is not running, and that the ETL is not scheduled to start.

Step 2: Delete the physical aggregate tables from the WaMart library that you backed up in step 1.

Start a Base SAS session, assign a library to the WaMart path, and delete the appropriate tables.

Step 3: Modify each aggregate job to be reloaded.

The aggregate jobs contain the numbers 1500 through 2099 inclusively, with _agg after the number. For each aggregate job to be reloaded, complete these steps:

1 In SAS Data Integration Studio, navigate to the <*Web mart*>/**`5.4 Jobs/ETL Jobs`** folder and open the aggregate job that is to be run.

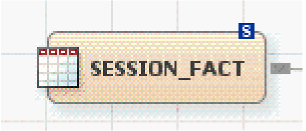
  Each aggregate has either a SESSION_FACT table, a DETAIL_FACT table, or both.

2 For each table, right-click either the **SESSION_FACT** or **DETAIL_FACT** table and select **Properties**. On the **Physical Storage** tab next to the Library box, click the **...** button. Navigate to the <*Web mart*>/**`WaMart`** folder and select the mart's **WaMar**t library object. Click **OK** and close the Properties window. The red **O** in the upper right corner of the table object indicates an Oracle table.

3 Click **Run** to run the modified job.

4 After you verify that the job ran without errors and the aggregate tables were created successfully, restore the job's original settings for every fact table object.

  a. Right-click the table and select **Properties**. On the **Physical Storage** tab next to the Library box, click the **...** button. Navigate to the <*Web mart*>/**`WaWork`** folder and select the mart's WaWork library object. Close the Properties window. The blue **S** in the upper right corner of the table object indicates a SAS data set.



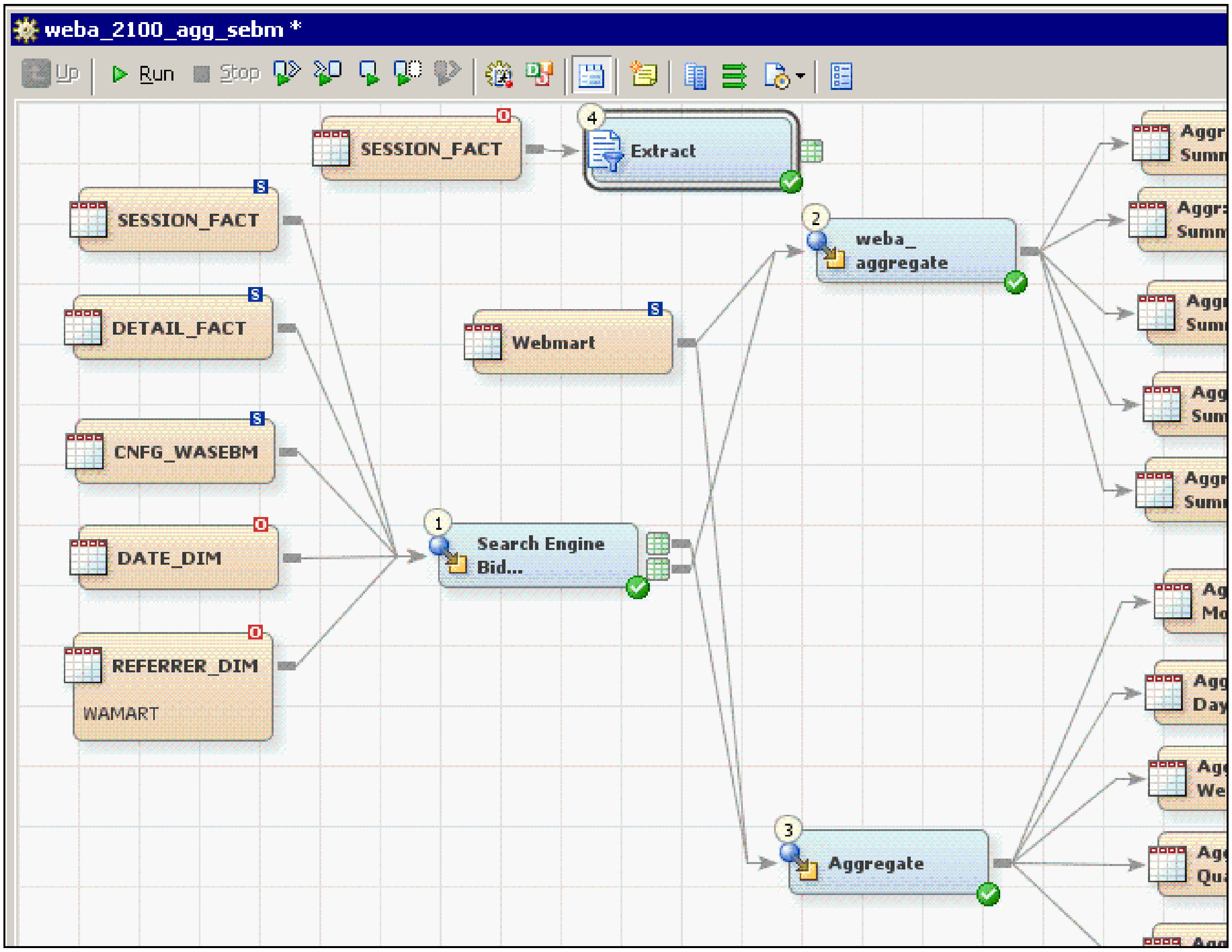
  b. Save the job and close it.

5 For the Weba_2100_Agg_Sebm job and the Weba_2150_Seb_Campaigns jobs, complete these steps:

  a. In SAS Data Integration Studio, navigate to the <*Web mart*> **`5.4 Jobs/ETL Jobs`** folder and open the aggregate job that is to be run.

  b. On the **Folders** tab, navigate to the <*Web mart*>/**`WaMart`** folder and drag the SESSION_FACT table to the diagram.

c. Click the **Transformations** tab, expand the **Data** category, and then drag the Extract transformation to the diagram. Connect the WaMart.SESSION_FACT object as input to the Extract transformation.



d. Double-click the **Extract** transformation. Click the **Mappings** tab. Use the Propagate from sources to targets icon to populate the Target table side of the diagram on the right.

Select the **session_dt** column under the Target table. Right-click the selected column, and then select **Expression ▸ Advanced**.

e. In the Expression window, convert the Oracle date value to a SAS date value. Click the **Functions** tab. Double-click the **Date and Time** functions folder, and then select the **DATEPART** function. Click the **Data Sources** tab. Double-click the **SESSION_FACT** table, and then select the **session_dt** column. Click **OK** to close the Expression Builder and Extract Properties windows.

f. Right-click the output table of the **Extract** transformation and select **Properties**.

g. On the **Physical Storage** tab, type **`session_fact`** in the **Physical Name** field. Select the **Create as view** check box. In the Location area, select **Redirect to a registered library**. Click the **...** button next to the Library box, and navigate to the <*Web mart*>**/`WaWork`** folder. Select the **WaWork** library object. Click **OK** to close the Extract Properties window.

h. Right-click on the line that connects the original SESSION_FACT object to the SEB Campaign Summary transformation (for the weba_2150_seb campaigns job), and then click **Delete**.

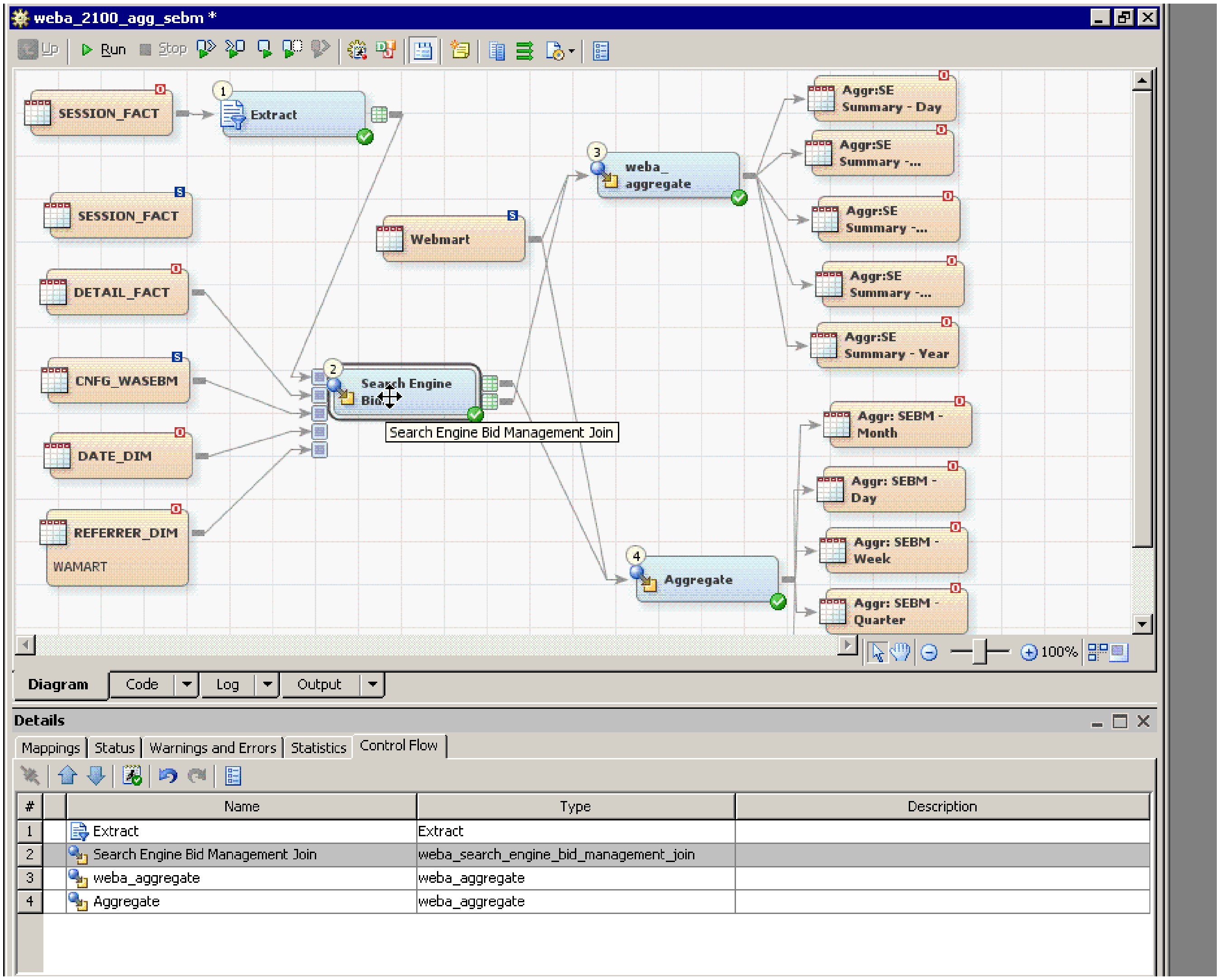
   Note: For the 2100_agg_sebm job, use the Search Engine Bid Management join.

i. Drag the output icon of the new Extract transformation and connect it to the SEB Campaign Summary transformation (for the weba_2150_seb campaigns job).

   Note: For the 2100_agg_sebm job, use the Search Engine Bid Management join.

j. At the bottom of the screen, click the **Control Flow** tab in the Details section. Select the new **Extract** transformation and use the up arrow to move it to the top of the list of transformations.

k. Right-click the **DETAIL_FACT** table and select **Properties**. On the **Physical Storage** tab next to the Library box, click the **...** button. Navigate to the <*Web mart*>**/`WaMart`** folder and select the mart's **WaMart** library object. Close the DETAIL_FACT Properties window. The red **O** in the upper right corner of the DETAIL_FACT object indicates an Oracle table.



#	Name	Type	Description
1	Extract	Extract	
2	Search Engine Bid Management Join	weba_search_engine_bid_management_join	
3	weba_aggregate	weba_aggregate	
4	Aggregate	weba_aggregate	

l. Click **Run** to run the modified job.

m. After you verify that the job ran without errors and the aggregate tables were created successfully, restore the job's original settings as in step 4.

- ❑ Right-click the **DETAIL_FACT** table and select **Properties**. On the **Physical Storage** tab next to the Library box, click the **...** button. Navigate to the <*Web mart*>/`WaWork` folder and select the mart's **WaWork** library object. Close the DETAIL_FACT Properties window. The blue **S** in the upper right corner of the DETAIL_FACT object indicates a SAS data set.
- ❑ You can close the job without saving it and the modifications will not be saved. If you did save the job, delete the new WaMart.SESSION_FACT table and the new Extract transformation. Reconnect the original WaWork.SESSION_FACT table to the SEB transformation. Save the job and close it.

6 Reschedule the ETL.

Aggregate Properties Options

Aggregation rules are determined by user selections. For example, By variables control the granularity of the table by enabling you to identify Class variables. Date is always used as a Class variable, so leaving this value blank aggregates results by date only. Statistics variables identify fields that are summarized to produce the output requested. Error types are a separate case and addressed later in this section. The order of the variables is irrelevant. Each type of variable is described here.

By Variables

sets the table granularity of the aggregates and the date field. Statistics are calculated (grouped) by the fields noted here, with the date. The default setting is blank.

Additional Summary Statistics by Date

enables the consumer to enter SQL code to create new fields. These fields are grouped by date only, regardless of any By variables mentioned earlier. This is useful for calculating denominators used in averages later. For example, `sum(session_count) as session_total` calculates a denominator for a percentage formula, which is then used in an information map. The default setting is blank.

Statistics Variables

summarizes numerical statistics to produce the desired granularity. These statistics must be numeric fields. The default setting is blank.

Add Day of Week Field

adds a grouping element named day_of_week to the result set as part of the final step if this variable is set to YES. The field is a result of a computation of the date to extract the day of the week by using the WEEKDAY SAS function.

Days/Weeks/Months/Quarters/Years of History

specifies the number of days, weeks, months, quarters, or years of history to retain, respectively. The term “history” is calculated at load time. The process takes the newest date value in all of the historical tables and new data, programmatically determines the start of history, and purges all data older than the starting date. With this algorithm, the process is data-driven and not based on the current date. The default values for days, weeks, months, quarters, and years are 365, 52, 12, 4 and 1, respectively.

Drop and Re-create Indexes

uses a load process to drop the indexes into the RDBMS table before appending, and will re-create the indexes upon completion of the load. If the setting is NO, the indexes remain intact throughout the load process. This setting has no impact if the repository is not Oracle. The default setting is NO.

Oracle Upsert

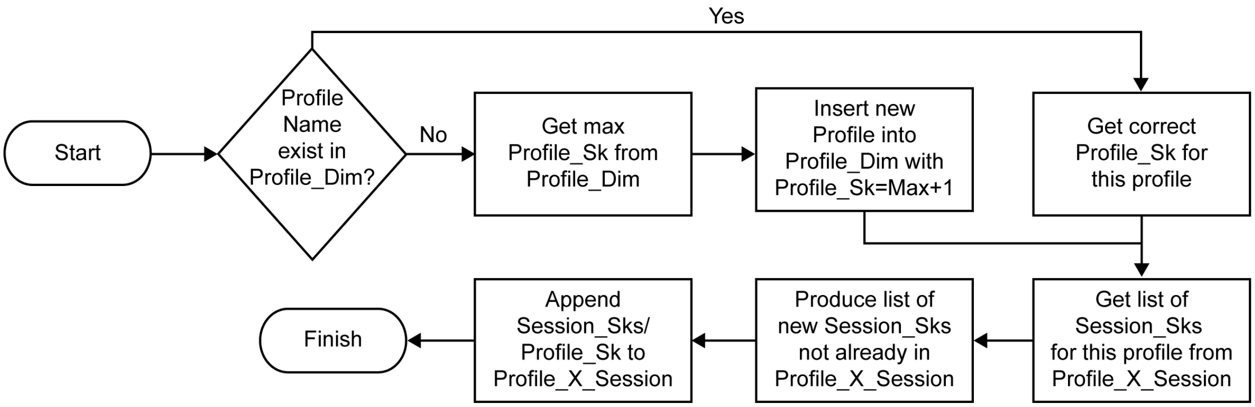
loads Oracle tables with the new upsert functionality, where updating and appending are completed in the same step, if set to YES. This option does not apply to non-RDBMS repositories. This setting has no impact if the upstream tables are not Oracle tables. The default setting is NO.

Bulkload

bulkloads Oracle tables during the append stage of the process, if set to YES. If set to NO, records are individually inserted into Oracle tables. If set to Conditional, then loads that have more records than the value of the THRESHOLD setting will be bulk loaded. Loads that have less than the THRESHOLD value and that have been given a Conditional value will be inserted.

You can create additional statistics by editing the transformation and adding the requisite option to a drop-down list. Open the trans_new macro and adjust the code. However, formats and data types are inherited from the data that is passed into the aggregate transformation. To adjust these, adjust the inputs to the transformation, and apply the changes to the aggregate tables directly before running the next job. Similarly, data can be reformatted for reporting at the time that the data is extracted.

Figure 5.1: Aggregate Transformation Process


Yes
Start
Profile Name exist in Profile_Dim?
No
Get max Profile_Sk from Profile_Dim
Insert new Profile into Profile_Dim with Profile_Sk=Max+1
Get correct Profile_Sk for this profile
Get list of Session_Sks for this profile from Profile_X_Session
Produce list of new Session_Sks not already in Profile_X_Session
Append Session_Sks/ Profile_Sk to Profile_X_Session
Finish

Visitor Aggregate

The visitor aggregate supplies statistics to the processes that require them. The source data is an aggregate table. Aggregate tables enhance performance by calculating statistics from warehouse aggregates directly. Any aggregate that uses this functionality must have the visitor aggregate as a dependent job. The process takes any aggregate with date information and adds two fields to them: repeat visitors and unique visitors.

Figure 5.2: Visitor Aggregate in SAS Data Integration Studio

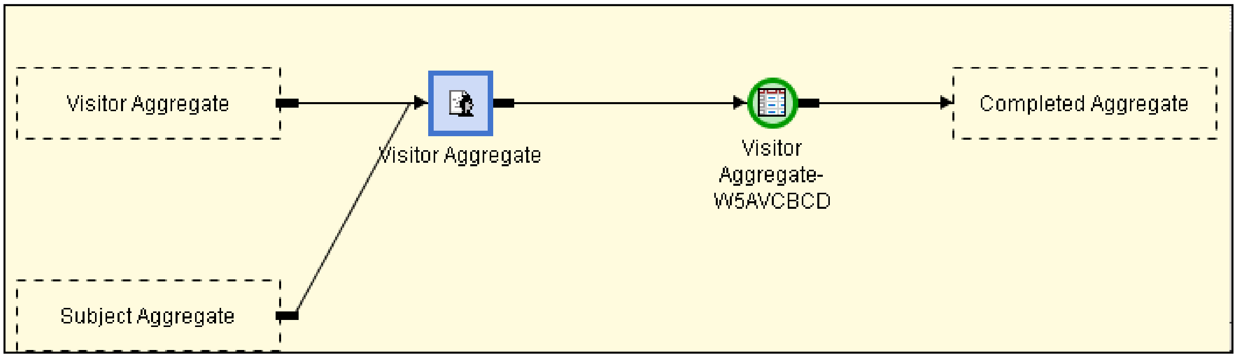
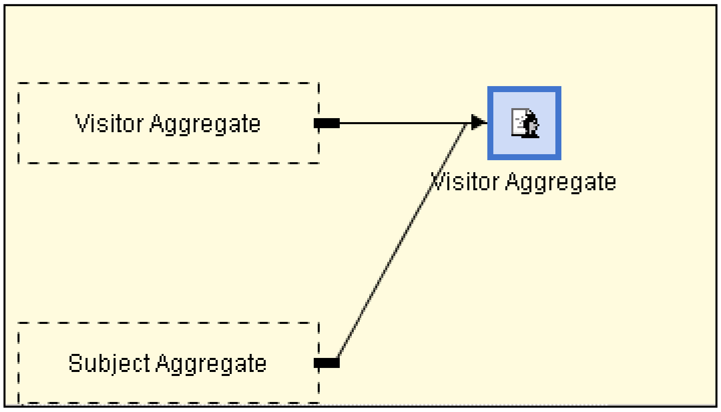


Figure 5.3: Visitor Aggregation Flow



The subject aggregate is modified to retain unique and repeat visitor information at a Session_Sk level. If the Repeat_Visitors and Unique_Visitors fields are in the source table aggregate, they will be dropped and added during execution with correct numbers.

Oracle Databases and the ETL Aggregate Jobs

When Oracle tables are created through the SAS/ACCESS Interface to Oracle LIBNAME engine, character column lengths might be multiplied. This is because of Oracle installation options that are related to the handling of multi-byte character set (MBCS) data. The problem occurs in the aggregate jobs and might cause warnings that character columns have different lengths than expected. Here are possible solutions to the problem:

- ❑ Modify the Oracle installation.
- ❑ Add these options to the Web mart Oracle library object definition by clicking **Advanced Options**, and then clicking the **Other Options** tab to prevent the column expansion from occurring:

  ADJUST_BYTE_SEMANTIC_COLUMN_LENGTHS=NO DBSERVER_MAX_BYTES=1 DBCLIENT_MAX_BYTES=1
- ❑ Add this statement to the autoexec.sas file that is used when the ETL for the aggregate jobs is run:

  ```
  %let orawrn=NO;
  ```

For more information, see the SAS/ACCESS Interface to Oracle chapter in *SAS/ACCESS 9.2 for Relational Databases: Reference*.

APPENDIX 6

Migrating Web Marts to 5.4

Preparing to Migrate

If you are upgrading to SAS Web Analytics 5.4 from 5.3, 5.3.1, or 5.3.3, prepare to migrate your Web mart by following these steps:

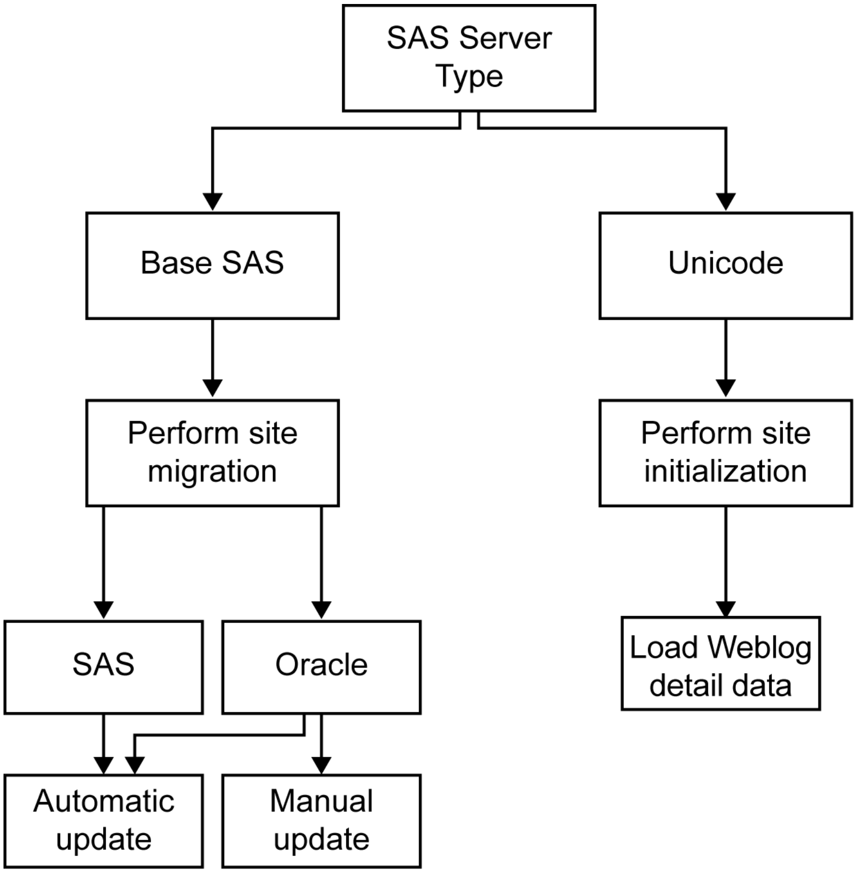
1. Check the folder authorizations against the recommended authorizations for the following folders. For more information, see Chapter 1:
   - ❑ **`/Products/SAS Web Analytics`**
   - ❑ **`/Shared Data`**
   - ❑ **`/System/Applications/SAS Web Analytics`**
2. Back up the metadata repository. For more information, see “About the Metadata Server Backup and Restore Utilities” in the *SAS 9.2 Intelligence Platform: System Administration Guide* at `http://support.sas.com/documentation/cdl/en/bisag/60945/HTML/default/a003267606.htm`.
3. Create backup copies of the Web mart directories.
4. Decide whether the SAS Unicode server will be implemented in 5.4 and take appropriate action.
5. (Oracle only) Analyze your tablespace requirements. For more information, see “Oracle Considerations” on page 190.
6. Update the Java value in the sasv9.cfg file for the JREOPTIONS SAS option from **-Xmx128m** to **-Xmx256**. For more information, see Appendix 4, “Tips and Tricks.”

How SAS Unicode Affects Your Migration Path

Your migration path is determined by whether the SAS Unicode server will be implemented. It is recommended that you implement the SAS Unicode server if you need the SAS Web Analytics Web application to support multiple character sets. For more information, see “Introduction to SAS Unicode” in Chapter 1.

If the SAS Unicode server is implemented, then reinitialize and reload all existing Web marts. If the SAS Unicode server is not implemented, then you can migrate all existing Web marts.

Figure A6.1: How SAS Unicode Affects Your Migration Path



The remaining topics in this appendix address the tasks that you need to perform according to your chosen migration path.

Oracle Considerations

If your data repository is Oracle, you must answer the following questions during migration:

- **Tablespace increase?** The migration increases column widths for all character fields and adds several columns. Analyze the amount of tablespace that is available and increase it if necessary. For information about the data model, see Appendix 1.
- **Update the tables automatically?** Determine whether the tables should be updated manually or updated automatically by the migration job.

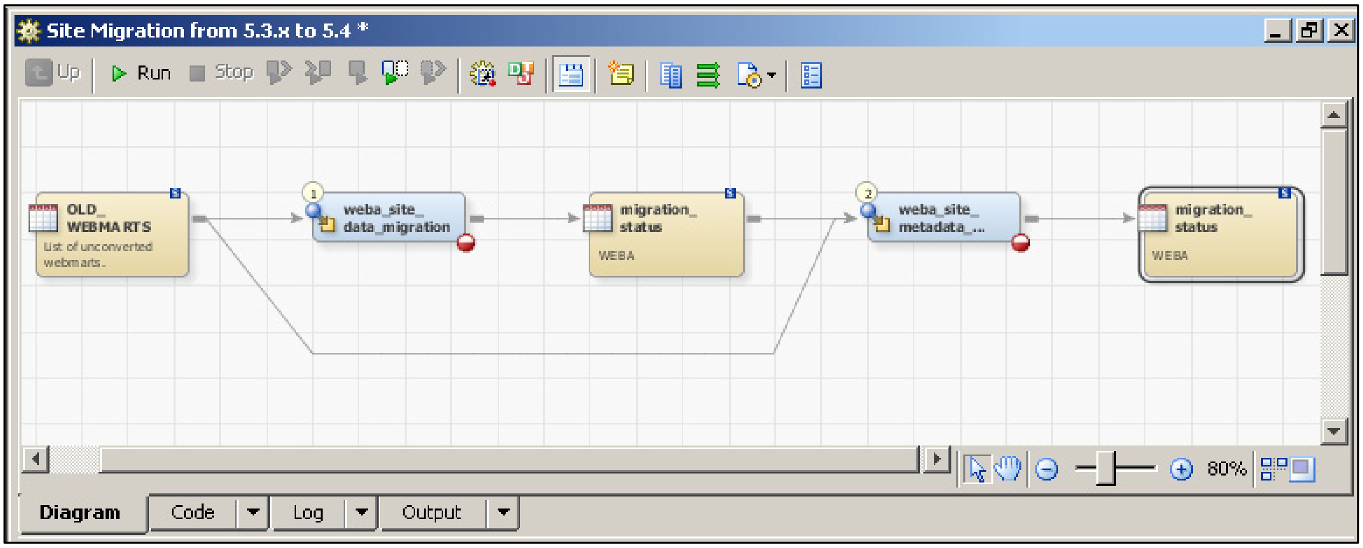
Site Migration Steps

1 Ensure that the last ETL scheduled for each site that is affected by the migration has completed successfully. This step should have occurred prior to upgrading to SAS Web Analytics 5.4.

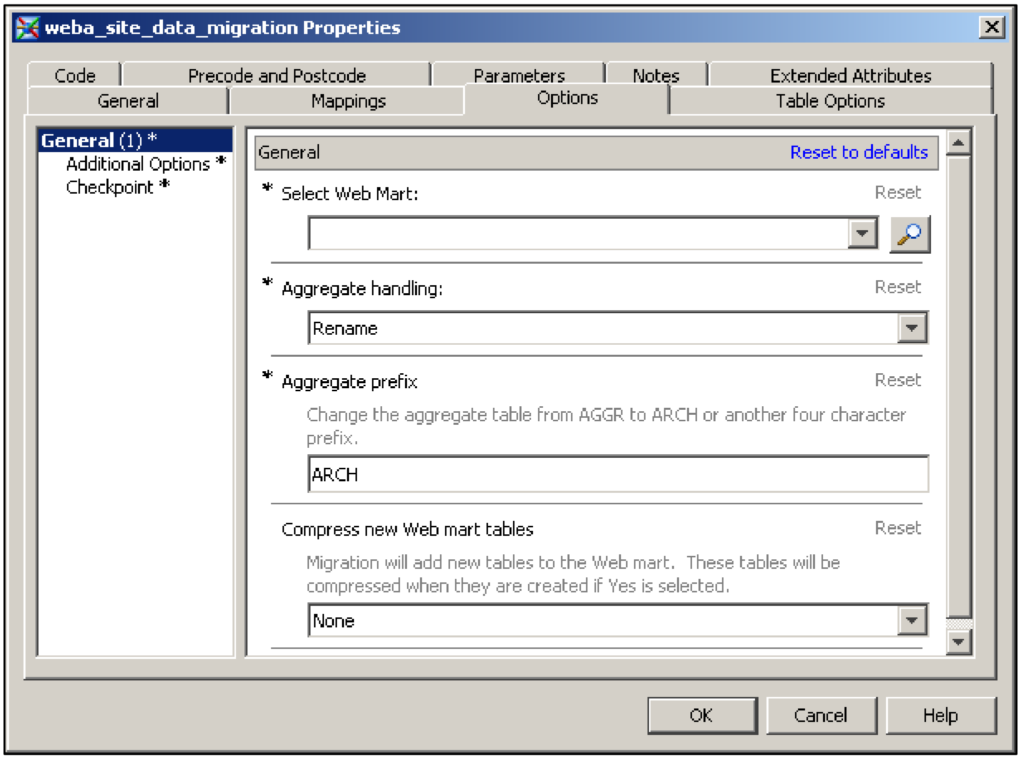
Note: Before you continue with the migration, complete the steps in "Preparing to Migrate" on page 189.

2 Open SAS Data Integration Studio and navigate to the `/Products/SAS Web Analytics/Sample Jobs` folder.

a. Double-click the **Site Migration from 5.3.x to 5.4** job.



b. Right-click on the **weba_site_data_migration** node and select **Properties**. Click the **Options** tab.



weba_site_data_migration Properties

* Select Web Mart:

* Aggregate handling: Rename

* Aggregate prefix

Change the aggregate table from AGGR to ARCH or another four character prefix.

ARCH

Compress new Web mart tables

Migration will add new tables to the Web mart. These tables will be compressed when they are created if Yes is selected.

None

c. Select a Web mart from the drop-down list.

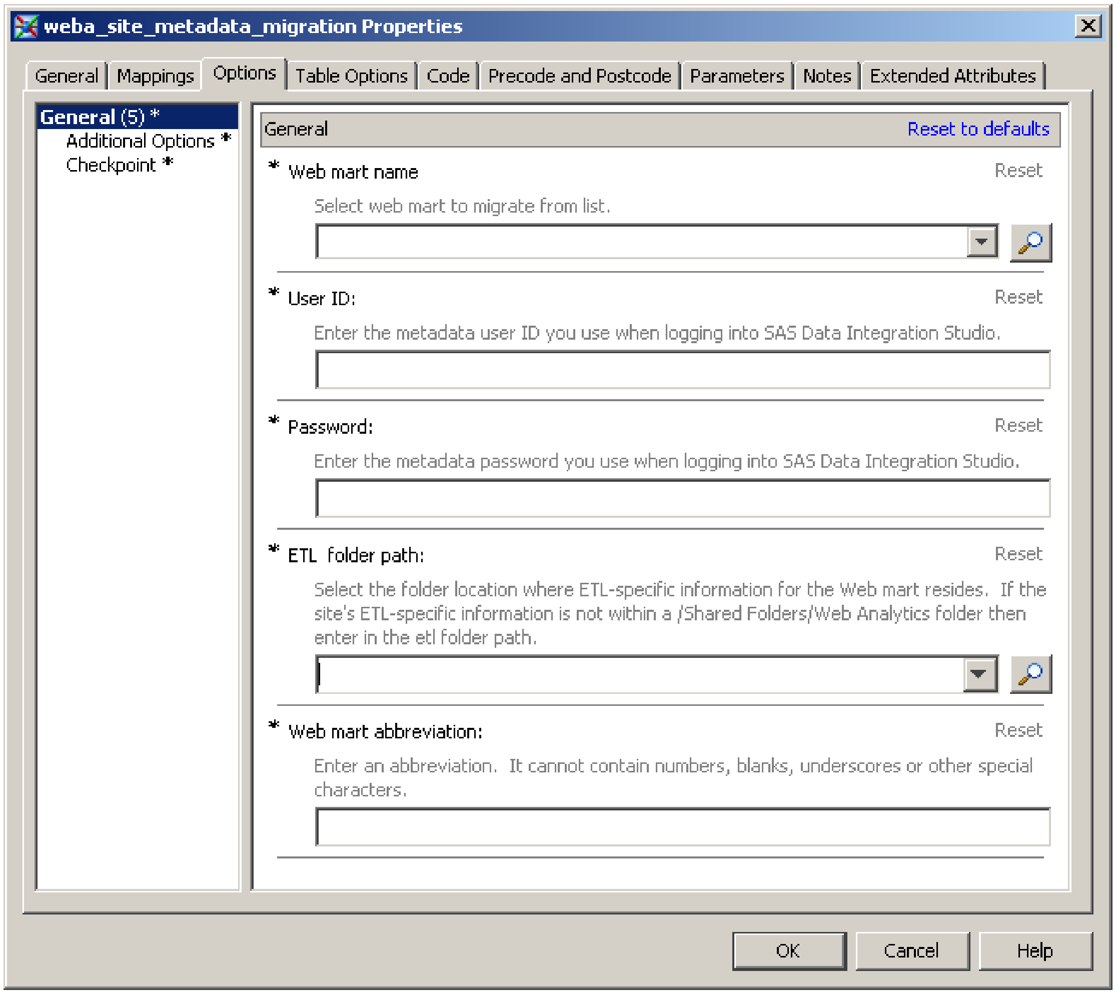
d. Choose how to handle aggregates. One of the requirements in moving from a 5.3x Web data mart to a 5.4 Web data mart is the re-creation of the aggregates. The choices are shown here:

Rename	renames the default.
Move	moves the existing aggregates to another location.
Delete	deletes the aggregates.

e. Click **OK**.

3

a. Right-click the **weba_site_metadata_migration** node and select **Properties**. Click the **Options** tab.



* Web mart name

Select web mart to migrate from list.

* User ID:

Enter the metadata user ID you use when logging into SAS Data Integration Studio.

* Password:

Enter the metadata password you use when logging into SAS Data Integration Studio.

* ETL folder path:

Select the folder location where ETL-specific information for the Web mart resides. If the site's ETL-specific information is not within a /Shared Folders/Web Analytics folder then enter in the etl folder path.

* Web mart abbreviation:

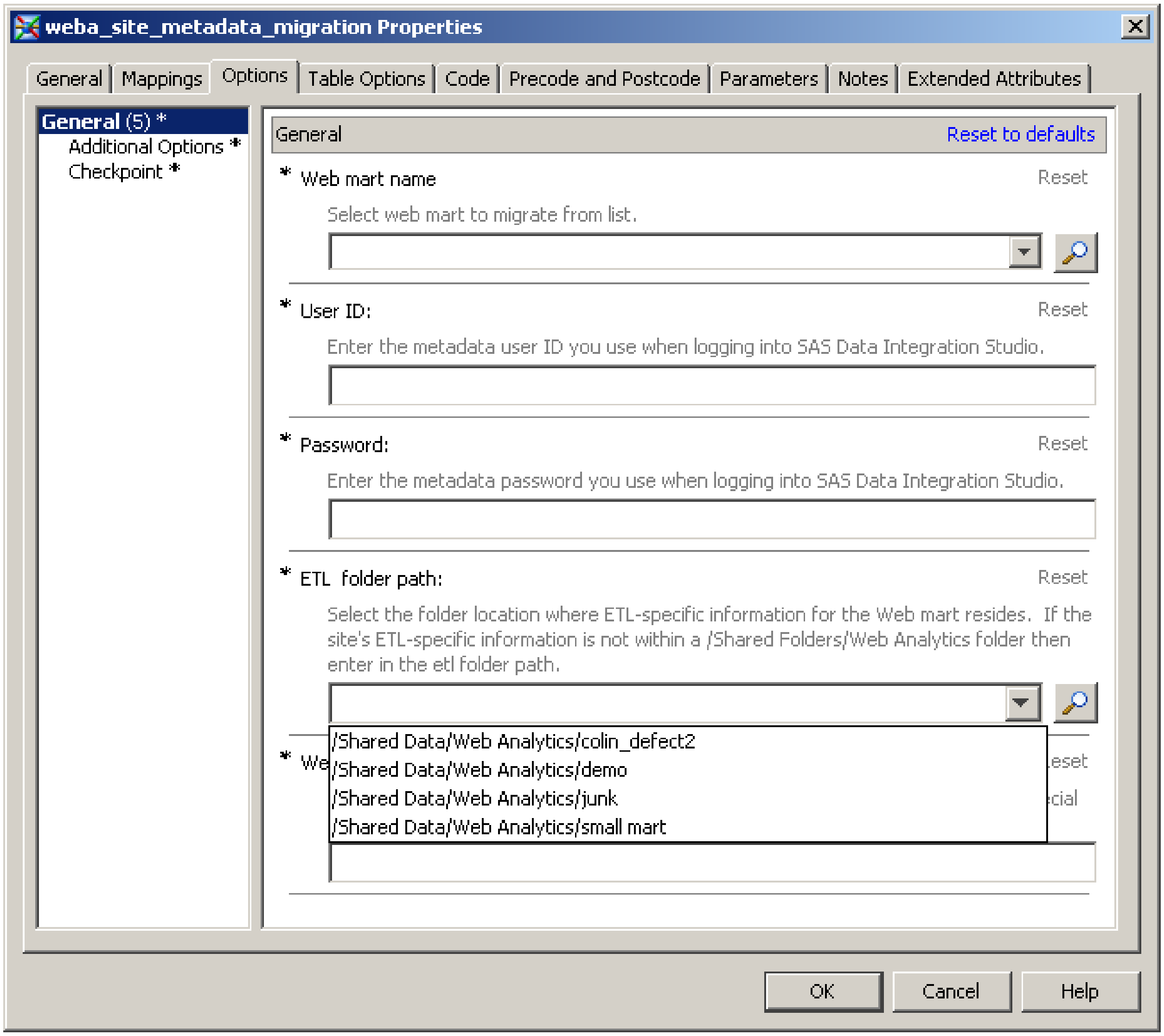
Enter an abbreviation. It cannot contain numbers, blanks, underscores or other special characters.

b. Select the **Web mart name** to migrate.

c. Enter the metadata **User ID** and **Password**.

d. Select the **Metadata Folder** location of the 5.3x ETL jobs. Either select the **/Shared Data/Web Analytics** folder or enter the complete folder pathname.

e. Click **OK**.



* Web mart name

Select web mart to migrate from list.

* User ID:

Enter the metadata user ID you use when logging into SAS Data Integration Studio.

* Password:

Enter the metadata password you use when logging into SAS Data Integration Studio.

* ETL folder path:

Select the folder location where ETL-specific information for the Web mart resides. If the site's ETL-specific information is not within a /Shared Folders/Web Analytics folder then enter in the etl folder path.

/Shared Data/Web Analytics/colin_defect2
/Shared Data/Web Analytics/demo
/Shared Data/Web Analytics/junk
/Shared Data/Web Analytics/small mart

* We

4 If your SAS Server tier runs in a Windows environment, you can run the job in SAS Data Integration Studio. If your SAS Server tier runs in a UNIX environment (including LINUX), deploy the migration job to the UNIX machine and run it. Follow the procedures for the Site Initialization job. For more information, see “Initialize the Site” in Chapter 2.

5 Navigate to the metadata folder that you specified in step 4, and then open the `5.4 Jobs/ETL Jobs` folder.

6 Customize the jobs.

7 Deploy all of the 5.4 jobs. For information about the jobs, see Appendix 7.

8 Delete the YY_* versions of the tables within the Web mart WaMart library.

Note: If a path data set is not migrated, it is retained but named yy_path_yyyymmdd.

9 Delete or rename Wacnfg.Cnfg_Input_History.

10 (Optional) Re-create the aggregates. Skip this step if you do not want to include historical data in the aggregates.

a. Open an interactive SAS session that has access to the SAS Web Analytics macros.

b. In the Program Editor, open the weba_regen_aggregates,sas program. This program is located in the following folders:

Windows	**`!SASROOT\weba\misc`**
UNIX	**`!SASROOT/misc/weba`**

c. Follow the directions in the program for regenerating the aggregates.

11 Migrate the Clickstream job. For more information, see “Upgrade Considerations” in the *SAS Data Surveyor for Clickstream 2.2: User’s Guide*.

12 Schedule the ETL.

Migrating Search Engine Bid Data

Conversion to SAS Unicode Server

To convert the Search Engine Bid Data (SEBD) data mart to SAS Unicode:

1 Complete the SAS Unicode server setup. For more information, see “Introduction to SAS Unicode” in Chapter 1.

2 Archive the current SEBD directory structure, and then delete the SEBD directory. The SEBD directory must be re-created as part of SEBD Initialization.

3 Back up the metadata repository.

4 If changes were made to the SEBD Web Report Studio reports, use SAS Web Report Studio to copy the reports to another folder.

5 If changes were made to any of the SEBD jobs, then copy those jobs to another folder in SAS Management Console.

6 In SAS Management Console, complete these steps:

a. Navigate to the folder where SEBD was installed, and delete the folder.

b. Navigate to /**`Products/SAS Web Analytics/Sample Jobs`** and run the SEBD Initialization job. For information, see “Initialize the Site” in Chapter 2.

Base SAS Changes

If you are migrating SEBD data from 5.3.1 to 5.4, make these updates after you install and configure SAS Web Analytics 5.4, but before you run an SEBD ETL.

Note: First, back up the SEBD directory.

Run the following code within a SAS batch session (<*your-server-installation-location*>**`/config/Lev1/SASApp/sas.bat`** or **`sas.sh`**):

```
/* set sebdcnfg library path, for example c:\data\sebd\sebdcnfg */

%let sebdcnfg_path =;
```

```
/* Create adwords column xml */
%weba_adwords_columns_xml(sebdcnfg=&sebdcnfg_path);
```

For more information about the adwords_col.xml file, see "Google AdWords Data" in Chapter 5.

Troubleshooting Migration

Site Metadata Migration

- ❑ The message "Site already migrated" is displayed when you are running the Site Migration job.

  This message indicates that the metadata has already been migrated. Verify that the following folders exist and are populated.
  /System/Applications/SAS Web Analytics/Web Analytics 5.4/Site/*<Web mart>*

  If that is not the case, restore the metadata repository by using the backup made before the migration process. Also ensure that the Web mart object and Support folder appear within this folder.
  /System/Applications/SAS Web Analytics/Web Analytics 5.3x/Site/*<Web mart>*
- ❑ Errors are generated when you work in UNIX because of spaces that are added to the Web mart name.

  After you make edits to the transformations in the Site Metadata Migration job, the code is saved. Blank spaces are added to the end of the Web mart name, and the name becomes 256 characters long, with); as the last two characters.

```
%let mig_webmart_name = %nrquote(RacesGolden2



                                                        );
```

  When the code is deployed to UNIX, the spaces are truncated, which causes an error.

  To avoid this error, manually remove the white space between the end of the Web mart name and the); characters.

Table Migration

WAMART Tables

- ❑ If users do not have proper authorization, check the operating system authorizations for the user ID that is running the SAS Data Integration Studio job. Make sure the user can modify and create new files within the directory.
- ❑ If there are problems creating the WaMart table, examine the PROC COMPARE output, and then fix the problem.

Deleting Aggregate Tables

If users do not have proper authorization, complete these steps:

1. Fix authorization problems. The steps depend on your operating system.
2. Manually delete the aggr_* data sets within the WaMart library.
3. Verify that the following tables within the WaCnfg library were created:

   Cnfg_metrics should be populated. If the table was not populated, run the following code:

   ```
   libname wacnfg "<wacnfg path>";
      data WaCnfg.cnfg_metrics;
         set sashelp.weba_monitor_metrics(rename=(key=name
            text=label) drop=lineno);
   run;
   ```

   Cnfg_wacmpgoal

   Cnfg_wacmpgoalpage
4. Edit **weba.migration_status.** Open weba.migration_status and edit the following fields:

   Aggr_status – Enter **'complete'**.

   Cnfg_status – Enter **'complete'**.
5. Run the last node within the Site Migration job.

Regenerating Aggregates

❑ The error message "Web mart has not been migrated" is displayed. If you receive this error message, verify the following items:

-The "Complete migration - set version" on the webmart object transformation was not excluded within the migration to 5.4 job.

-There were no error messages in the migration program log.

If both items are verified, complete these steps:

1. Open SAS Management Console.
2. Navigate to the **`/System/Applications/SAS Web Analytics/Web Analytics 5.4/Site/`**<*Web mart*> folder.
3. In the right pane, right-click on the **webmart** object and select **Properties**.

Address any errors in the log.

APPENDIX 7

The ETL Jobs

ETL jobs can be divided into groups according to their function. For a list of ETL Jobs by function, see “Introduction to the SAS Web Analytics ETL Jobs” in Chapter 3.

The following table lists all of the jobs that make up the ETL schedule for SAS Web Analytics. If the value of the Optional column is Yes, then the ETL administrator can decide to run the job. If a job is customizable, then the ETL administrator can update the job to meet the needs of an individual Web site.

Note: Jobs with identical numbers can run simultaneously. For example, there are many jobs numbered “1300.” All of these jobs can be scheduled to run at the same time during ETL, provided that all jobs numbered less than 1300 have successfully completed.

Jobs with an asterisk (*) need performance tuning depending on the size of the Web or tagging logs. For information, see “Performance Tuning Options for the Load Jobs” in Chapter 3.

Table A7.1: WEBA ETL Jobs

Job Name	Dependency	Description	Optional?	Customizable?
Clickstream job or jobs	Web log or tagging log creation	Reads in and groups Web log or tagging log data by visitor. The output data set name must use the 'input filename prefix' defined in the weba_1100_health_check job options (default=weblog_detail).	No	Yes
weba_1100_data_health_check	Clickstream job or jobs that create the input data set. The name must begin with the value of the 'input filename prefix' option (default=weblog_detail).	Checks Web log detail data for domains with missing values and ensures that the current input files have been properly loaded. Reads the input files and performs some mandatory and optional checks.	No	Yes
weba_1101_load_search_engine_dim	None	Single-use job to add more search sites to search_engine_dim table	Yes	Yes
weba_1200_warehouse_staging_tables	weba_1100_data_health_check	Reads the data set validated by the Health Check job using Clickstream jobs and creates all of the necessary staging tables.	No	No
weba_1220_goal_staging_tables	weba_1200_warehouse_staging_tables	Stages data used to load the goal tables.	No	No
weba_1230_link_staging_tables	weba_1100_data_health_check, weba_1200_warehouse_staging_tables	Stages data used to load the link tables.	Yes	No

Job Name	Dependency	Description	Optional?	Customizable?
weba_1300_load_browser_dim	weba_1200_warehouse_staging_tables	Loads the Browser_Dim table with new browsers not already occurring in the table. Performance tuning can occur if necessary.	No	No
weba_1300_load_domain_dim	weba_1200_warehouse_staging_tables	Loads the Domain_Dim table with new domains not already occurring in the table. Performance tuning can occur if necessary.	No	No
weba_1300_load_goal_dim	weba_1220_goal_staging_tables	Loads the goal dimension table.	No	No
weba_1300_load_int_search_term_dim	weba_1200_warehouse_staging_tables	Loads the internal search term dimension table. Requires setting up the Clickstream job to track internal searches.	No	No
weba_1300_load_ip_address_dim	weba_1200_warehouse_staging_tables	Loads the IP address dimension table.	No	No
weba_1300_load_link_dim	weba_1230_link_staging_tables	Loads the link dimension table.	Yes	No
weba_1300_load_link_type_dim	weba_1230_link_staging_tables	Loads the link type dimension table.	Yes	No
weba_1300_load_origination_dim	weba_1200_warehouse_staging_tables	Loads the origination dimension table.	No	No
weba_1300_load_page_dim*	weba_1200_warehouse_staging_tables	Loads the Page_Dim table with new pages.	No	No
weba_1300_load_platform_dim	weba_1200_warehouse_staging_tables	Loads the Platform_Dim table with new platforms not already occurring in the table.	No	No
weba_1300_load_referrer_query_string_dim*	weba_1200_warehouse_staging_tables	Loads the REFERRER_QUERY_STRING_DIM table with new query strings.	No	No

Job Name	Dependency	Description	Optional?	Customizable?
weba_1300_load_search_term_dim*	weba_1200_warehouse_staging_tables	Loads the SEARCH_TERM_DIM table with new search terms.	No	No
weba_1300_load_server_dim	weba_1200_warehouse_staging_tables	Loads the SERVER_DIM table with new servers.	No	No
weba_1300_load_session_x_status_code*	weba_1200_warehouse_staging_tables	Loads the SESSION_X_STATUS_CODE table with new session-status code combinations.	No	No
weba_1300_load_status_codes	weba_1200_warehouse_staging_tables	Loads the STATUS_CODE_DIM table with new status codes.	No	No
weba_1300_load_user_agent_dim*	weba_1200_warehouse_staging_tables	Loads the USER_AGENT_DIM table with new IP addresses.	No	No
weba_1300_load_user_dim*	weba_1200_warehouse_staging_tables	Loads the USER_DIM table with new user identifications.	No	No
weba_1300_load_visitor_dim*	weba_1200_warehouse_staging_tables	Loads the VISITOR_DIM table with new visitor IDs. Visitor ID is assigned within the Clickstream jobs.	No	No
weba_1305_load_referrer_dim*	weba_1300_load_domain_dim	Loads the REFERRER_DIM table with new referrer URLs.	No	No
weba_1310_load_session_fact*	weba_1200_warehouse_staging_tables	Updates the SESSION_FACT table with new session data.	No	No
weba_1315_load_session_attribution	weba_1200_warehouse_staging_tables	Loads the session attribution table.	No	No
weba_1320_load_detail_fact*	weba_1200_warehouse_staging_tables	Updates the DETAIL_FACT table with new session detail data.	No	No
weba_1330_load_detail_x_link	weba_1230_link_staging_tables	Loads the detail_x_link table.	No	No

Job Name	Dependency	Description	Optional?	Customizable?
weba_1330_load_goal_x_session	weba_1220_goal_staging_tables	Loads the goal_x_session table.	No	No
weba_1410_profile_search_engines	weba_1320_load_detail_fact weba_1310_load_session_fact	This is an example profile job. See "Profile Jobs" in Chapter 3.	Yes	Yes
weba_1500_agg_int_searches	weba_1200_warehouse_staging_tables	Updates the Aggr_Int_Search_&interval tables.	No	No
weba_1500_agg_search_terms	weba_1310_load_session_fact	Updates the Aggr_Term_&interval summary tables.	No	Yes
weba_1600_active_funnels	weba_1320_load_detail_fact weba_1310_load_session_fact	Updates one Waanly data set for each active funnel, as defined by the funnel configuration metadata.	No	Yes
weba_1600_agg_visitor	weba_1600_active_funnels	Updates the Aggr_Visitor_&interval tables.	No	Yes
weba_1800_agg_hourly_status	weba_1600_agg_visitor	Updates the Aggr_Hourly_Status_&interval data sets.	No	Yes
weba_1900_agg_page_status	weba_1800_agg_hourly_status	Updates the Aggr_Page_&interval, Aggr_Bounce_&interval, Aggr_Exit_Pg_&interval, Aggr_Entry_Pg_&interval, and Aggr_Status_&interval tables.	No	Yes
weba_2000_agg_browsers_platforms	weba_1900_agg_page_status	Updates the Aggr_Platform_&interval and Aggr_Browser_&interval tables.	No	Yes
weba_2050_agg_general_statistics	weba_2000_agg_browsers_platform	Updates the Aggr_Daily_Total_&interval and Aggr_Hourly_Total_&interval tables.	No	Yes

Job Name	Dependency	Description	Optional?	Customizable?
weba_2100_agg_sebm	weba_2000_agg_browsers_platforms	Updates the Aggr_Sebm_&interval tables.	No	Yes
weba_2150_agg_seb_campaigns	weba_2100_agg_sebm	Updates the Aggr_seb_Campaign_&interval tables.	No	Yes
weba_2200_create_path_data	weba_2150_create_path_data	Updates the path analytical tables.	No	Yes
weba_2550_create_cube	weba_2100_agg_sebm and completed load	Creates GoalPageScore cube.	Yes	No
weba_2500_insight_monitor1	weba_1200_warehouse_staging_tables	Creates daily statistics summary used in insight or monitor reports.	No	Yes
weba_2500_insight_monitor2	weba_1200_warehouse_staging_tables	Creates referrer summary used in insight or monitor reports.	No	No
weba_2501_prep_insight_monitor	weba_2500_insight_monitor1 weba_2500_insight_monitor2	Creates waanly.anly_daily_metrics, plus weekly and monthly versions.	No	Yes
weba_2575_cube_update	weba_2550_create_cube weba_2100_agg_sebm	Updates the GoalPageScore Cube.	Yes	No
weba_2600_etl_done	weba_2501_prep_insight_monitor	Removes input data and allows another ETL job to begin executing (optional).	No	No
weba_9999_maxmind_load	Not applicable	Loads MaxMind geographic data. MaxMind is a recommended source of Internet Protocol (IP) address information that is associated with a geographic location. MaxMind also uses the U.S. Heat Map report. If you do not want to use the U.S. Heat Map report, do not run this job.	Yes	Yes

Jobs That Require the SAS Tag

The following ETL jobs require the SAS Tag. The jobs are required only when tagging server logs are being processed.

- weba_1230_link_staging_tables
- weba_1300_load_link_dim
- weba_1300_load_link_type_dim

Using the weba_run_etl Script

For large sites, it is important to schedule your ETL jobs so that groups of jobs can run concurrently. However, for small sites and for start-up testing, you might prefer to run your ETL sequentially. SAS Web Analytics provides a script that enables you to do this easily. The program weba_run_etl.sas can be found in the following locations:

Windows

```
<sasroot directory>\SASFoundation\9.2\weba\sasmisc
```

UNIX

```
<sasroot directory>/SASFoundation/9.2/misc/weba
```

Note: In SAS Web Analytics 5.4, this script cannot be used if the Web mart name contains the following string: `.sas`

In addition, this script cannot be used if the folder name for the Web mart or the deployed job contains blanks.

To use the program:

1. Make a copy of the weba_run_etl.sas program in a subdirectory of your Web mart's main directory.

   Tip: Create a subdirectory named **sas** under your Web mart's main directory and store the script in it.

2. Deploy the jobs for your Web mart.

   Tip: Create a subdirectory named **deployed** under the directory in which the weba_run_etl.sas program resides, and deploy the jobs to that location.

3. Edit the weba_run_etl script, supplying valid values for all parameters in the section **Edit these parameters as needed**. Explanations and examples for each parameter are provided in the program. Save your changes.

4. Run the script as a batch job, or submit the script in an interactive SAS session. The SAS session that is used to run the script must have access to the SAS Web Analytics macros. The two easiest ways to run the script are shown here:

   - ❑ Use the SAS Batch Server from your SASApp server context.
   - ❑ Use the SASApp server script to initiate the interactive SAS Session.

   You can also use point the –config option to the sasv9.cfg file in the SASApp folder along with the standard SAS executable program. For Windows, use sas.exe. For UNIX, use sas. For more information, see "Accessing SAS Web Analytics Macros" in Appendix 4.

The script sequentially executes each job that is stored in the directory that contains the deployed jobs.

The logs from the individual jobs are written to a subdirectory under your Web mart's main directory named **`Logs`**.

APPENDIX

8 Defining Profiles

Overview

Before a profile can be used to filter analysis results, it must first be defined. The definition process involves two steps:

1. Add a row to the Profile_Dim table.
2. Create a job that runs during ETL to add rows to the Profile_X_Session table.

An administrator must create a new SAS Data Integration Studio job for each profile. Each profile job can reference only one site. Profile jobs can be combined in SAS Data Integration Studio, but not divided.

The administrator is responsible for using the default SQL join transformation or any custom code to produce a table that contains a list of Session_Sks that fit the profile. The records are then appended to Profile_X_Session and Profile_Dim as necessary.

To create a profile:

1. Define the profile. The ETL administrator working with a business user needs to define a set of rules used to group visits.

   For example, an analyst would define the following criteria (rules) for a profile:

   - □ The profile is used by browsers with a name of “Internet Explorer.”
   - □ The profile must have no referrer, where referrer_desc equals NO REFERRER.
   - □ Visits occurred between Jan 1, 2005, and June 1, 2006.
   - □ Where page_desc Z was hit at some point during the session.

2. Create a job. The ETL administrator needs to create a job that will create a data set. The data set is a list of visits that meet the criteria defined by the analyst.

In the preceding example, the ETL administrator joins SESSION_FACT with BROWSER_DIM, DATE_DIM, and REFERRER_DIM. BROWSER_DIM contains the browser_nm (browser name) field, and REFERRER_DIM contains the referrer_desc field. DATE_DIM houses the cal_dt field, and SESSION_FACT contains the session_sk field as well as keys to the necessary dimensions.

Inside the transformation or code, the ETL administrator must add any necessary “where” logic: Date_Dim.Cal_Dt must be between Jan 1, 2005, and June 1, 2006; Browser_Dim.Browser_Nm must equal Internet Explorer; and Referrer_Dim.Referrer_Desc must equal NO REFERRER. The administrator’s query should produce a list of Session_Sks in an output table that can be used as input to the profiling transformation. The SQL Join utility in SAS Data Integration Studio provides a point-and-click interface to query construction.

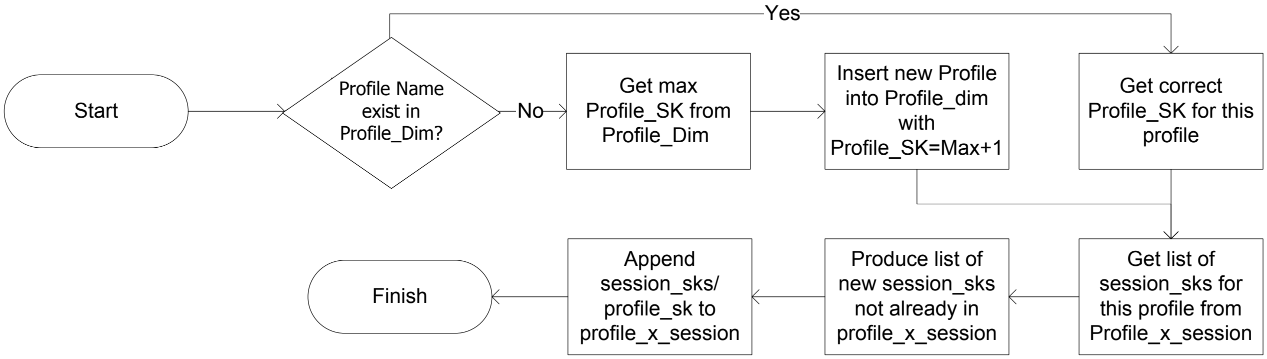
A subquery is also necessary to identify sessions that accessed page_desc Z. The user also needs to create an inner join for the matches between the preceding query and the subquery. The subquery pulls Page_Desc_Sks from PAGE_DESC_DIM (where page_desc=’Z’) and matches the results with Page_Desc_Sks from the DETAIL_FACT table. It keeps only the unique Session_Sks. SQL join transformations or other code can accomplish this task. Administrators can also manipulate data used in various stages throughout the ETL process to improve performance, such as by flagging records on input data and reusing the data. Administrators must be aware of the possible problem with this approach to avoid the application of ETL business rules that might occur throughout the process. SAS Web Analytics recommends using WaWork or WaMart data exclusively for this purpose. If users use WaSrc input data directly for this job, they will bypass the business rules that are applied during ETL.

The ETL administrator customizes the profiling transformation with profile-specific information. All available options are listed in the following table. The flowchart in Figure 3.1 outlines tasks that the transformation takes to load tables appropriately. Note that the profile has one input and two outputs. The input to the job contains the list of Session_Sks to append, whereas the outputs are the Profile_X_Session and Profile_Dim tables.

Table A8.1: Profiling Options

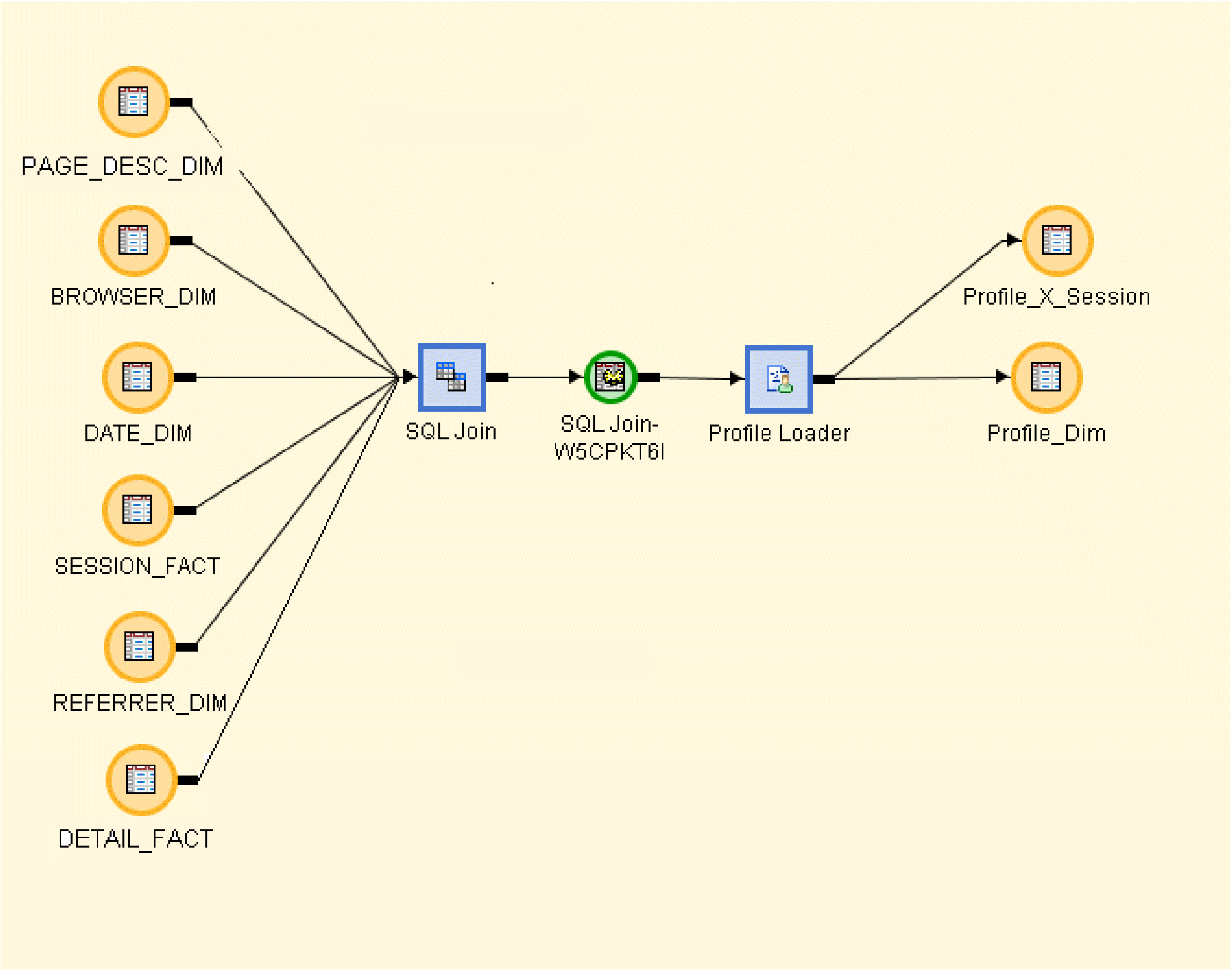
Option	Valid Values	Description
Profile Name	Any 40 characters	The name of the profile as it is stored in the PROFILE_DIM table. If the name changes, then it is treated as a new profile. Each profile must have a unique name.
Profile Description	Any 100-character value	The description that is stored in the PROFILE_DIM table that corresponds to the name.
Drop old records?	Yes, No	If set to YES, then Profile_X_Session is purged of old data for this profile before loading. If NO, then new records are appended to Profile_X_Session. Moreover, a setting of YES without a specified input purges the Profile_Dim and Profile_X_Session of information related to the profile.
Bulkload	Yes, No, Conditional	Set to YES to inserts the records by dropping constraints and streaming records into the relational database management system (RDBMS) bulkload utility. Set to NO to generate one insert statement for each record. For information, see your database's documentation on bulkload. Conditional means that the following threshold applies. If the staging table has fewer records than the threshold, then the records are loaded without bulkload. If the threshold is exceeded, then bulkload is turned on. This setting has no effect if the warehouse is in SAS data sets.
Threshold	(Nonzero number)	Only operational if bulkload is set to conditional. If the staging table has more records than the threshold, then the records are bulk loaded. Otherwise, they are inserted. This has no effect if the warehouse is in SAS data sets.
DropIndexes?	Yes, No	If set to YES, then all indexes are dropped before loading. Any indexes that exist on the table at the time the indexes were dropped will be recreated. If any indexes were previously created in SAS Data Integration Studio and the General Options-Additional Options 'Generate indexes on target table=Yes' option is set, be sure to reset this to 'Generate indexes on target table=No'. You need to do this or an error will occur as an attempt is made to create the indexes after the DropIndexes feature has already recreated them. If DropIndexes is set to NO, then SAS does not do anything to indexes before loading.

Figure A8.1: Profile Task Flow


Yes
Start
Profile Name exist in Profile_Dim?
No
Get max Profile_SK from Profile_Dim
Insert new Profile into Profile_dim with Profile_SK=Max+1
Get correct Profile_SK for this profile
Get list of session_sks for this profile from Profile_x_session
Produce list of new session_sks not already in profile_x_session
Append session_sks/ profile_sk to profile_x_session
Finish

The complete SAS Data Integration Studio job flow appears similar to Figure A8.2.

Figure A8.2: SAS Data Integration Studio Profiling Job



Modifying Profiles

Profiles can be modified and redeployed for scheduling, but you need to determine whether the old rules apply to past history, or whether the new rules should apply to past history. If old rules apply to past history, then you can change the rules in the Join logic and redeploy the job. This assumes that the “Drop old records” option is set to No. If the option is set to Yes, then the option needs to be changed accordingly before the rule change and redeployment.

If new rules apply to past history, then the past history needs to be replaced. Make the required business rule changes and set “Drop old records” to Yes and run the job once. The warehouse is then updated to reflect new business rules on past history. You now have a choice of changing the “Drop old records” option to No. Redeploy the job to pick up changes for future runs.

Deleting Profiles

Removing a profiling job from the scheduled flow halts updates to the profiling tables, but it does not remove past history. You can determine whether the profile needs to be unscheduled if the profile no longer needs to be updated, or whether the profile's data should be removed. The process described in this section is recommended for dismantling a profiling job and removing its data from the warehouse.

To remove a job from the warehouse ETL schedule and remove the profile's definition from the warehouse:

- ❑ Unschedule the job. Use SAS Management Console and LSF flow manager to remove the job from the flow and reroute dependencies accordingly.
- ❑ Remove all inputs to the profile loader object.
- ❑ Run the profile interactively with the replace option set to YES. Running the profile without an input purges warehouse tables of that profile's information.
- ❑ Check the log for errors. If there are no errors, delete the job. If there are errors, fix them and rerun as necessary.

Profile Scheduling

The administrator must schedule a process and include it in a job flow in order to enable incremental updating. When contemplating placement in a process flow, it is very important to note that profiling jobs *must* execute after *all* warehouse ETL has taken place. If this rule is violated, then Profile_X_Session might contain incorrect, incomplete results or violations of referential integrity.

To guarantee referential integrity, set Bulkload to No. This uses the constraints on the appropriate RDBMS tables and logs any violations of referential integrity and generates RDBMS errors. A single error will cause the job to fail. For information about bulkload options, see the SAS/ACCESS Interface to Oracle chapter in *SAS/ACCESS 9.2 for Relational Databases: Reference*.

Performance Considerations

Profiling processes can be performance-intensive. Therefore, reconstructing queries might lead to performance gains. Incremental loads can take advantage of staging tables to limit the amount of data to process to new sessions only. To do so, build the profile normally and run it once. Replace the input detail, or the session fact table, or both with its corresponding table from the staging area, and schedule the job. Fact tables in the staging area always contain only sessions that have been added to the data warehouse.

Create a Job to Run during ETL

After a profile has an entry in the Profile_Dim table, you must create your own job to run during the ETL process (for example, the Weba_1410_Profile_Search_Engines job). The purpose of this job is to use information from the current ETL to create one row for every session that meets the profile’s criteria. These rows are then loaded into the Profile_X_Session table.

For example, the criterion might be the search engine for the session's first request, which is indicated by the Search_Engine_Sk value in the session's Session_Fact row. If the search_engine_sk is for Google (its Search_Engine_Desc field contains 'Google'), then the profile’s condition has been met. The session is associated with the profile by a new row in the Profile_X_Session table.

The Profile_X_Session table contains the following columns:

Session_Sk: the session surrogate key; an integer value that identifies a row in the Session_Fact table.

Profile_Sk: the profile surrogate key; an integer value that uniquely identifies a row in the Profile_Dim table.

A Session_Sk can occur more than once in the Profile_X_Session table if it meets the criteria for more than one profile. A session_sk does not occur in the Profile_X_Session table if it fails to meet the criteria for any active profile definition.

APPENDIX 9

Recommended Reading

Recommended Reading

Here is the recommended reading list for this title:

- *SAS BI Dashboard: User's Guide*
- *SAS Data Integration Studio: User's Guide*
- *SAS Data Surveyor for Clickstream Data: User's Guide*
- *SAS Information Map Studio: Getting Started with SAS Information Maps*
- *SAS Intelligence Platform: Security Administration Guide*
- "Administering SAS Enterprise Guide" in the *SAS Intelligence Platform: Desktop Application Administration Guide*
- "Administering SAS Information Map Studio" in the *SAS Intelligence Platform: Desktop Application Administration Guide*
- "Administering SAS Business Intelligence Dashboard" in the *SAS Intelligence Platform: Web Application Administration Guide*
- "SAS Web Report Studio Administration" in the *SAS Intelligence Platform: Web Application Administration Guide*
- *SAS Language Reference: Dictionary*
- *SAS Web Report Studio: User's Guide*
- *Scheduling in SAS 9.2*
- *Tutorial: Getting Started with SAS Enterprise Guide*

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

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

Customers outside the United States and Canada, please contact your local SAS office for assistance.

Glossary

active session
a session that is still in progress for either of the following reasons: 1) the visitor is currently requesting resources such as pages, pictures, or files at the Web site, or 2) the visitor has not made any requests for a while, but the session timeout (usually 30 minutes) has not been reached.

AdWords
Google's pay-per-click (PPC) advertising product. AdWords advertisers create and budget campaigns that contain keywords that trigger their ads. The ads for relevant searches are displayed as Sponsored Links.

authenticated realm
a group of Web pages that are accessible only to users who have authenticated themselves in some way. Users typically authenticate themselves by entering a user ID and a password.

bytes received
the number of bytes that a Web server has received from a particular client browser. Most Web server log files do not record bytes received. See also bytes sent.

bytes sent
the total number of bytes that a server has delivered in response to a request. Because of retransmissions and network problems, bytes sent can sometimes be larger than the size in bytes of the resource or file that was received. Bytes sent is sometimes referred to as bytes transferred.

click-through
the act of navigating to a Web page by selecting a link in an e-mail message.

clickstream analysis
the analysis and interpretation of the actions of Web site visitors. These actions are recorded in the Web log as a chain of time-ordered related events, such as a trail of mouse clicks that a visitor leaves. The purpose of clickstream analysis is to understand and predict the actions of visitors as well as the paths that visitors take through a site. This analysis typically involves data-mining techniques such as identifying sequences and associations.

clickstream reporting
the process of summarizing the actions that are recorded in a Web server log file into various classes, dimensions, or buckets. The summarization is based on the visitors' URLs, the amount of time spent on each page, and elements of the domain names. This reporting describes demographic information about the visitor population, the site activity rates, and the relative demand for various areas of the site, such as ad banners or links on a page.

content group
a collection of pages within a Web site that share similar content or provide similar services to visitors. This subset of the Web site can be considered as a separate Web site in order to enable Web analyses that are usually available only for an entire Web site.

content type
a value that tells a client's Web browser how to interpret and display a transferred object such as an image file, a sound file, or a video file. For example, GIF, JPG, TIFF, MIDI, and WAV are content types.

domain
a database of users that has been set up by an administrator by using a specific authentication provider such as LDAP or the host operating system. The domain name should be unique within your enterprise. For example, you should not have a Windows domain and a UNIX domain that are both named “SALES.”

entry page
the first page that a visitor views when entering a Web site.

entry point
the first page that an Internet visitor views when visiting a Web site. In SAS Web Analytics, the entry point page marks the start of a session. See also exit point.

ETL
see extract, transform, load.

exit page
the last page that a visitor views before leaving a Web site.

exit point
the last page that a visitor views before leaving a Web site. In SAS Web Analytics, the exit point marks the end of a session.

extract, transform, load
a data warehousing process in which data is extracted from outside sources, transformed according to operational and quality needs, and loaded into a target database.

file count
the total number of files that a particular Web site visitor downloads during a session. See also hit and page request.

file hit
See hit.

hit

the result of a successful request (sent to a Web server) for a resource such as an HTML page, a GIF file, or an executable file. Each hit generates an entry in a Web server log file. By contrast, a page request (a particular type of hit) does not include the objects on the page. Requests for an HTML file and a GIF file are both considered to be hits, but only the request for the HTML file is typically considered to be a page request. See also page request.

IP address

a unique network address that is assigned to each computer that is connected to the Internet. The IP address can be specified in either of two formats: Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6). The IPv4 format consists of four parts in dot-decimal notation, as in 123.456.789.0. The IPv6 format can consist of up to eight groups of four hexadecimal characters, delimited by colons, as in FE80:0000:0000:0000:0202:B3FF:FE1E:8329.

measure data item

a classification of data items. The values of measure data items are aggregated (unless otherwise specified) and can be used in computations or analytical expressions.

metric

any standard of measurement that is used as a basis for evaluation or comparison. For example, ROI (return on investment) is a metric that is commonly used by businesses as a basis for making decisions, and bytes per second throughput is a common performance metric.

organic search result

the method by which a user enters a Web site. The user accesses the Web site through a link within a search engine result page that is not a paid keyword link.

organization

the company, institution, or other collective group with which a Web site visitor is affiliated. The organization is determined by converting the client computer's numeric IP address to the domain name, which usually contains either the company name or a recognizable abbreviation.

organization type

the last segment of a domain name, which identifies the type of organization with which a Web site visitor is affiliated. For example, the organization type .COM indicates a commercial business; .GOV indicates a government organization; and .EDU indicates an educational institution. The organization type can be determined by converting the numeric IP address of the visitor's computer to the domain name.

page count

the total number of pages identified in a Web server log file. The page count does not include objects on a Web page, such as GIF files or audio files. Page count and page views are synonyms.

page request

an attempt to access a Web page. Each page request generates an entry in a log file. Unlike a hit, a page request does not include the objects on the page, such as GIF files and audio files. A hit includes all objects on the page as well as the page itself. See also visit and hit.

portlet
a Web component that is managed by a Web application and that is aggregated with other portlets to form a page within the application. Portlets can process requests from the user and generate dynamic content.

referrer
a Web page that provides a link to another page.

SAS Stored Process
a SAS program that is stored on a server and defined in metadata, and which can be executed by client applications. Short form: stored process.

Search Engine Bid Management
the analytics used to optimize the keywords that are selected within search engine marketing campaigns.

search engine bid optimization
the process of refining the keywords used within search engine marketing campaigns to optimize the number of keyword auction wins within a marketing budget. Increasing the number of auction wins increases the number of times campaign ads are displayed for search engine users, and therefore reduces the overall marketing cost of the campaign.

session
a single period during which a software application is in use, from the time the application is invoked until its execution is terminated.

session ID
a unique number that is assigned to a Web site visitor and which is used to track the visitor's path and the time of entry and exit.

session start
the time of a visitor's request for an entry point page of your Web site. The session start time is recorded in the Web server log file.

status code
in a Web server log file, a three-digit code that the server issues to describe the success or failure of a visitor's request for a file from a Web site.

stored process
see SAS Stored Process.

Uniform Resource Identifier
see URI.

Uniform Resource Locator
see URL.

unique visitor
an individual visitor to a Web site. Unique visitors can be identified by various methods, such as an IP address+user agent, a cookie, or a login ID. Depending on how a visitor is identified, the visitor's identity might or might not correspond to an actual person.

unique visitors
the number of individuals who visit your Web site within a specified reporting period (hour, day, week, or month, depending on the report that you select). A unique visitor can have more than one session during the reporting period.

URI
a string that identifies resources such as files, images, and services on the World Wide Web. A URL is a type of URI.

URL
a character string that is used by a Web browser or other software application to access or identify a resource on the Internet or on an intranet. The resource could be a Web page, an electronic image file, an audio file, a JavaServer page, or any other type of electronic object. The full form of a URL specifies which communications protocol to use for accessing the resource, as well as the directory path and filename of the resource. Short form: URL.

visit
an instance of a person who uses a Web browser to access a Web site.

visitor
an inferred individual (derived by measuring browsers filtered for spiders and robots), within a designated reporting time frame, with activity that consists of one or more interactions with a site.

Web funnel
a report that analyzes a visitor's progression through a specified sequence of pages. Visitor inflow and outflow is shown for each step in the sequence.

Web page overlay
a report that displays selected Web metrics overlaid on a Web page.

Web path
a sequence of page views that visitors traverse on a Web site.

Web path report
a report that analyzes the sequence of page views that visitors take on a Web site.

Web performance insight
a report that analyzes and determines the statistically significant metrics that drive a target metric while showing the forecast and historical predictions for the metrics. It is used for decision support and what-if analysis.

Web performance monitor
a report that analyzes and monitors the performance of metrics, and shows forecasts, trends, and performance metrics against a goal. It is used for decision support, spotting business trends, and tracking business objectives.

Web site
a collection of Web pages, or the data that is stored within the Web site's data mart, which contains Web log information.

Index