THE POWER TO KNOW.®

SAS® Web Analytics 5.3
Administrator's Guide

Third Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® Web Analytics 5.3: Administrator's Guide, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® Web Analytics 5.3: Administrator's Guide, Third Edition

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

1st electronic book, June 2010

Contents

What's New in the SAS Web Analytics 5.3 Administrator's Guide

Overview

Web Analytics 5.3.3 has the following new features and enhancements:

- ❑ parameters that control data that is used for Path reports
- ❑ a fully functional Google AdWords 2009(09) JAR file
- ❑ procedures for adding search sites
- ❑ updated procedures for removing Web sites
- ❑ capabilities for regenerating aggregates for Oracle data
- ❑ integration with SAS Customer Intelligence products
- ❑ support for SAS Unicode server
- ❑ updated procedures for extracting Google AdWords search engine bid data
- ❑ a new Clickstream Page Tag job for preprocessing
- ❑ procedures for migrating Web marts to 5.3.3

Parameters That Control Data Used for Path Reports

The weba_2200_create_path_data job has parameters that control several important characteristics of the data that is used for Path reports. You can use SAS Data Integration Studio to modify these parameters to make more or less data available for path analysis.

Fully Functional Google AdWords 2009(09) JAR File

A fully functional Google AdWords 2009(09) JAR file is available on the data (server) tier so that you can access additional features of the Google AdWords API.

Procedures for Adding Search Sites

You can update the SEARCH_ENGINE_DIM table to include new search sites such as YAHOO or Lycos. You can update this manually or programmatically.

Updated Procedures for Removing Web Sites

You can remove a Web site by running a macro to delete its metadata, or by manually deleting its metadata.

Capabilities for Regenerating Aggregates for Oracle Data

You can now edit the aggregate transformation to reload Oracle aggregate tables with all of the history that is currently stored in the warehouse tables.

Integration with SAS Customer Intelligence

SAS Web Analytics provides analysis of 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 in an e-mail broadcast to the customer by SAS Digital Marketing or by SAS Real-Time Decision Manager as an offer for single channel and multi-channel campaigns. If an e-mail recipient opens the e-mail and clicks the link to a target Web site, SAS Web Analytics can track online behavior for the campaign.

Using campaign identifiers, SAS Web Analytics can update the response history data on the machine that is running SAS Marketing Automation for campaigns whose Web channel goals have been reached. The new ETL job that updates the response history data is weba_2550_ma_response_history_update.

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. This association enables the Web analyst to determine how SAS Customer Intelligence campaigns are driving traffic to Web sites and how well campaigns are driving actions within Web sites.

Support for SAS Unicode Server

With SAS Unicode server support, the SAS Web Analytics Web application provides a multi-lingual user interface that now operates consistently with the locale and language settings of the browser. SAS Web Analytics supports languages with multi-byte character set (MBCS) encodings.

New Clickstream Page Tag Job

The Clickstream Page Tag job, weba_0050_page_tag_log_detail_preprocessing, prepares the clickstream detail data set Tagged_DDS for processing. It removes unnecessary Event variable records that contain the values CLICK and SUBMIT from the Tagged_DDS data set.

Migrating Web Marts from 5.3 to 5.3.3

New detailed instructions for migrating your Web marts to SAS Web Analytics 5.3.3 are included in this document.

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.

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 historical 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 5.3.3 Architecture

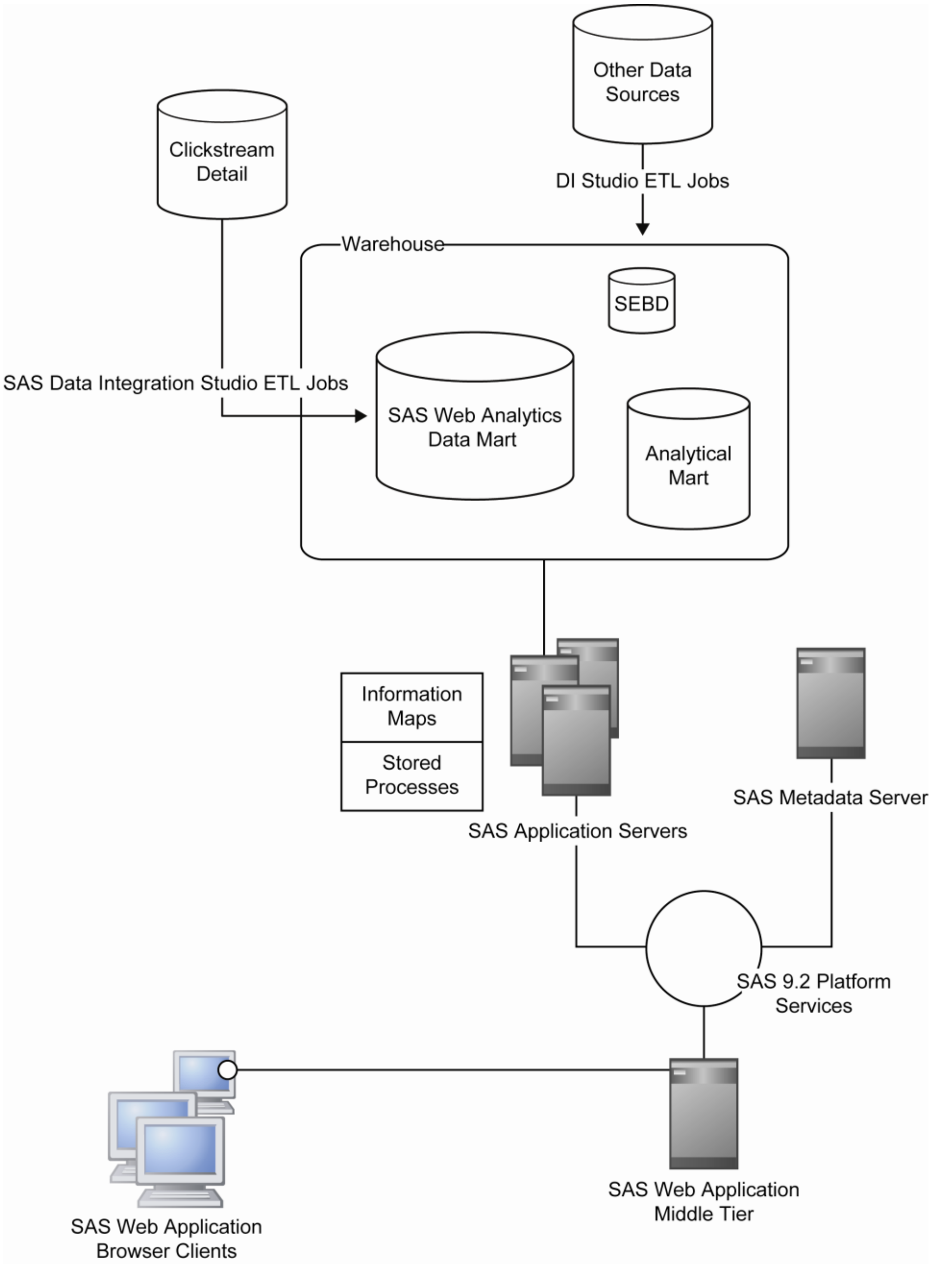
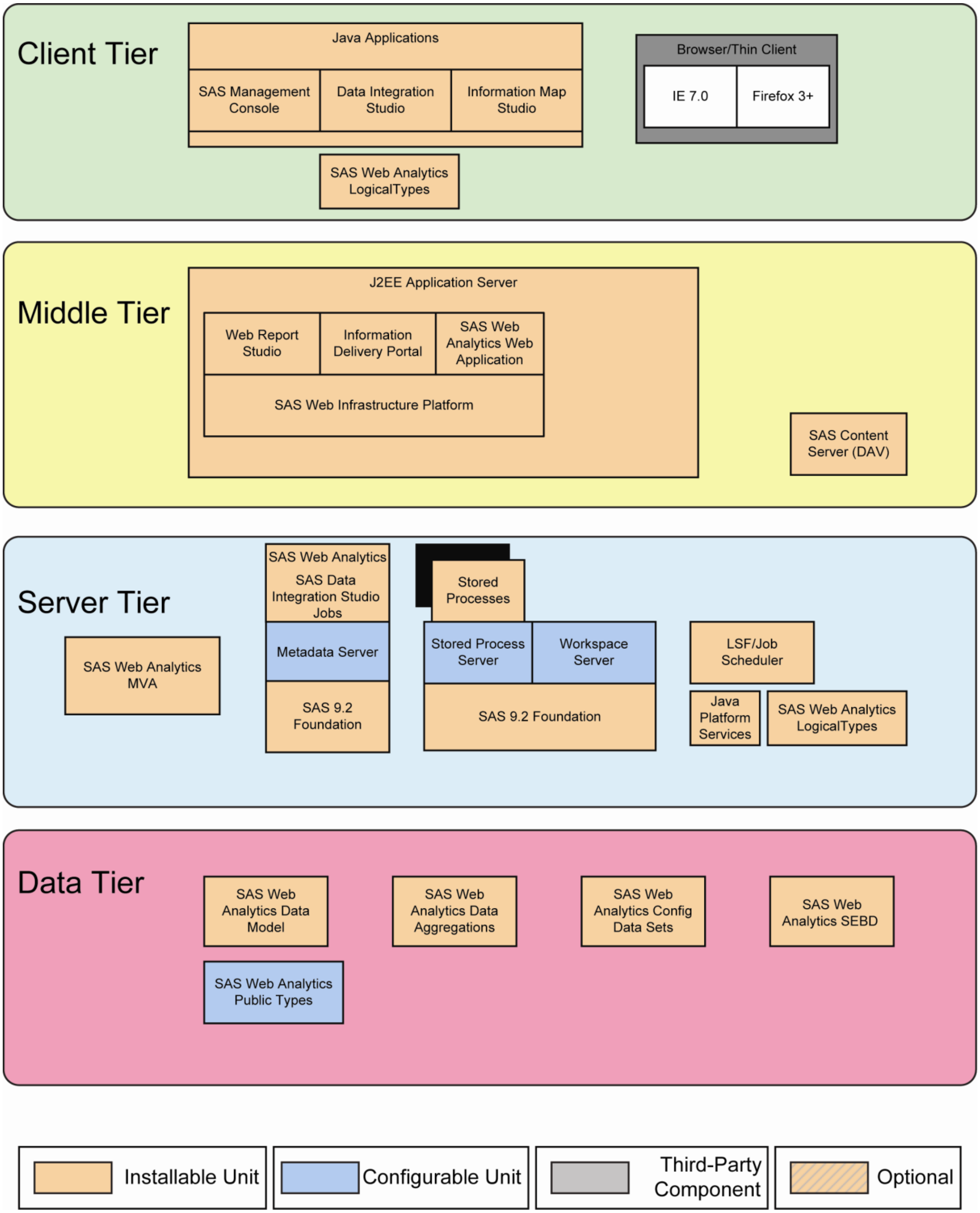


Figure 1.2: SAS Web Analytics 5.3.3 Topology



Accessibility Features of SAS Web Analytics

A complete list of accessibility features is available in the Introduction of the *SAS Web Analytics 5.3 User’s Guide, Second Edition*. You can access the guide in the following ways:

- ❑ Go to **http://support.sas.com/documentation/onlinedoc/webanalytics/**.
- ❑ Select **Contents** from the **Help** menu within the SAS Web Analytics 5.3.3 software. You can select a PDF or HTML version of the User’s Guide.

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.

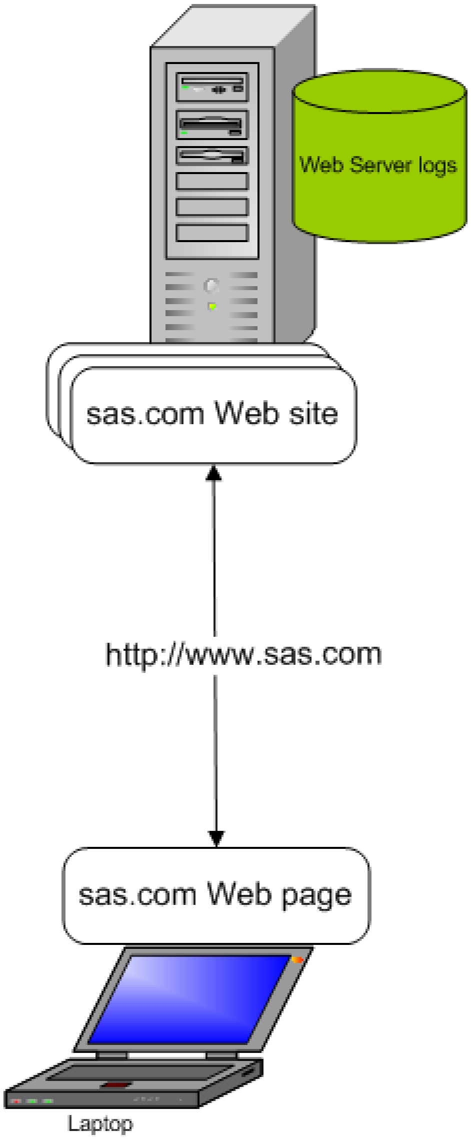
Example: Connect Back-End Data to a Site Visit by Using a Query String

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

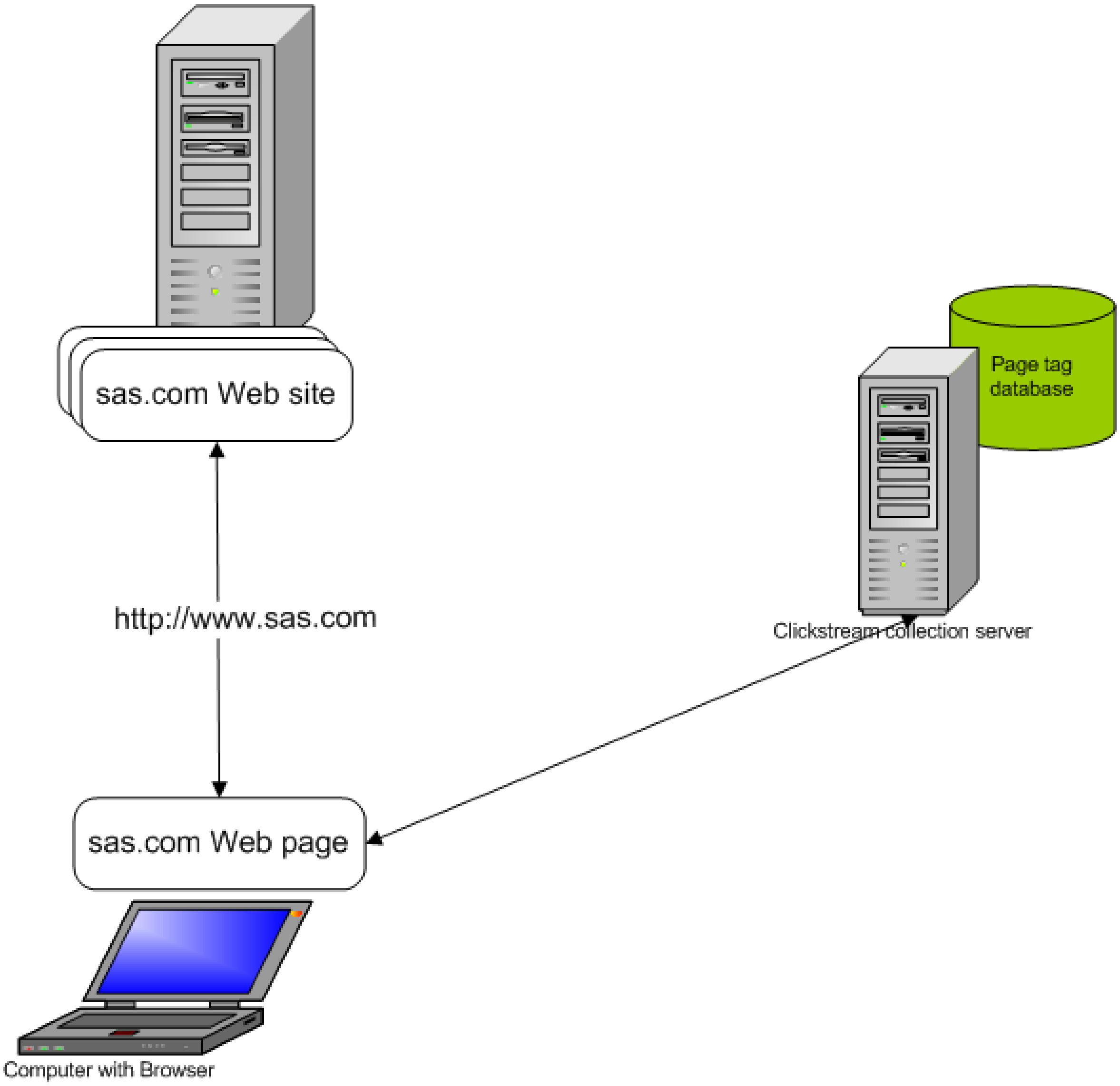
Web data consists of Web site page requests by a Web browser to a Web server. An individual page request can trigger requests within the Web server for the GIF, JPEG, JavaScript, and other files associated with the page request. All of the file requests are recorded in a Web server log, which can be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



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



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 2.1 and the *SAS Data Surveyor for Clickstream Data 2.1: 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 the Windows environment 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).

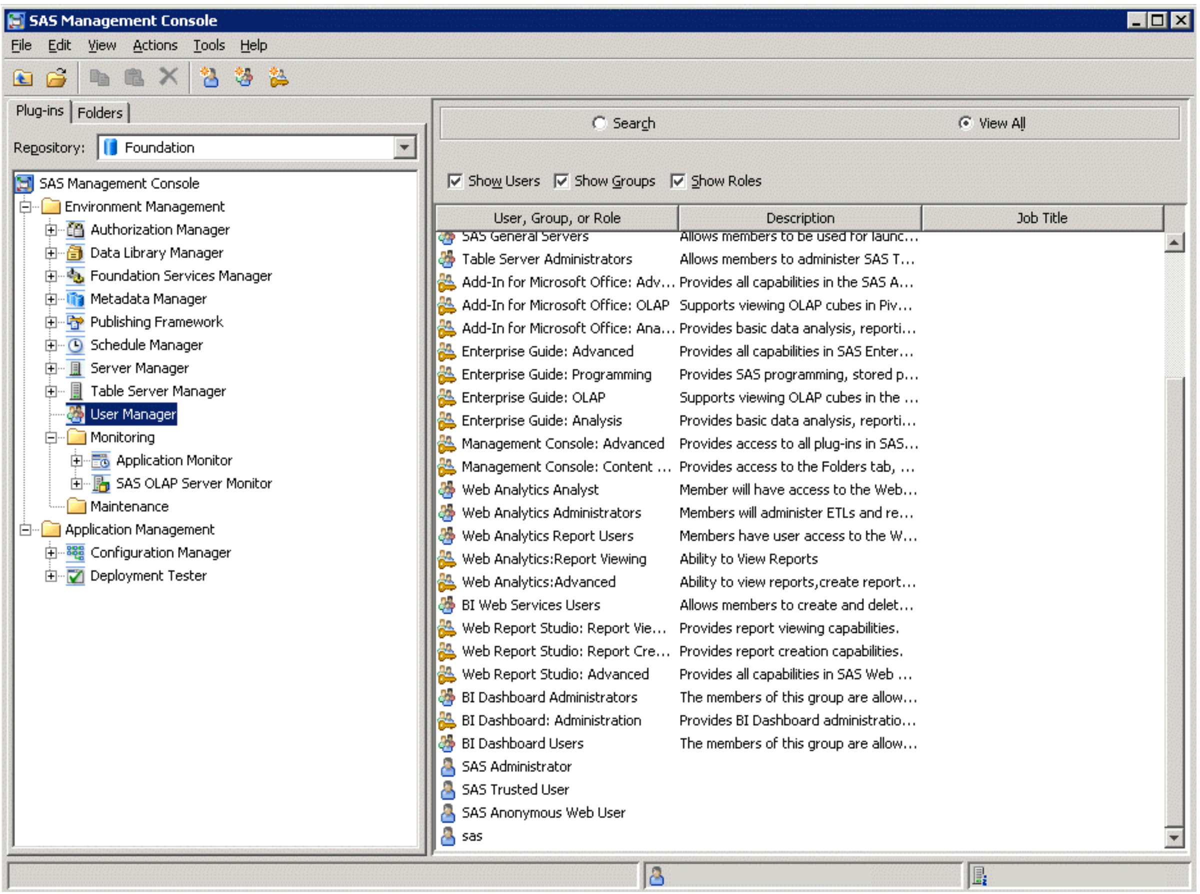
For information about Google AdWords, keywords, or keyword auctions, go to **http://www.google.com/corporate/** and click **Google AdWords**. For information about the Google AdWords API, go to **http://code.google.com/apis/adwords/docs/developer/index.html**.

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
SAS General Servers	Allows members to be used for launc…	
Table Server Administrators	Allows members to administer SAS T…	
Add-In for Microsoft Office: Adv…	Provides all capabilities in the SAS A…	
Add-In for Microsoft Office: OLAP	Supports viewing OLAP cubes in Piv…	
Add-In for Microsoft Office: Ana…	Provides basic data analysis, reporti…	
Enterprise Guide: Advanced	Provides all capabilities in SAS Enter…	
Enterprise Guide: Programming	Provides SAS programming, stored p…	
Enterprise Guide: OLAP	Supports viewing OLAP cubes in the …	
Enterprise Guide: Analysis	Provides basic data analysis, reporti…	
Management Console: Advanced	Provides access to all plug-ins in SAS…	
Management Console: Content …	Provides access to the Folders tab, …	
Web Analytics Analyst	Member will have access to the Web…	
Web Analytics Administrators	Members will administer ETLs and re…	
Web Analytics Report Users	Members have user access to the W…	
Web Analytics:Report Viewing	Ability to View Reports	
Web Analytics:Advanced	Ability to view reports,create report…	
BI Web Services Users	Allows members to create and delet…	
Web Report Studio: Report Vie…	Provides report viewing capabilities.	
Web Report Studio: Report Cre…	Provides report creation capabilities.	
Web Report Studio: Advanced	Provides all capabilities in SAS Web …	
BI Dashboard Administrators	The members of this group are allow…	
BI Dashboard: Administration	Provides BI Dashboard administratio…	
BI Dashboard Users	The members of this group are allow…	
SAS Administrator		
SAS Trusted User		
SAS Anonymous Web User		
sas		

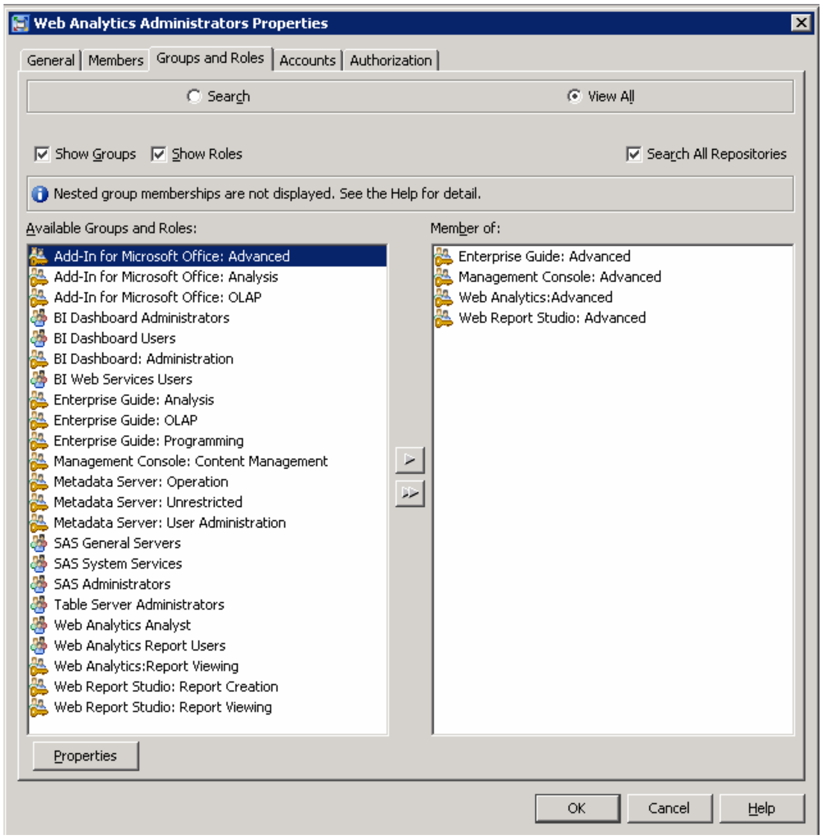
The following table lists the user groups provided for SAS Web Analytics and describes their uses.

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 and is a member of the **Web Analytics Analyst** group. 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.



Web Analytics Administrators Properties

○ 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:

- Add-In for Microsoft Office: Advanced
- Add-In for Microsoft Office: Analysis
- Add-In for Microsoft Office: OLAP
- BI Dashboard Administrators
- BI Dashboard Users
- BI Dashboard: Administration
- BI Web Services Users
- 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:

- Enterprise Guide: Advanced
- 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 ETL Administrators	Read, write, and administer access to all folders within this folder structure.
/Products/SAS Web Analytics/Data Sources	Web Analytics Analyst	Read-only access.
/Products/SAS Web Analytics/Data Sources	Web Analytics Report Viewers	Read-only access.
/Products/SAS Web Analytics/Reports	Web Analytics Analyst	Read and write access.
/Products/SAS Web Analytics/Reports	Web Analytics Report Users	Read-only access.
/Products/SAS Web Analytics/SEBD/Data Sources/SEBDMART	Web Analytics Analyst	Read-only access.
/Products/SAS Web Analytics/SEBD/Data Sources/SEBDMART	Web Analytics Report Users	Read-only access.
/System/Applications/SAS Web Analytics	Web Analytics Administrators	Read, write, and administer access to all folders within this folder structure.
/System/Applications/SAS Web Analytics/Web Analytics/Sites	Web Analytics Analyst	Read-only access.
/System/Applications/SAS Web Analytics/Web Analytics/Sites	Web Analytics Report Users	Read-only access.
/Shared Data	Web Analytics ETL Administrators	Read, write, and administer access to all folders within this folder structure.
/Shared Data/Web Analytics	Web Analytics Analyst	Read and write access.
/Shared Data/Web Analytics	Web Analytics Report Users	Read-only access.

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. (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. (See “Pre-Initialization Setup” on page 19.)

4 Initialize the site. (See “Site Initialization” on page 19.)

5 Copy the appropriate clickstream job to the log files for processing. (See “Post-Site Initialization Steps” on page 24.)

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: 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. (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.

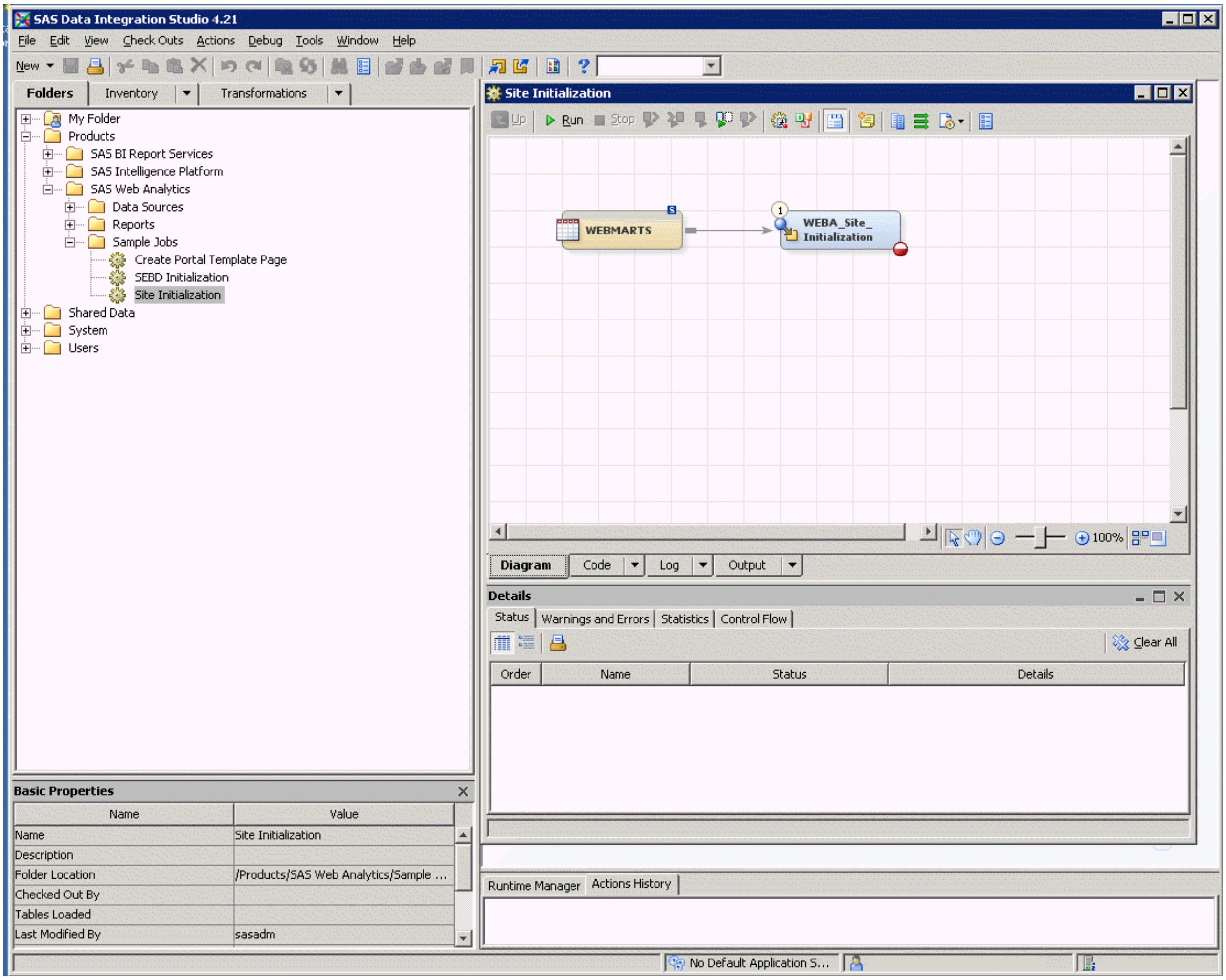
   Note: It is recommended that you follow best practices for setting up an Oracle server. Best practices include creating the DBUSER Group, which can be set up to contain logon information for Oracle databases.

   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*.

3. 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 22.

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

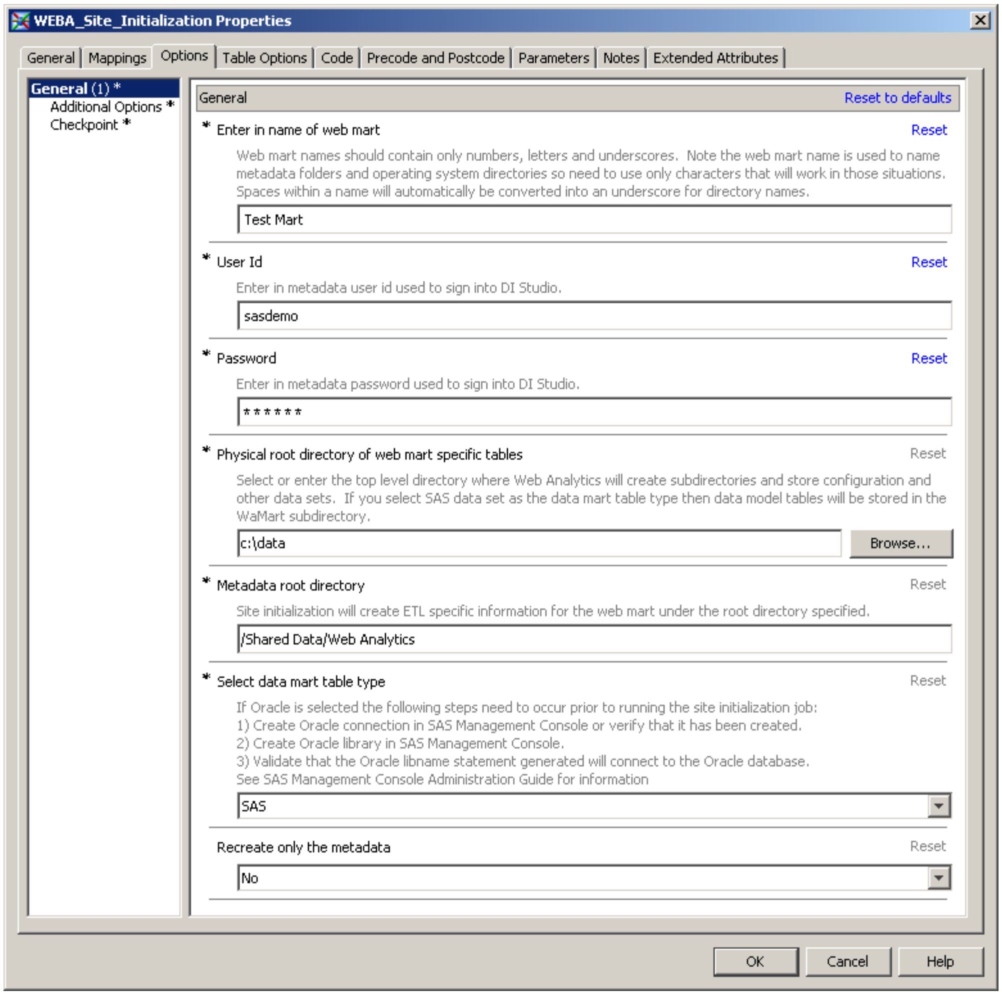
4 Double-click the **WEBA_Site_Initialization** transformation within the selected job, and then select the **Options** tab in the Properties dialog box.

5 Fill in the required information as described on the **Options** tab.

Note: The User ID should be a member of the Web Analytics group or have Read and Write access to the metadata folder **`/System/Applications/SAS Web Analytics/Web Analytics 5.3.3`** and the folder entered into the **Metadata root directory** field.

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.)

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



* Enter in name of web mart

Web mart names should contain only numbers, letters and underscores. Note the web mart name is used to name metadata folders and operating system directories so need to use only characters that will work in those situations. Spaces within a name will automatically be converted into an underscore for directory names.

Test Mart

* User Id

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

sasdemo

* 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 subdirectories and store configuration and other data sets. If you select SAS data set as the data mart table type then data model tables will be stored in the WaMart subdirectory.

c:\data

* Metadata root directory

Site initialization will create ETL specific information for the web mart under the root directory specified.

/Shared Data/Web Analytics

* Select data mart table type

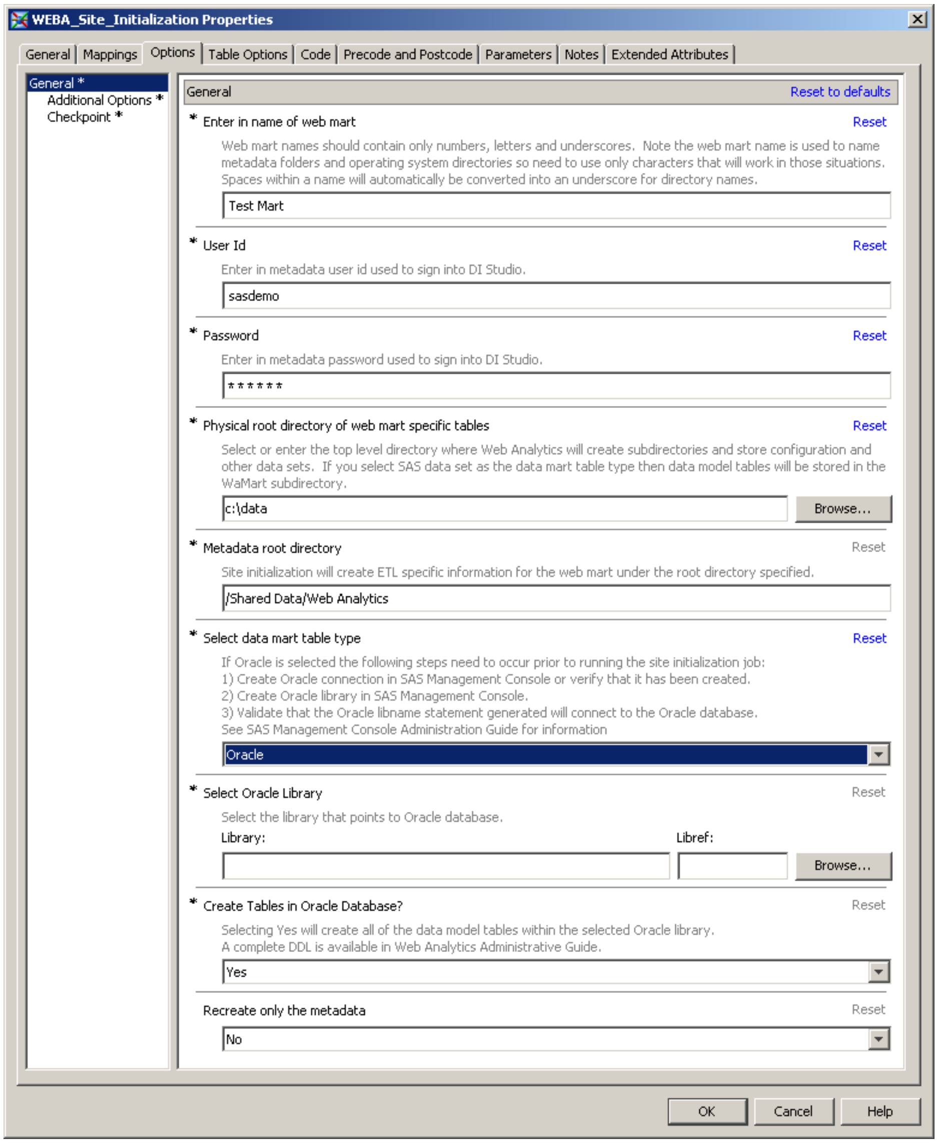
If Oracle is selected the following steps need to occur prior to running the site initialization job:
1) Create Oracle connection in SAS Management Console or verify that it has been created.
2) Create Oracle library in SAS Management Console.
3) Validate that the Oracle libname statement generated will connect to the Oracle database.
See SAS Management Console Administration Guide for information

SAS

Recreate only the metadata

No

The following is a sample **Options** tab for initializing a data mart that contains Oracle tables. If Oracle is selected, you will need to create an Oracle library object before running the Site Initialization job.



General

* Enter in name of web mart

Web mart names should contain only numbers, letters and underscores. Note the web mart name is used to name metadata folders and operating system directories so need to use only characters that will work in those situations. Spaces within a name will automatically be converted into an underscore for directory names.

Test Mart

* User Id

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

sasdemo

* 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 subdirectories and store configuration and other data sets. If you select SAS data set as the data mart table type then data model tables will be stored in the WaMart subdirectory.

c:\data

* Metadata root directory

Site initialization will create ETL specific information for the web mart under the root directory specified.

/Shared Data/Web Analytics

* Select data mart table type

If Oracle is selected the following steps need to occur prior to running the site initialization job:
1) Create Oracle connection in SAS Management Console or verify that it has been created.
2) Create Oracle library in SAS Management Console.
3) Validate that the Oracle libname statement generated will connect to the Oracle database.
See SAS Management Console Administration Guide for information

Oracle

* Select Oracle Library

Select the library that points to Oracle database.

Library: Libref:

* Create Tables in Oracle Database?

Selecting Yes will create all of the data model tables within the selected Oracle library.
A complete DDL is available in Web Analytics Administrative Guide.

Yes

Recreate only the metadata

No

Click **OK**.

6 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 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.

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:

  `5.3.3 Jobs` – all ETL jobs reside in this folder

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

`EvntScr`
`WaAnly`
`WaCnfg`
`WaMart`
`WaSrc`
`WaWork`

- ❑ Operating System directories. The following directories were created under the operating system root directory:

`EvntSrc`
`WaAnly`
`WaCnfg`
`WaMart`

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

`WaSrc`
`WaWork`

- ❑ 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 completely disables the Web mart.

Post-Site Initialization Steps

- ❑ 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.
- ❑ Verify that the correct directories were 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.

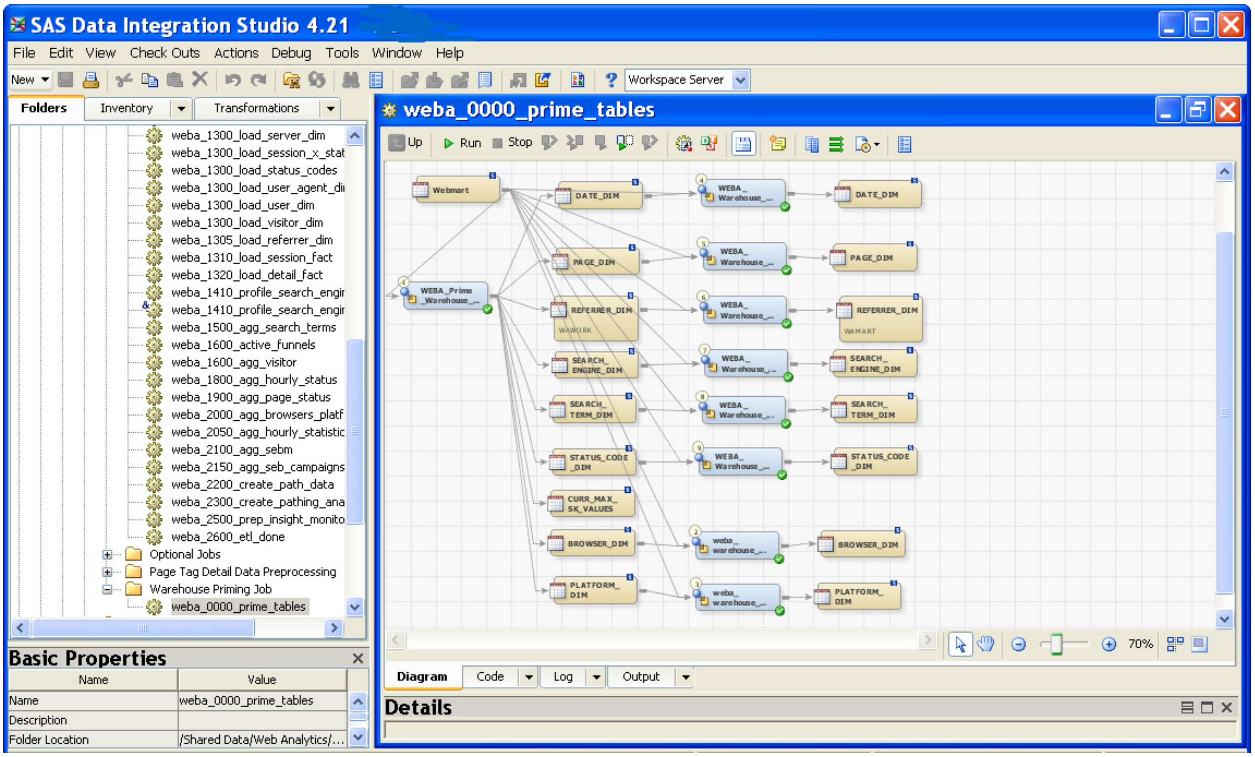
❑ After the data mart tables have been created and the indexes have been applied, you are ready to pre-fill the dimensional tables.

Navigate to **Shared Data/Web Analytics**/<*Web mart name*>/**5.3.3 Jobs/Warehouse Priming Job**. Open the **weba_0000_prime_tables** job in Data Integration Studio and run it. If the job is successful, then several dimensional tables will be pre-filled with records. Your site is now ready to deploy and schedule the ETL.

❑ 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 completely disables the Web mart.


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_engir
weba_1410_profile_search_engir
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/...

Your site is now ready to deploy and schedule the ETL. For information about setting up the ETL, see Chapter 3, “Setting Up and Working with ETL Tasks.”

Troubleshooting

Setting Up Users

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.

CHAPTER

3

Setting Up and Working with ETL Tasks

Setting Up and Working with Web Data ETL Tasks

Overview of Setup and ETL Tasks

- Verify that site initialization has been run. (See Chapter 2, “Setting Up a Web Mart.”)
- Add and set up the Clickstream job or jobs as described in the *SAS Data Surveyor for Clickstream Data User’s Guide*.
- Verify that the Clickstream job will result in at least one data set with the prefix **weblog_detail** in the **WaSrc** directory**.**
- Deploy and schedule ETL jobs in the <*Web mart*> **\5.3.3 Jobs\ETL Jobs** folder using the Platform Load Sharing Facility (LSF) or another scheduler.

Creating the WEBMART Data Set

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;
```

SAS Web Analytics ETL Jobs

Required ETL Jobs

ETL jobs can be divided into these groups:

- reading Web or tagging logs and grouping a single visitor’s data: Clickstream jobs. For more information, see the SAS Data Surveyor for Clickstream Data documentation.
- preprocessing job 0050 (required if you are using page tag data. For more information, see “Page Tag Preprocessing” on page 33)
- loading SAS Web Analytics data mart tables: jobs 1100–1320
- aggregating update: jobs 1500–2150

- ❑ preparing analytic report data: jobs 1410, 2200, 2300, and 2500
- ❑ cleaning up job 2600

The following table lists all of the jobs that make up the ETL schedule for Web Analytics. If a job is optional, then the ETL administrator can decide whether to run the job or not. If a job is customizable, then the ETL administrator can update the job to meet the needs of an individual Web site. Jobs with an asterisk (*) need performance tuning depending on the size of the Web or tagging logs. For information about performance tuning, see "WEBA ETL Jobs and Performance Tuning" on page 33.

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

Table 3.1: WEBA ETL Jobs

Job Name	**Dependency**	**Description**	**Optional?**	**Customizable?**
Click stream job or jobs	Web log or tagging log creation	Reads in and groups Web log or tagging log data by visitor.	No	Yes
weba_0050_page_tag_log_detail_ preprocessing	Clickstream Tagging Log job	Job splits the output data set of the Clickstream Tagging Log Job into two data sets. Data set WEBLOG_DETAIL *_xxxx* will contain records where the requested file was loaded. Data set EVENT_DETAIL *_xxx* will contain records where an event occurred.	Required if using page tag data	Yes

Job Name	Dependency	Description	Optional?	Customizable?
weba_1100_data_health_check	Clickstream job or jobs that create the weblog_detail_1 data set	Checks Web log detail data for domains with missing values and ensures that the current input files have been properly loaded. It also reads the input files and performs some mandatory and optional checks.	No	Yes
weba_1200_warehouse_staging_tables	weba_1100_data_health_check	Reads the detail data set created using Clickstream jobs and creates all of the necessary staging tables.	No	No
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_ip_address_dim*	weba_1200_warehouse_staging_tables	Loads the IP_ADDRESS_DIM table with new IP addresses.	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

Job Name	Dependency	Description	Optional?	Customizable?
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
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_1320_load_detail_fact*	weba_1200_warehouse_staging_tables	Updates the DETAIL_FACT table with new session detail data.	No	No

Job Name	Dependency	Description	Optional?	Customizable?
weba_1410_profile_search_engines	weba_1320_load_detail_fact weba_1310_load_session_fact	This is an example profile job – See "(Optional) Profile Jobs " on page 36.	Yes	Yes
weba_1500_agg_search_terms	weba_1310_load_session_fact	Updates the Aggr_Search_Term_&interval summary data sets.	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 data sets.	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 data sets.	No	Yes
weba_2000_agg_browsers_platforms	weba_1900_agg_page_status	Updates the Aggr_Platform_&interval data sets.	No	Yes
weba_2050_agg_hourly_statistics	weba_2000_agg_browsers_platform	Updates the Aggr_Daily_Total_&interval data sets.	No	Yes
weba_2100_agg_sebm	weba_2000_agg_browsers_platforms	Updates the Aggr_sebm_&interval data sets.	No	Yes
weba_2150_agg_seb_campaigns	weba_2100_agg_sebm	Updates the Aggr_seb_Campaign_&interval data sets.	No	Yes
weba_2200_create_path_data	weba_2150_create_path_data	Updates the path analytical data sets.	No	Yes
weba_2300_create_pathing_analytics	weba_2200_create_path_data	Creates path analytical data sets.	No	No
weba_2500_prep_insight_monitor	weba_2300_createpathing_analytics_data	Creates waanly.anly_daily_metrics, plus weekly and monthly versions.	No	Yes

Job Name	Dependency	Description	Optional?	Customizable?
weba_2550_ma_response_history_update	weba_1200_warehouse_staging_tables	Updates SAS Marketing Automation response history.	Yes	Yes
weba_2600_etl_done	weba_2500_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 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, then do not run this job.	Yes	Yes

Page Tag Preprocessing

Introduction to the Clickstream Page Tag Job

The Clickstream Page Tag job creates a detail data set (named Tagged_DDS by default) that contains multiple record types. The record type is identified by the Event variable, which can contain the values CLICK, SUBMIT, LOAD, or Blank.

To prepare the Tagged_DDS data set for processing within the SAS Web Analytics ETL Tagged_DDS, you need to remove the Event variable records that contain the values CLICK and SUBMIT from the detail data set. You can remove the records by using one of these methods:

- ❑ Modify the Clickstream Page Tag job to keep records only when the Event variable value is LOAD or Blank.
- ❑ Run the weba_0050_page_tag_log_detail_preprocessing job before running the weba_1100_data_health_check job.

Preparing to Run the weba_0050_page_tag_log_detail_preprocessing Job

Prepare to run the weba_0050_page_tag_log_detail_preprocessing job by completing these steps:

1 Implement the Clickstream page tag on all (preferable) taggable pages within your Web site. For more information, see the *SAS Data Surveyor for Clickstream Data 2.1: User's Guide*.

2 Set up the Clickstream collection server and schedule saving the log to a file. For more information, see the *SAS Data Surveyor for Clickstream Data 2.1: User's Guide*.

3 In SAS Data Integration Studio, run the Site Initialization job. For more information, see "Site Initialization" in Chapter 2.

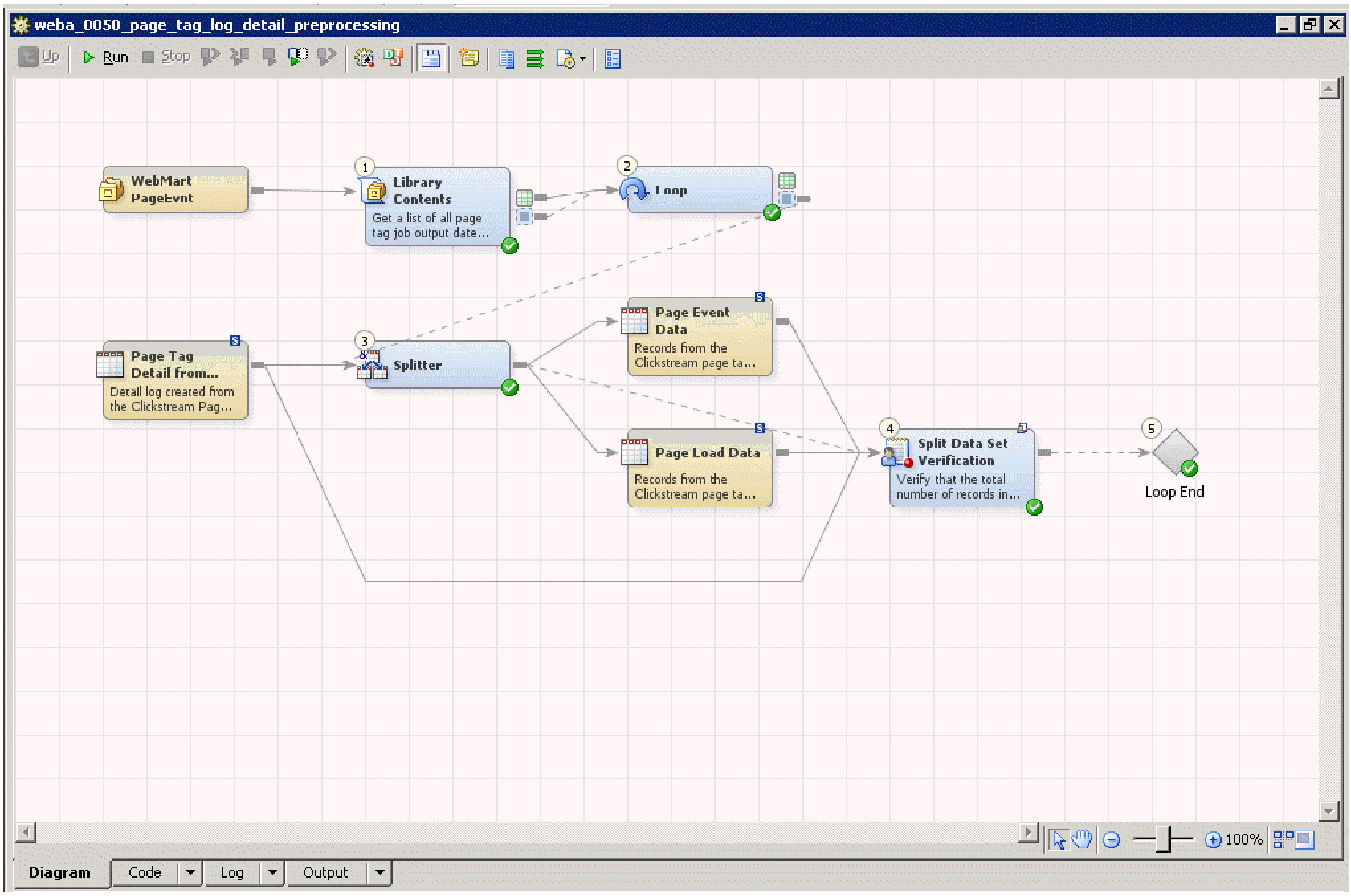
4 Run the Warehouse Priming job. For more information, see "Post-Site Initialization" in Chapter 2.

5 Follow the Clickstream instructions for selecting the Clickstream Page Tag job and running the setup. It is recommended that you add the Clickstream Page Tag job to the `5.3.3 Jobs` folder within the Web site folder.

6 Modify the weba_0500_page_tag_log_detail_preprocessing job if necessary.

Setting Up the weba_0050_page_tag_log_detail_preprocessing Setup Job

The preprocessing job resides in the **`Page Tag Detail Data Preprocessing`** folder under **`5.3.3 Jobs`**.


weba_0050_page_tag_log_detail_preprocessing
Up
Run
Stop
WebMart PageEvnt
1
Library Contents
Get a list of all page tag job output date...
2
Loop
Page Tag Detail from...
Detail log created from the Clickstream Pag...
3
Splitter
Page Event Data
Records from the Clickstream page ta...
Page Load Data
Records from the Clickstream page ta...
4
Split Data Set Verification
Verify that the total number of records in...
5
Loop End
100%
Diagram
Code
Log
Output

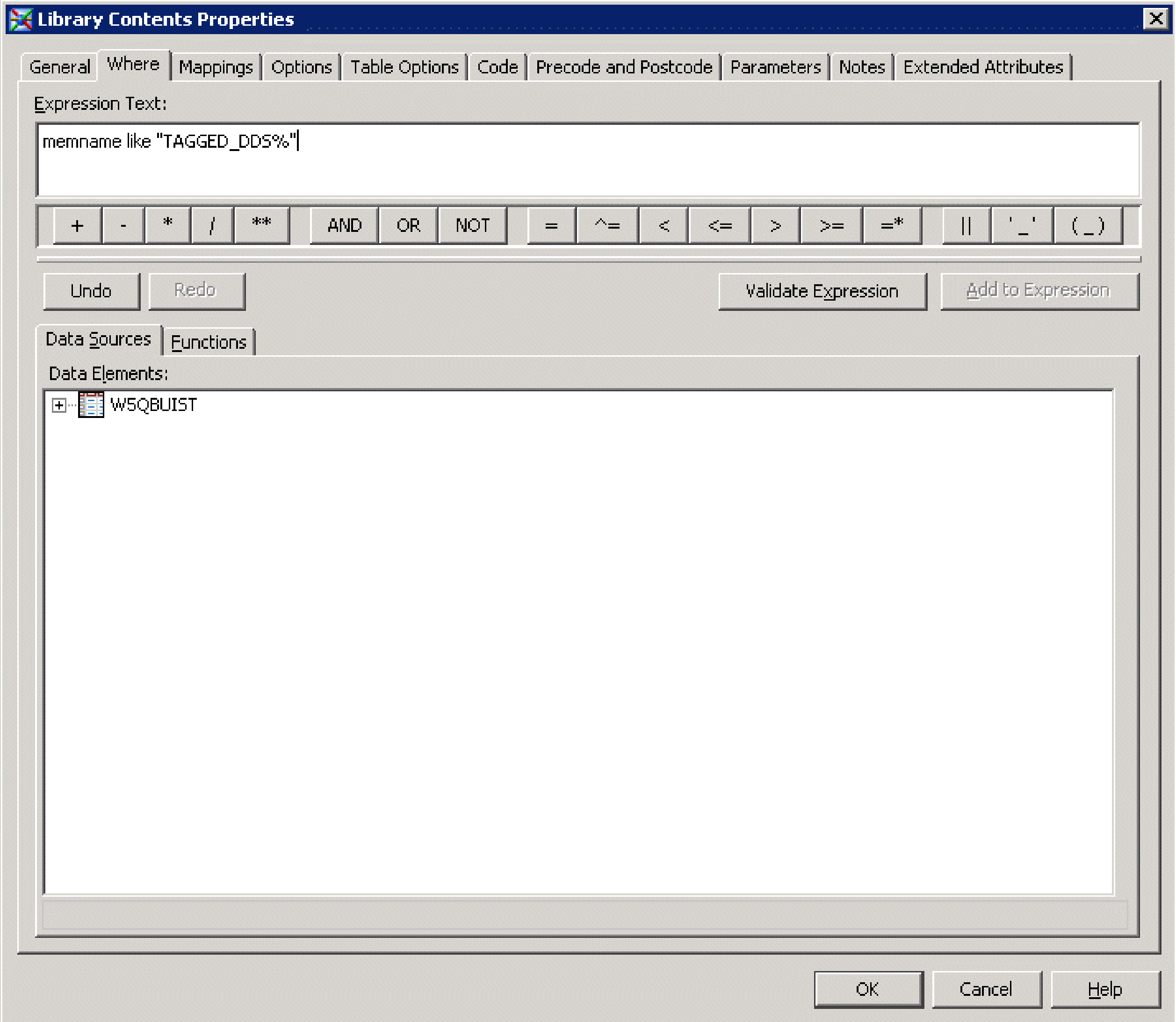
Some modifications to the job are required:

- The weba_0050_page_tag_log_detail_preprocessing job requires that the output from the Clickstream Page Tag job be placed into the EVNTSRC directory within the Web mart directory structure. You need to perform the following tasks to either accommodate that requirement or change the EVNTSRC directory:

  - Create a job that moves the output data set from the Clickstream Page Tag job to EVNTSRC.
  - Verify that the library of the Page Tag Detail from the Clickstream table object is EVNTSRC.

- The first step in the preprocessing job uses the data set name TAGGED_DDS to determine the data sets to be processed by the job. If the data set name is changed within the Clickstream job, then you need to change the Library Contents node within the preprocessing job accordingly.

  Note: The name of the Clickstream job output data set also needs to be changed. By default, the data set name of the final output data set within the Clickstream Page Tag job is TAGGED_DDS.



Expression Text:

```
memname like "TAGGED_DDS%"
```

WEBA ETL Jobs and Performance Tuning

There are three tuning options for each table:

- ❑ bulk-load tuning.
- ❑ regular tuning.
- ❑ data-driven tuning, which uses different load methods based on the size of the incoming data. This option does not take the size of the target table into account.

There is also an add-and-drop index functionality, which becomes more critical during a warehouse's lifetime.

Oracle Load Options

There is also an UPSERT switch, which is an Oracle option for updating and inserting records simultaneously during a load.

Drop and Re-create Indexes

The drop and re-create indexes option varies from application to application. Customers can add their own indexes and will likely use the option at a table level.

(Optional) Profile Jobs

You also have the option of adding profile jobs. Each job is responsible for creating a separate profile.

Profiling flags sessions so the subsets 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 to use this profile in an interactive request, then the analysis results that are viewed will automatically be limited to sessions that match the category—that is, the sessions that were sent to the site by Google.

To maintain profile jobs, administrators are responsible for obtaining a list of sessions in a category and running a SAS Data Integration Studio transformation that uses the list to flag the sessions in the warehouse. The administrator schedules the profile to execute on a regular basis, and as a result, sessions are flagged as they are appended to the warehouse tables.

Two types of profiles, visitor and event, can be used independently or combined. A visitor profile describes information about a Web site's user, whereas an event profile describes characteristics about when and how 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. In other words, a Males profile can define all male users, and a separate Young Adult profile can define all users between 20 and 30 years of age. Users can then choose to select data when it matches the Males profile or the Young Adult profile. Users can also select data that matches one or the other (all males or 20- to 30-year-olds) and both (all 20- to 30-year-old males.)

How to Define a Profile

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.

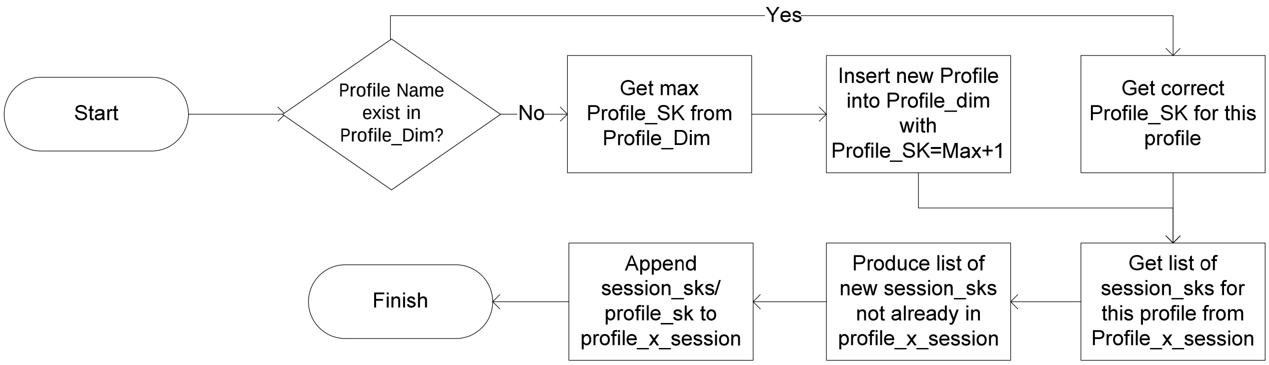
A subquery is also necessary to identify sessions that accessed page_desc Z. The user also needs to inner join the matches between the preceding query and the subquery, which pulls page_desc_sks from page_desc_dim, where page_desc=’Z’, and matches the results with page_desc_sks from the DETAIL_FACT table, keeping 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 the requisite records on input data and reconsuming such 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 consuming WaWork or WaMart data exclusively for this purpose. If users choose to consume WaSrc input data directly for this job, they will bypass the business rules that are applied during ETL, which would circumvent the purpose of their queries.

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 3.2: Profiling Options

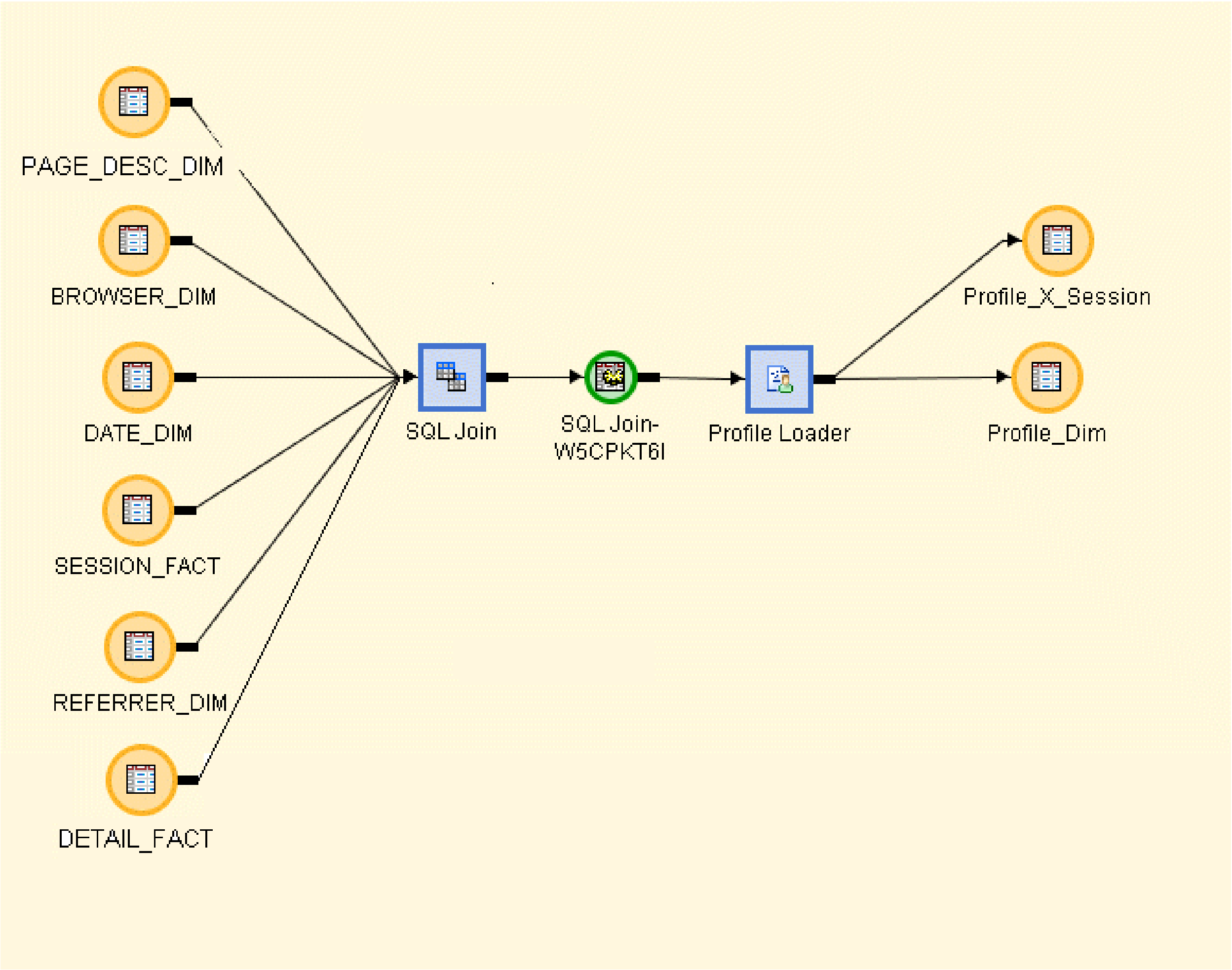
Option	Valid Values	Description
Profile Name	Any 40 characters	The name of the profile as it is stored in the PROFILE_DIM table. If the name changes, then it is treated as a new profile. Each profile must have a unique name.
Profile Description	Any 100-character value	The description that is stored in 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 3.1: Profile Task Flow


Start
Profile Name exist in Profile_Dim?
Yes
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 3.2.

Figure 3.2: SAS Data Integration Studio Profiling Job



Modifying Profiles

Profiles can be modified and redeployed for scheduling, but the administrator needs to decide 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 the administrator 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 requisite business rule changes and set "Drop old records" to YES and run the job once. Upon completion, the warehouse is updated to reflect new business rules on past history. The administrator now has a choice of changing the "Drop old records" option to NO. In any event, the job needs to be redeployed to pick up changes for future runs.

Deleting Profiles

While simply removing a profiling job from the scheduled flow effectively halts updates to the profiling tables, it does not remove past history. The administrator has to decide whether the profile needs to be unscheduled if the profile simply no longer needs to be updated, or whether the profile's data should be removed altogether. The stepwise process described in this section is recommended for proper dismantling of a profiling job and removal of its data from the warehouse.

To remove a job from the warehouse ETL schedule and subsequently 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 ETL (for example, WEBA_1410_PROFILE_SEARCH_ENGINES). 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.

If you want a profile to identify sessions that have been driven to a site by the Google search engine, then you first need to identify the event that triggers that the condition has been satisfied. This event is 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 (that is, its search_engine_desc field contains 'Google'), then the profile's condition has been met and 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.

Customizing the ETL Jobs

Best Practices

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 needed 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* 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, 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.

Customization

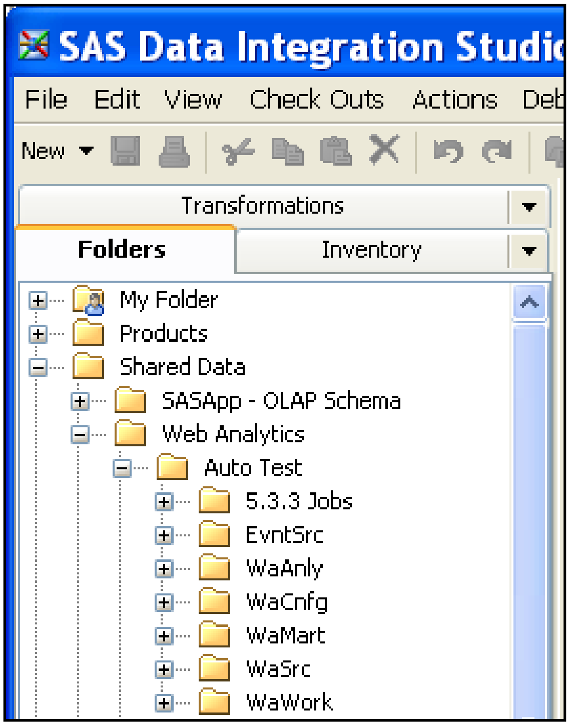
Here are some guidelines for customizing a site's ETL job after the site has been initialized:

Assigning a Libref

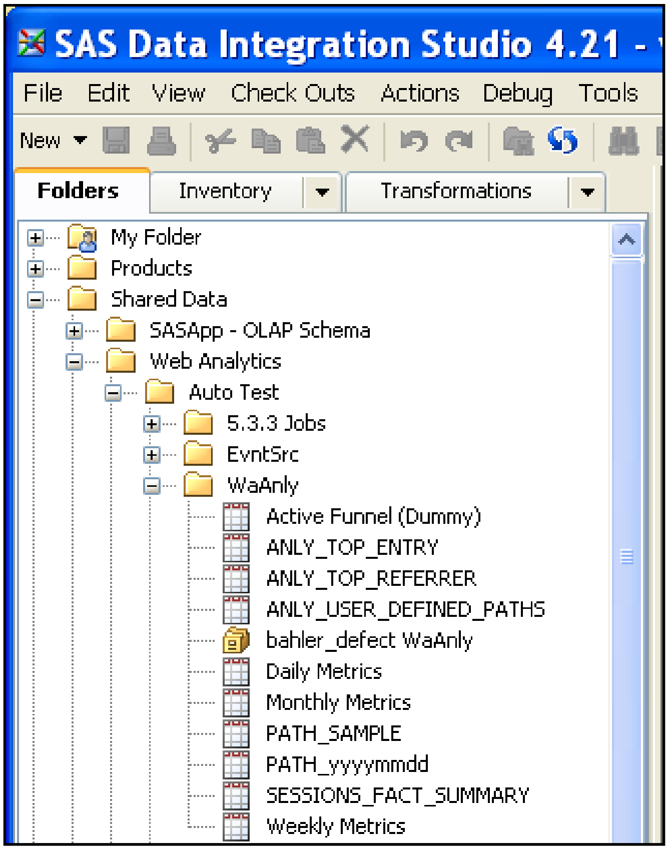
By default, the site initialization job assigns a site's library locations. The default library locations are a set of subdirectories under the root operating system directory specified within the job.

To change a library location for a site:

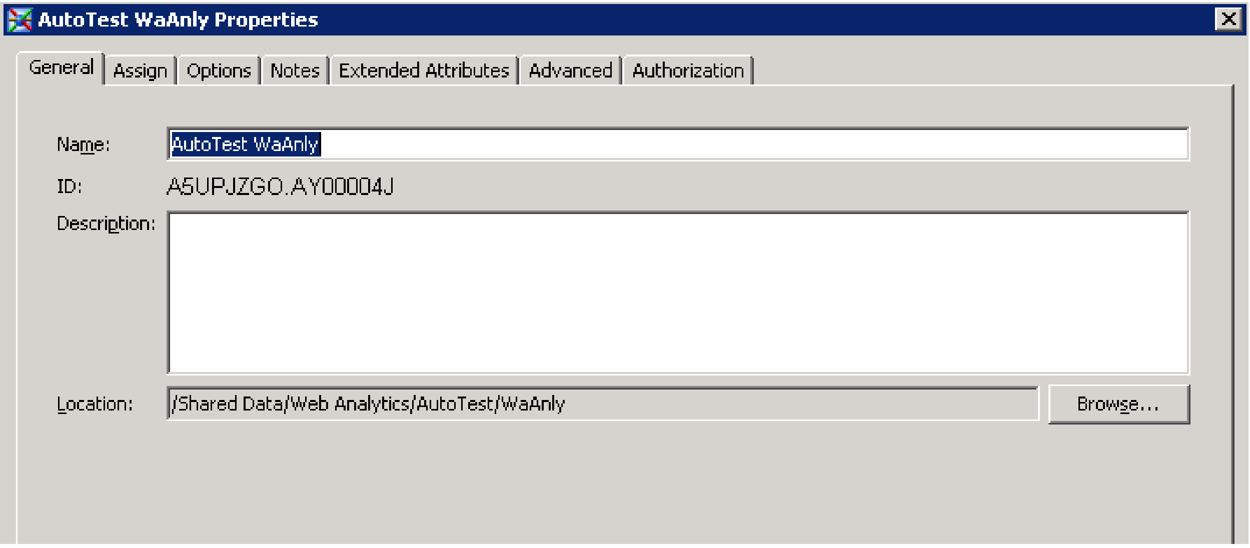
3. Open SAS Data Integration Studio.
4. Navigate to the metadata folder for a site.
5. Expand the folder. Six subfolders appear. The subfolders that contain the library objects are **WaAnly**, **WaCnfg**, **WaMart**, **WaSrc**, and **WaWork**.



6 Expand the folder that contains a library object.



7 Double-click the library object to open the Properties window.



Name: AutoTest WaAnly

ID: A5UPJZGO.AY00004J

Description:

Location: /Shared Data/Web Analytics/AutoTest/WaAnly

8 Update the library location. SAS library – click the **Options** tab and update the directory information. For more information about updating SAS library options, see the SAS Data Integration Studio Help.

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

WEBA ETL Jobs That Might Require Performance Tuning

Performance Tuning Options

There are three types of tuning options for each table:

- ❑ bulk-load tuning option.
- ❑ regular tuning option.
- ❑ data-driven tuning option – uses different load methods based on the size of the incoming data. It does not take the size of the target table into account.

Oracle Load Options

There is also an upsert option, which is an Oracle option for updating and inserting simultaneously during a load.

Drop and Re-create Indexes

The drop and re-create indexes option varies from application to application. Sites can add their own indexes, particularly at a table level.

Jobs

The following table shows the jobs that might need to have one or more performance tuning options altered.

Table 3.3: ETL Jobs That Might Need Performance Tuning Options Altered

Job Name	Associated Job	Dependency	Description
weba_1100_data_health_check	Not applicable	Clickstream jobs completed successfully	Checks Web log detail data for domains with missing values and ensures that the current input files do not have the same number of records as a previously processed input file. It also reads the input files and performs some mandatory and optional checks.
weba_1300_load_ip_address_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the IP_ADDRESS_DIM table with new IP addresses.
weba_1300_load_page_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the PAGE_DIM table with new pages.
weba_1300_load_platform_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the PLATFORM_DIM table with new platforms.
weba_1300_load_referrer_query_string_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the REFERRER_QUERY_STRING_DIM table with new query strings.
weba_1300_load_search_term_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the SEARCH_TERM_DIM table with new search terms.
weba_1300_load_session_x_status_code	Not applicable	weba_1200_warehouse_staging_tables	Loads the SESSION_X_STATUS_CODE table with new session-status code combinations.
weba_1300_load_user_agent_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the USER_AGENT_DIM table with new IP addresses.
weba_1300_load_user_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the USER_DIM table with new user IDs.
weba_1300_load_visitor_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the IP_ADDRESS_DIM table with new visitor IDs. The visitor ID is assigned within the clickstream jobs.
weba_1305_load_referrer_dim	Not applicable	weba_1200_warehouse_staging_tables	Loads the REFERRER_DIM table with new referrer URLs.
weba_1310_load_session_fact	Not applicable	weba_1200_warehouse_staging_tables	Updates the SESSION_FACT table with new session data.
weba_1320_load_detail_fact	Not applicable	weba_1200_warehouse_staging_tables	Updates the DETAIL_FACT table with new session detail data.

The following table shows the jobs that can be customized on a per-site basis.

Table 3.4: WEBA ETL Customizable Jobs

Job Name	Associated SAS Data Integration Studio Job	Dependency	Description	Optional?
weba_1410_profile_ search_engines	Parameterized_WEBA_1410_ PROFILE_SEARCH_ ENGINES	weba_1320_load_detail_ Fact	Updates Profile_X_Session for search engine, Google, and Yahoo profiles.	No
weba_1500_agg_ search_terms	Not applicable	weba_1400_profile_search_ Engines	Updates Aggr_Referrer_Search_ Term_&interval.	No
weba_1600_active_ funnels	Not applicable	weba_1500_agg_search_ terms	Updates the active funnels analysis data sets - anly.d#.	No
weba_1600_agg_ visitor	Not applicable	weba_1600_active_funnels	Updates the Aggr_Visitor_&interval data sets.	No
weba_1800_agg_ hourly_status	Not applicable	weba_1700_agg_visitor	Updates the AGGR_Hourly_Status_&in terval data sets.	No
weba_1900_agg_ page_status	Not applicable	weba_1800_agg_hourly_ status	Updates the AGGR_PAGE_&interval data sets.	No
weba_2000_agg_ browsers_platforms	Not applicable	weba_1900_agg_page_ status	Updates the Aggr_Platform_&interval data sets.	No
weba_2050_agg_ hourly_statistics	Not applicable	weba_2000_agg_browsers_ platform	Updates the Aggr_daily_total_&interva l data sets.	No
weba_2100_agg_ sebm	Not applicable	weba_2000_agg_browsers_ platforms	Updates the AGGR_SEBM_&interval data sets.	No
weba_2150_agg_ seb_campaigns	Not applicable	weba_2100_agg_sebm	Updates the AGGR_SEB_CAMPAIGN_ &interval data sets.	No
weba_2200_create_ path_data	Not applicable	weba_2150_create_path_ data	Updates clickstream analysis data sets.	No
weba_2300_create_ pathing_analytics	Not applicable	weba_2200_create_path_ data	Creates clickstream analysis data sets.	No
weba_2500_prep_ insight_monitor	Not applicable	weba_2300_top_entry_ referrer	Creates waanly.anly_daily_metrics , plus weekly and monthly versions.	No
weba_2550_ma_ response_history_ update	Not applicable	weba_1200_warehouse_ staging_tables	Sends response history updates to the SAS Customer Intelligence interface (SAS Marketing Automation).	Yes

Job Name	Associated SAS Data Integration Studio Job	Dependency	Description	Optional?
weba_2600_etl_done	Not applicable	weba_2500_prep_insight_ Monitor	Removes input data and allows another ETL job to begin executing (optional).	No
weba_9999_ maxmind_load	Not applicable	Not applicable	Loads MaxMind geographic data.	Yes

Customizing the Job to Create Path Data

The weba_2200_create_path_data job has parameters 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 pathing 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 pathing 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.

❑ **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.

To change the values of the parameters:

1 Open SAS Data Integration Studio and navigate to <*Web mart*>/`5.3.3 Jobs/ETL Jobs`.

2 Click the **weba_2200_create_path_data** job.

3 Click the **Stat Pathing** transformation, and then open **Properties**.

4 Click the **Options** tab and update the value of the desired parameter.

5 Save your changes.

6 Redeploy the job.

Extending the Data Warehouse

Overview

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

Warehouse tables contain a predefined set of fields. To extend the data warehouse 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 making the necessary changes to create an extension table:

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 field.

3 Determine which data warehouse fact extension table (DETAIL_FACT_EXT or SESSION_FACT_EXT) or dimension table (*dimension*_DIM) the field should be added to.

4 If necessary, modify the clickstream jobs to capture the new fields, or create a new job that will update the Weblog_Detail_1 data set to contain this new field before the weba_1200_warehouse_staging_tables job is run.

Steps for Creating a *fact*_EXT Table

The *fact*_EXT table contains attributes for a new metrics in addition to the *fact*_SK attribute. A new metric is created using business rules that are specific for an individual Web site.

Complete the following steps before you create or update a *fact*_EXT table:

Step 1 - Update the Metadata Table

Update the metadata for the *fact*_EXT table in the *Web mart*/WAMART folder. The table must belong to the WAMART library within that folder.

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

2 Right-click the appropriate *fact*_EXT table and select **Properties.** Add the new columns.

Step 2 - Create the Physical Table

If you are using SAS as the data repository, then open an interactive SAS session or SAS Enterprise Guide. Use the DATA step to create the DETAIL_FACT_EXT or SESSION_FACT_EXT table before initially populating the table through a load job.

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

Step 3 - Load the Table

Modify one of the sample jobs for loading an extension table. The sample jobs are found within the Optional Jobs folder under the *Web mart* 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 by the session_sk.

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.

Adding More Fields to a Warehouse Table

If you are using SAS data sets as the data repository, then adding fields to an extension table will not be a problem. If you are using Oracle, then check with your database administrator for best practices for adding columns to a table.

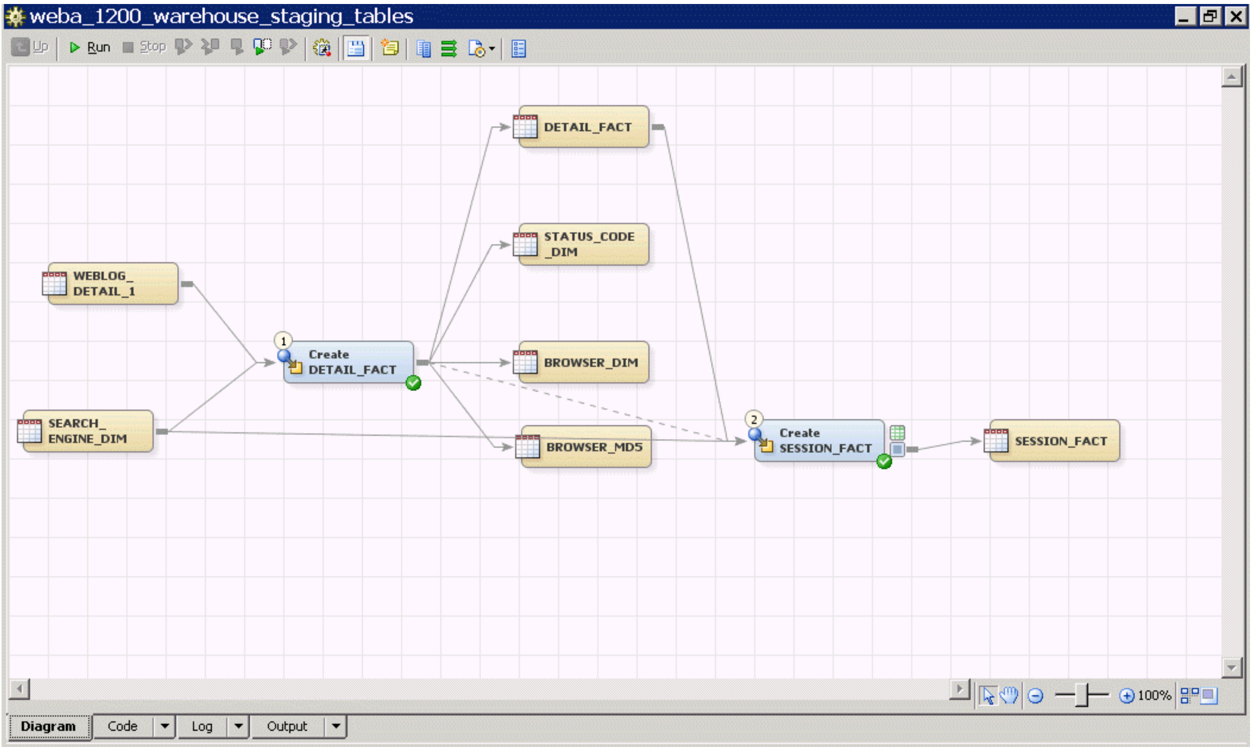
Adding *fact* Table Columns

Steps for Creating a DETAIL_FACT_EXT Table

Step 1 – Modify the weba_1200_warehouse_staging_tables Job

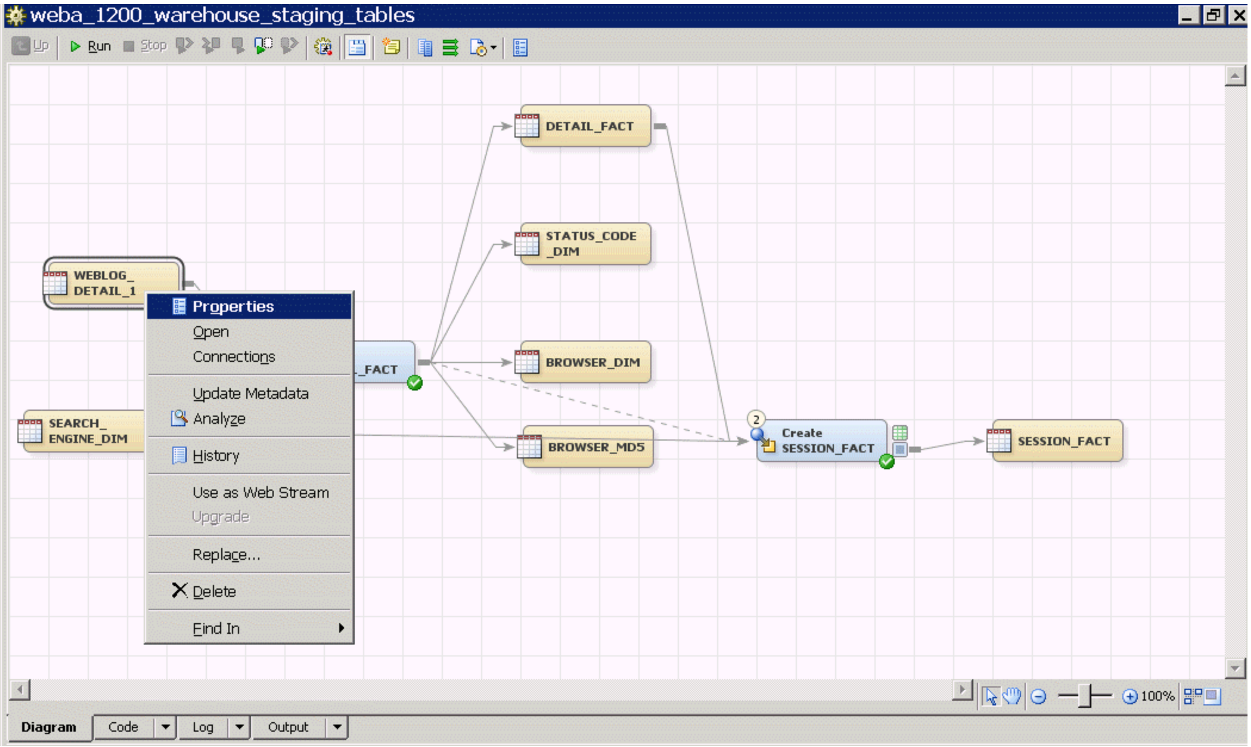
To add variables to the WAWORK.DETAIL_FACT table:

1 From the <*Web mart*>`/5.3.3 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.

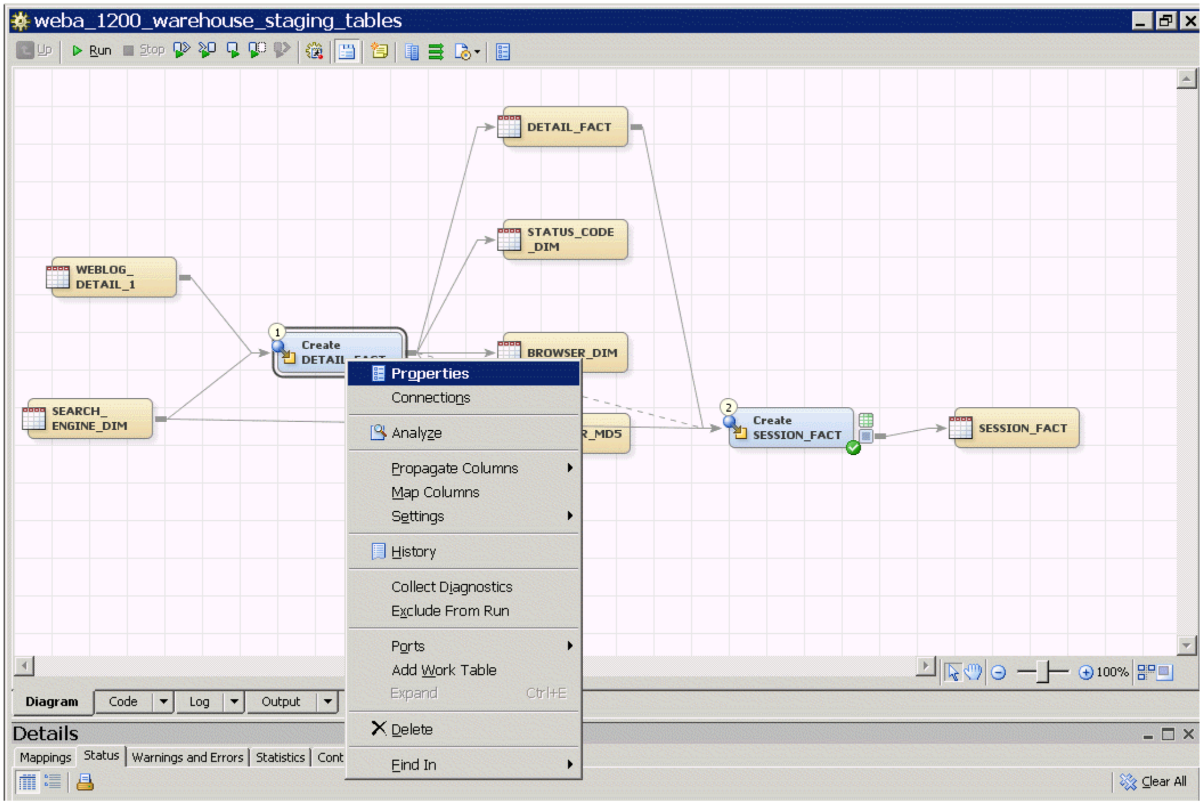


2 Add a preprocessing step for the WEBLOG_DETAIL_1 table, if necessary.

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.



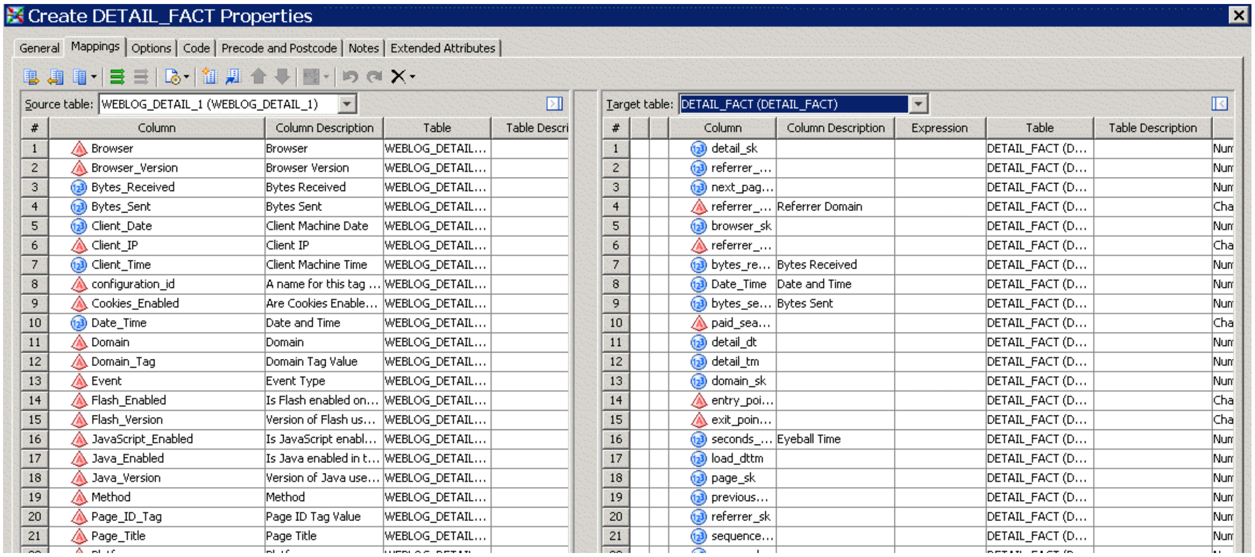
4 Double-click **Create Detail Fact** to open the Properties window.



5 Click the **Mappings** tab.

a. Select **Source Table - Weblog_Detail_1** and **Select Target Table - Detail_Fact**.

Note: Columns are not mapped.



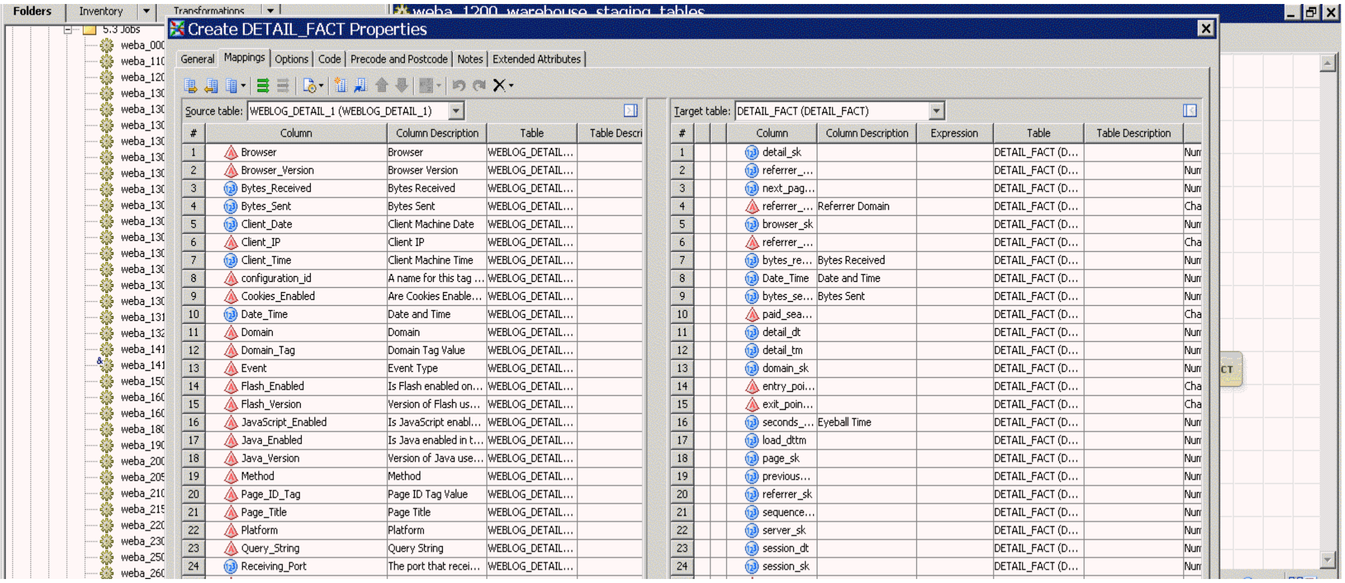
Source table: WEBLOG_DETAIL_1 (WEBLOG_DETAIL_1)

#	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

b. On the left side, right-click **Column** to add to the DETAIL_FACT table. Select **Propagate --> Select Source Columns --> To Targets.**



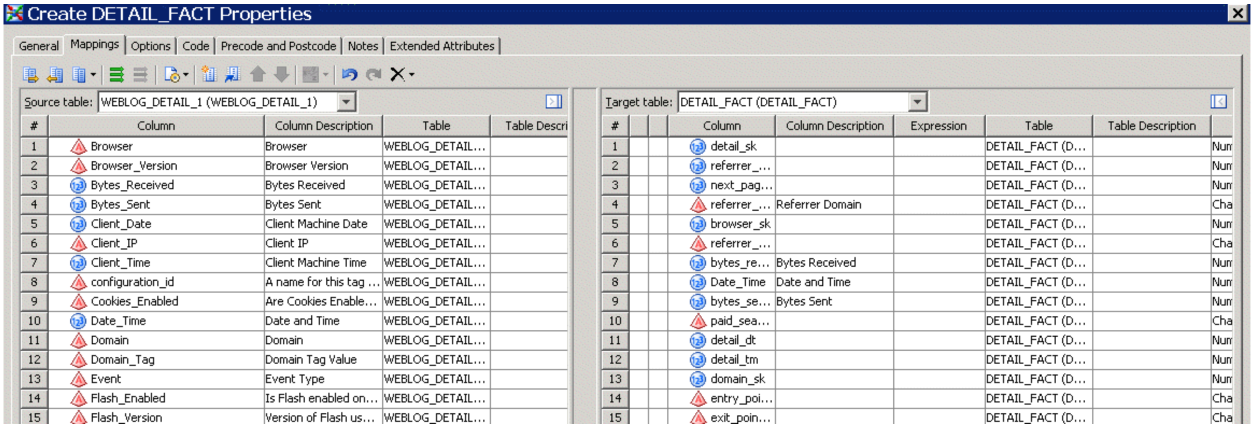
Source table: WEBLOG_DETAIL_1 (WEBLOG_DETAIL_1)

#	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...	
22	Platform	Platform	WEBLOG_DETAIL...	
23	Query_String	Query String	WEBLOG_DETAIL...	
24	Receiving_Port	The port that recei...	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
22	server_sk			DETAIL_FACT (D...		Num
23	session_dt			DETAIL_FACT (D...		Num
24	session_sk			DETAIL_FACT (D...		Num

c. 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).



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...	

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

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

7 Save the job.

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

Step 2 - Create the DETAIL_FACT_EXT table

Step 3 - Load the DETAIL_FACT_EXT table

The table loading *must* occur after the weba_1320_load_detail_fact job has been run. The following steps are required to load the DETAIL_FACT_EXT table:

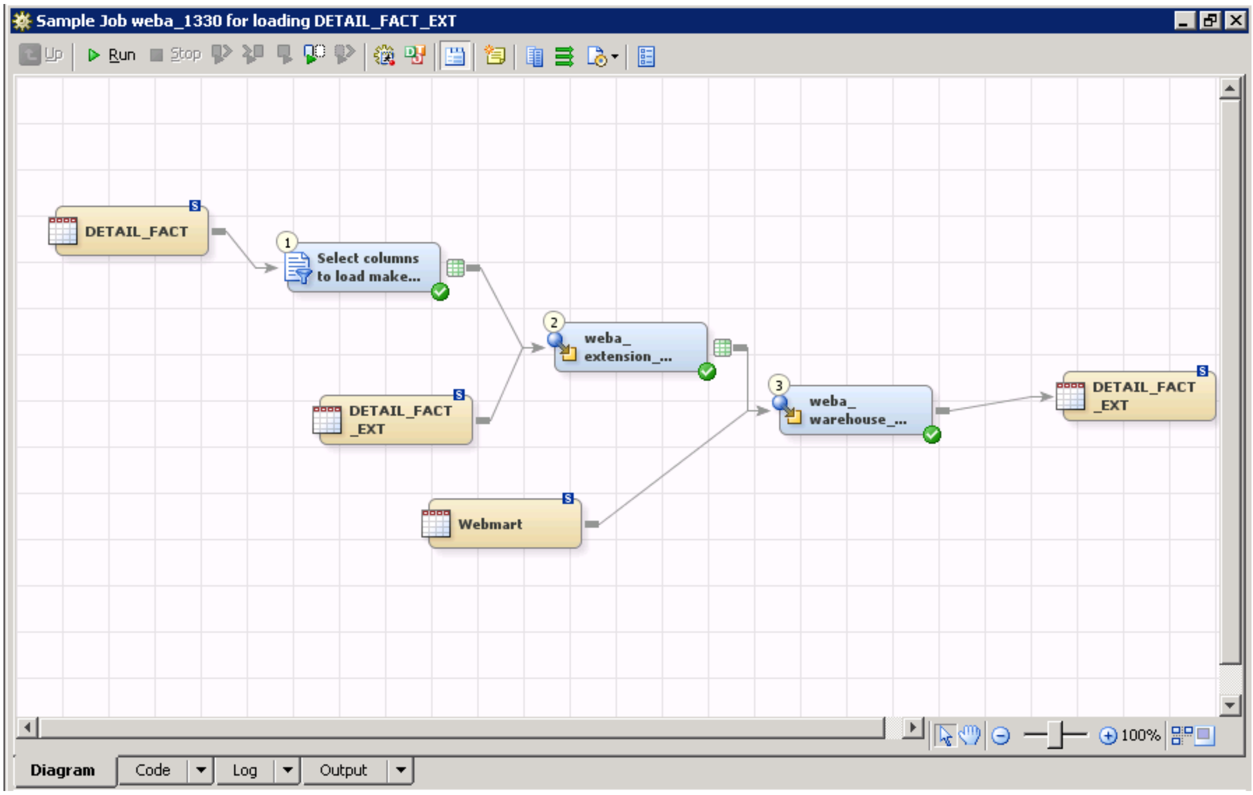
❑ Under the <*Web mart*>**/5.3.3 Jobs/Optional Jobs** folder, select **Sample Job weba_1330 for Loading Detail_Fact_Ext**. This job is a template for creating a job that will load the DETAIL_FACT_EXT table.

Recommendations:

- Make a copy of this template job with a new name that indicates where in the ETL the job should run.
- Move the copy under the **`5.3.3 Jobs/ETL Jobs`** folder.
- This job should run within the ETL after the weba_1320_load_detail_fact job is run.

❑ In the SQL Join transformation, select the columns from the WAWORK.DETAIL_FACT table that should be loaded into WAMART.DETAIL_FACT_EXT.

❑ 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.



Steps for Creating a SESSION_FACT_EXT Table

Step 1 – 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.

Step 2 - Modify the Sample Job weba_1335 for Loading the SESSION_FACT_EXT Table

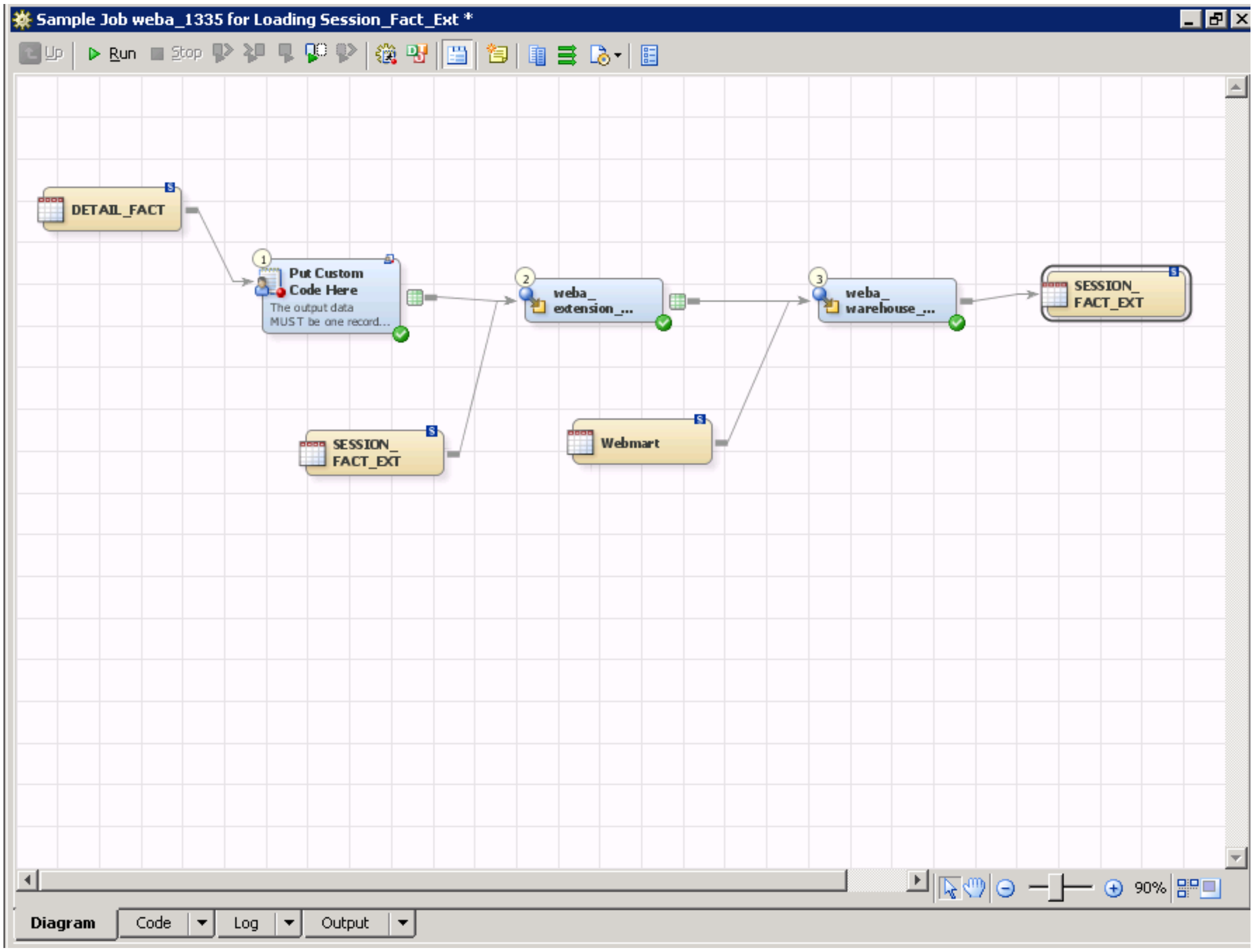
This *must* occur after the weba_1320_load_detail_fact job has been run. The following steps are required to load the SESSION_FACT_EXT table:

- □ Under the <*Web mart*>`/5.3.3 Jobs/Optional Jobs` folder, select s**ample 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 job with a new name that indicates where in the ETL the job should run.
  - Move the copy under the `5.3.3 Jobs/ETL Jobs` folder.
  - This job should run within the ETL after the weba_1320_load_detail_fact job is run.

- □ In the SQL Join transformation, select the columns from the WAWORK.SESSION_FACT table that should be loaded into WAMART.SESSION_FACT_EXT.
- □ 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.



Adding *dimension* Table Columns

Each dimension table load job appends a Work table to a dimension table and (optionally) updates records in the dimension table with the records that are supplied in a separate update table.

To add columns to the append:

1. Modify the appropriate WAMART dimension table to add the required fields.
2. Modify the Work data set in the dimension table load job to ensure that it contains the new required fields. The Work 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*.

   In SAS Data Integration Studio, right-click the **weba_warehouse_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.
3. Run the job and make sure there are no warnings or errors.
4. If necessary, redeploy the job for scheduling.

The weba_warehouse_table_loader enables you to update one field on a target table. Complete these steps in SAS Data Integration Studio:

1. Add an input data set to the weba_warehouse_table_loader. The input data set should contain a key field and the field to update. All keys that are supplied in this data set should exist in the target table with the same names and data characteristics.
2. On the **Options** tab for the weba_warehouse_table_loader, set **Key Field** to the name of the key field to use. Set **Field to update** to the name of the field that you want to update.
3. Run the job.
4. If necessary, reschedule the job.

Adding Other Key Performance Indicators to the weba_2500_prep_insight_monitor Job

There are two ways to add key performance indicators (KPIs) for use in the Performance Monitor and Insight Reports.

Create 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 key point to remember when adding KPIs is that the Insight Monitor Rollup transformation works by summing up all of the numbers that are passed into it by session_dt. Therefore, the input from the Insight Monitor Rollup transformation must consist of a data source with one record per date, and must consist of exclusively numeric fields in addition to the date itself (which is also numeric.)

To add KPIs, create a table that contains the requisite statistics and session_dt by using the SESSION_FACT table or any other table as a source. It is recommended that the user create a new job that accomplishes these two tasks:

1 Creates a new table that contains only the KPIs and session_dt. The KPIs must be numeric and summed by date.

2 Summarizes the new KPIs by session_dt and saves the results to a permanent data set.

The job has these two 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.

Modify the weba_2500_prep_insight_monitor Job

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 numerics can be summed to roll up the statistics by week or month. Thus, the 2500 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. Call the WEBA_Data_Step_Join Transformation 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:

locale - (leave blank)
name - variable name
label - descriptive label for metric

Troubleshooting the ETL Jobs

Health Check Jobs

Overview of the Heath Check Process

A health check is run automatically in two pieces. One job executes at the beginning of an ETL schedule, and another job runs at the ETL's successful execution. Each ETL job has its own set of options that can be finely tuned for threshold purposes, tests, and so on.

Health Checks Run at the Beginning of the Scheduled ETL

The health check job exits with an error status if it believes that a file is invalid or has already been loaded. You can then execute the next job in the list to override the error. Alternatively, you can take corrective action and rerun the health check job.

The health check keeps track of which files it has loaded or is in the process of loading, effectively preventing the same ETL from accidentally being triggered more than once.

This health check job accomplishes several things:

- obtains a list of data sets currently in its WASRC library
- reduces the list to those data sets having an appropriate prefix name
- ensures that each data set has required columns
- runs the previous health checks on each
- outputs a set of statistics on each input file to a data set in WACNFG containing input history, marking each valid data set as **in progress**
- creates a single data set for ETL's consumption

The job consists of several tables and a custom transformation WEBA_Input_Data_Health_Check. The transformation enables the user to customize the job via several options:

Record Threshold? Yes or No
enables or disables a health check that requires an input file to contain at least a specific number of records.

Threshold Percentage
a number from 0–100 representing the maximum allowed difference in record count between the current file and past history. The default is 95.

Threshold Definition
number of past successful file loads that should be used to determine history as defined in threshold percentage. The default is 7 days.

Input filename prefix
enters the prefix portion of the name of the data set that was created by the clickstream job.

Check for gaps between hours? Yes/No
checks data records for gaps in history.

Allowed datetime gaps in data
an integer that represents the maximum allowed gaps in terms of hours allowed.

Automatically adjust input file field lengths if necessary

adjusts field lengths to SAS Web Analytics specifications. Using this option avoids multiple warning messages in the log that indicate 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. It is not necessary to use UTF-8 input data to create a SAS Web Analytics data mart encoded as UTF-8; standard clickstream data can be used as input.
- To convert the clickstream data to UTF-8, it is not necessary to expand the columns.
- To convert the clickstream data to UTF-8 and also expand the columns, do not expand the columns to lengths greater than those expected by the SAS Web Analytics data mart. The data mart has already been expanded by approximately 30% on selected columns to accommodate MBCS data.

Post-ETL Health Check

This health check job is run after a successful ETL executes, provided it was part of the ETL schedule. The health check performs the following tasks:

- cleans out the signal file.
- turns off the **in progress** indicator in the history file.
- (optional) deletes the WASRC data sets that have been loaded. You can turn it on and off by setting the option in the WEBA_Post_ETL_Cleanup transformation properly.

Health Checks Performed

Table 3.5: Health Checks Performed

Check	Description	Action	Required (Yes/No/Partial)
Domain = ' '	Determine whether any record in Weblog detail data has DOMAIN equal to missing.	Abort check program - trigger failure of schedule.	Yes
Verify sort order	By default, Clickstream Data Surveyor should sort the Detail data set by SESSION_ID, DATE_TIME, and RECORD_ID. Verify that it is sorted.	Sort data set by session_id datetime record_id.	Yes
Check for SESSION_IDs that have rows where SESSION_CLOSED equals both 0 and 1	A SESSION_ID should be either closed (all its rows have SESSION_CLOSED=1) or open (all its rows have SESSION_CLOSED=0) within a single clickstream data set. A closed session is one whose activity is completely contained in the current ETL. An open session is one whose activity might be continued over into a later ETL. This situation can occur primarily when an ETL contains concatenated Clickstream Detail output.	Abort the program with ABEND to ensure job failure. Issue an error message stating that the records contain conflicting information.	Yes

Check	Description	Action	Required (Yes/No/Partial)
Ensure that input data exists	The WASRC library should contain the input Weblog Detail data set that will be processed within the ETL.	Abort the program with ABEND to ensure job failure.	No
Check for an identical file	Check to see whether the input Weblog data set resembles a previously loaded Weblog data set. The WaCnfg.cnfg_input_hist data set contains the following: process dttm number of records min record ID (contains both recordid and datetime) max record ID (contains both recordid and datetime) progress_check - status of warehouse load	Abort the check program if all fields match.	Partial
Weblog file is too small	A low number of records in the Weblog detail data set could indicate that there were Weblog cutting problems. 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.	Abort the check program to ensure job failure.	Yes
Missing data for time periods	See if there are records for all hours within the Detail data set, and error out if a “hole” is too large. For example, Weblogs are cut for a site at 0015 each day, and the expectation is that there would be records in each hour of the day. If no records reflect hours 0800–1000, then stop processing to determine what the issue is before continuing.	Abort the check program to ensure job failure.	No

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. In both cases, correct the input data or address the disk issues and rerun the job.

If the warehouse has been partially loaded, then the administrator will have to carefully roll back to a valid warehouse and rerun ETL from the top. To troubleshoot, read the SAS log, find the first error message, and correct it. Restore data as necessary, and repeat the process.

ETL Error Messages

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 – see the *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."

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. The following table lists all of the aggregate tables, a description of each, and a description for each field:

Table 4.1: Site Warehouse Aggregate Tables

Aggregate Table Name	Description
Daily Total Day Aggregate – Day	Various Web log statistics broken down by day and hour of day.
Daily Total Day Aggregate – Week	Various Web log statistics broken down by week and hour of day.
Daily Total Day Aggregate – Month	Various Web log statistics broken down by month and hour of day.
Daily Total Day Aggregate – Quarter	Various Web log statistics broken down by quarter and hour of day.
Daily Total Day Aggregate – Year	Various Web log statistics broken down by year and hour of day.
Hourly Status Aggregate – Day	Usage statistics aggregated by day, hour, and status code.
Hourly Status Aggregate – Week	Usage statistics aggregated by week, hour, and status code.
Hourly Status Aggregate – Month	Usage statistics aggregated by month, hour, and status code.
Hourly Status Aggregate – Quarter	Usage statistics aggregated by quarter, hour, and status code.
Hourly Status Aggregate -- Year	Usage statistics aggregated by year, hour, and status code.
Page Aggregate – Day	Page usage statistics aggregated by day, first requested page flag, valid page flag, and page.
Page Aggregate – Week	Page usage statistics aggregated by week, first requested page flag, valid page flag, and page.
Page Aggregate – Month	Page usage statistics aggregated by month, first requested page flag, valid page flag, and page.
Page Aggregate – Quarter	Page usage statistics aggregated by quarter, first requested page flag, valid page flag, and page.
Page Aggregate – Year	Page usage statistics aggregated by year, first requested page flag, valid page flag, and page.
Platform Aggregate – Day	Client platform information aggregated by day, platform, browser, and browser version number.
Platform Aggregate – Week	Client platform information aggregated by week, platform, browser, and browser version number.

Aggregate Table Name	Description
Platform Aggregate – Month	Client platform information aggregated by month, platform, browser, and browser version number.
Platform Aggregate – Quarter	Client platform information aggregated by quarter, platform, browser, and browser version number.
Platform Aggregate – Year	Client platform information aggregated by year, platform, browser, and browser version number.
Referrer Search Term Aggregate – Day	Referrer and search term usage aggregated by day, referrer, search term, and search engine.
Referrer Search Term Aggregate – Week	Referrer and search term usage aggregated by week, referrer, search term, and search engine.
Referrer Search Term Aggregate – Month	Referrer and search term usage aggregated by month, referrer, search term, and search engine.
Referrer Search Term Aggregate – Quarter	Referrer and search term usage aggregated by quarter, referrer, search term, and search engine.
Referrer Search Term Aggregate – Year	Referrer and search term usage aggregated by year, referrer, search term, and search engine.
Aggr: SEBM – Day	Search Engine Bid data aggregated by day, page, search term, and ppc_flag.
Aggr: SEBM – Week	Search Engine Bid data aggregated by week, page, search term, and ppc_flag.
Aggr: SEBM – Month	Search Engine Bid data aggregated by month, page, search term, and ppc_flag.
Aggr: SEBM – Quarter	Search Engine Bid data aggregated by quarter, page, search term, and ppc_flag.
Aggr: SEBM – Year	Search Engine Bid data aggregated by year, page, search term, and ppc_flag.
Aggr: SEB_CAMPAIGN_DAY	Campaign information aggregated by day, search engine, campaign, and page.
Aggr: SEB_CAMPAIGN_WEEK	Campaign information aggregated by week, search engine, campaign, and page.
Aggr: SEB_CAMPAIGN_MONTH	Campaign information aggregated by month, search engine, campaign, and page.
Aggr: SEB_CAMPAIGN_QUARTER	Campaign information aggregated by quarter, search engine, campaign, and page.
Aggr: SEB_CAMPAIGN_YEAR	Campaign information aggregated by year, search engine, campaign, and page.
Visitor Aggregate – Day	Visitor information aggregated by date and visitor.
Visitor Aggregate – Week	Visitor information aggregated by week and visitor.
Visitor Aggregate – Month	Visitor information aggregated by month and visitor.

Aggregate Table Name	Description
Visitor Aggregate – Quarter	Visitor information aggregated by quarter and visitor.
Visitor Aggregate – Year	Visitor information aggregated by year and visitor.

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 4.2: 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.
Daily Total Day Aggregate – Day	status_code_302_count	Number of times status code 302 was displayed.
Daily Total Day Aggregate – Day	status_code_304_count	Number of times status code 304 was displayed.
Daily Total Day Aggregate – Day	status_code_400_count	Number of times status code 400 was displayed.
Daily Total Day Aggregate – Day	status_code_401_count	Number of times status code 401 was displayed.
Daily Total Day Aggregate – Day	status_code_403_count	Number of times status code 403 was displayed.
Daily Total Day Aggregate – Day	status_code_404_count	Number of times status code 404 was displayed.

Table	Variable	Description
Daily Total Day Aggregate – Day	status_code_405_count	Number of times status code 405 was displayed.
Daily Total Day Aggregate – Day	status_code_408_count	Number of times status code 408 was displayed.
Daily Total Day Aggregate – Day	status_code_500_count	Number of times status code 500 was displayed.
Daily Total Day Aggregate – Day	status_code_501_count	Number of times status code 501 was displayed.
Daily Total Day Aggregate – Day	daily_sessions	Number of visits over a 24-hour span.
Page Aggregate – Day	session_count	Number of visits.
Page Aggregate – Day	page_count	Number of page views.
Page Aggregate – Day	entry_page_count	Number of times a page was an entry page for a visit.
Page Aggregate – Day	exit_page_count	Number of times a page was an exit page for a visit.
Page Aggregate – Day	prosp_bounce	Number of first valid pages opened in a visit (prospective bounces).
Page Aggregate – Day	one_hit_session_count	Number of bounces.
Page Aggregate – Day	entry_total	Number of total entry pages for a day (used with entry_page_count to calculate percentages).
Page Aggregate – Day	exit_total	Number of total exit pages for a day (used with exit_page_count to calculate percentages).
Page Aggregate – Day	session_total	Number of visits for a day (used with session_count to calculate percentages).
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.

Table	Variable	Description
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.

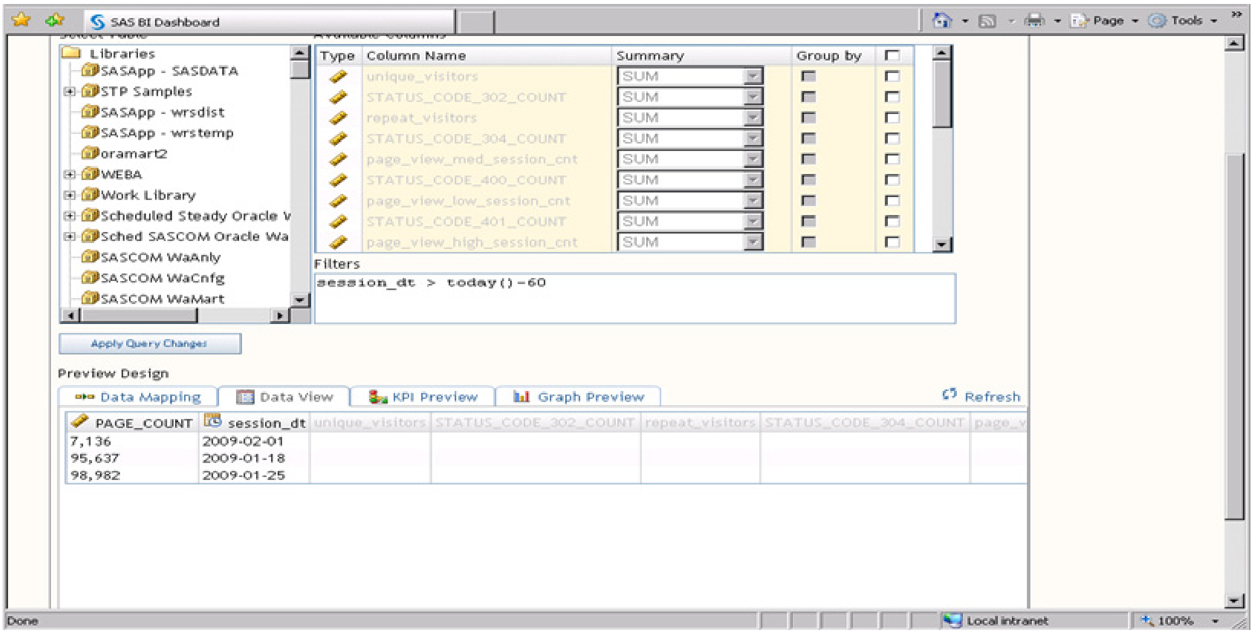
Example: Setting Up a Dashboard

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 62.

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**.



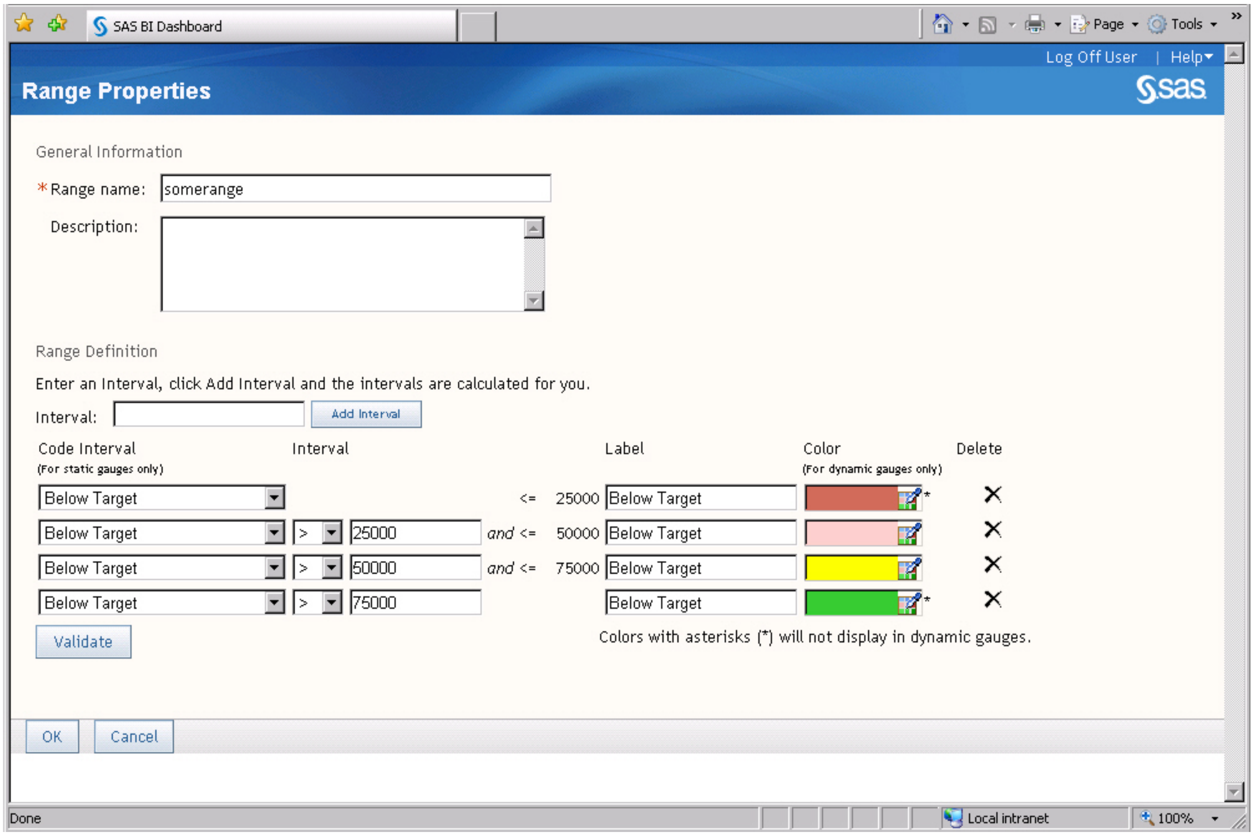
Type	Column Name	Summary	Group by	□
	unique_visitors	SUM	□	□
	STATUS_CODE_302_COUNT	SUM	□	□
	repeat_visitors	SUM	□	□
	STATUS_CODE_304_COUNT	SUM	□	□
	page_view_med_session_cnt	SUM	□	□
	STATUS_CODE_400_COUNT	SUM	□	□
	page_view_low_session_cnt	SUM	□	□
	STATUS_CODE_401_COUNT	SUM	□	□
	page_view_high_session_cnt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

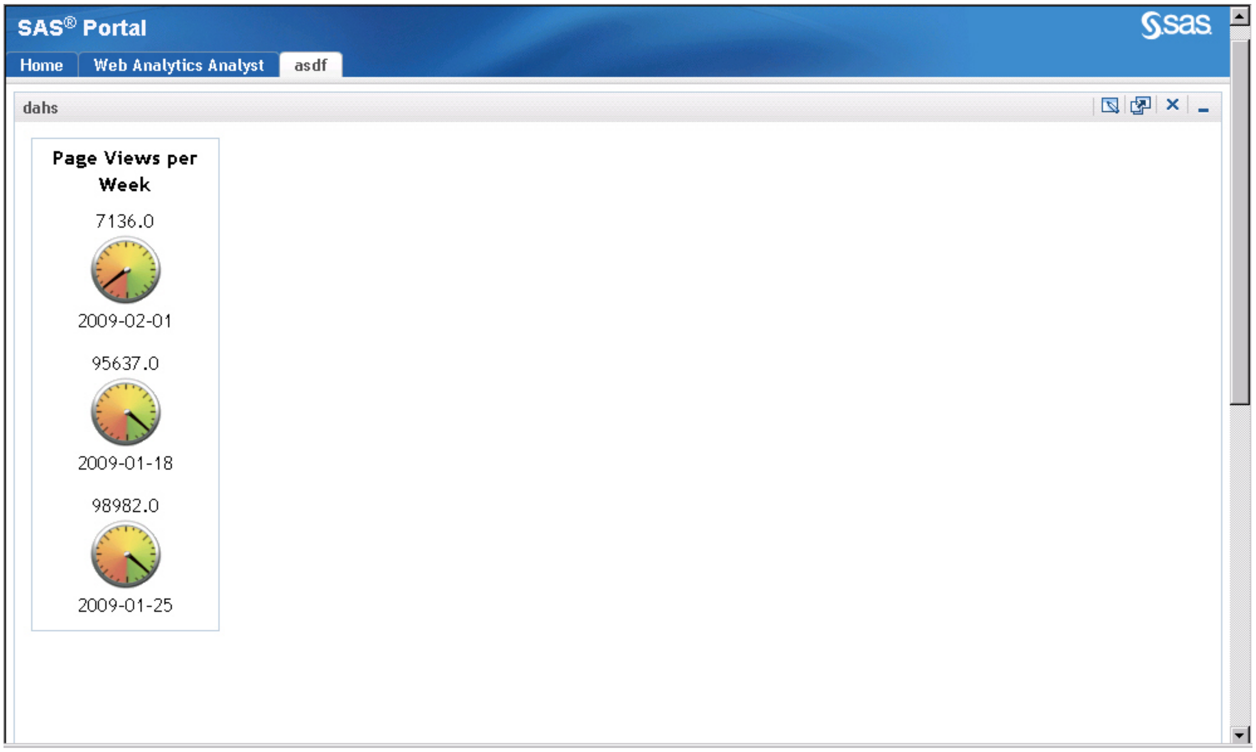
Enter an Interval, click Add Interval and the intervals are calculated for you.

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 5.3.3 provides numerous SAS Web Report Studio reports, shipped with the product, which are built on SAS Information Maps. These information maps are more advanced than typical ones, because they must support a wider set of requirements than a custom information map that you might build for your specific report.

The SAS Information Maps that are shipped with SAS Web Analytics 5.3.3 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`**.

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_bouncerate.sas	aggr_page_day and both fact tables
Browsers	map_weba_browser_platform	stp_weba_platform_browser	stp_weba_platform_browser.sas	aggr_platform
DayofWeek	map_weba_daily_total	stp_weba_daily_total	stp_weba_daily_total.sas	aggr_daily_total
Error Status	map_weba_pages_client_error	stp_weba_pages_client_error	stp_weba_pages_client.sas	aggr_page
Exit Pages	map_weba_pages_exit_pages	map_weba_pages_exit_pages	stp_weba_pages.sas	aggr_page
Hourly Metrics	map_weba_daily_total	stp_weba_daily_total	daily_total.sas	aggr_daily_total
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 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_page
Platform	map_weba_browser_platform	stp_weba_platform_browser	stp_weba_platform_browser.sas	aggr_platform
Top Referrer Entry Pages	map_weba_pages_entry_pages	stp_weba_pages_entry_pages	stp_weba_pages.sas	aggr_page
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
Status Codes Per Hour	map_weba_status_codes	stp_weba_status_codes	stp_weba_status_codes.sas	aggr_hourly_status
Status Codes	map_weba_status_codes	stp_weba_status_codes	stp_weba_status_codes.sas	aggr_hourly_status
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_page
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, and aggregates because adverse effects could result. 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 a SAS Data Integration Studio template job that the SAS administrator can use to add SAS Web Analytics content to the SAS Information Delivery Portal. The job adds pages for the SAS Web Analytics Analyst and SAS Web Analytics Report Users groups, and is described in the following sections.

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:

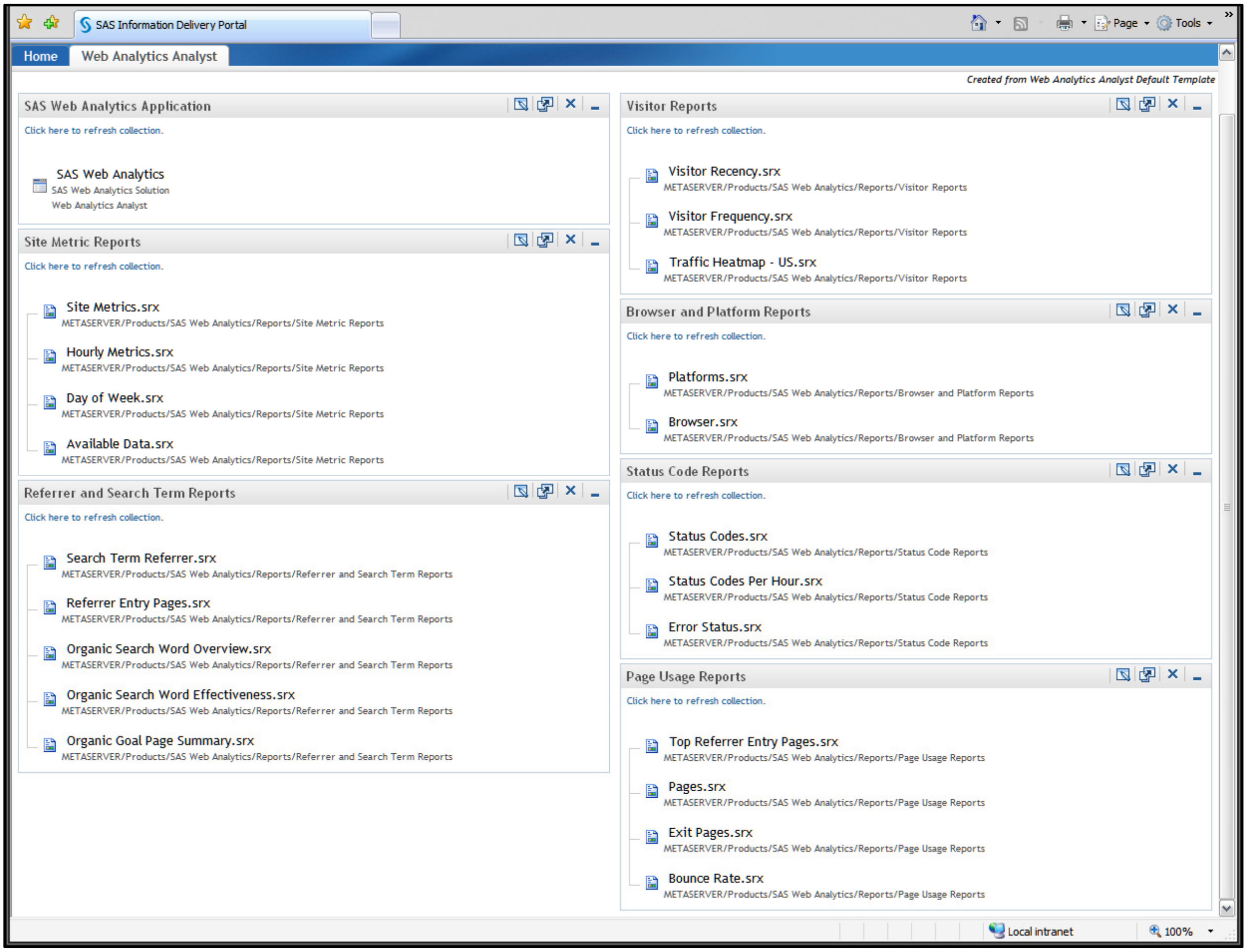
1. Create the SAS Web Analytics portal pages. This step must be completed by a SAS Portal administrator. See the next section for detailed instructions.
2. Log on to the SAS Information Delivery Portal. The SAS Web Analytics Portal page that displays on one of the tabs is appropriate for your group. If your group is SAS Web Analytics Analyst, then the Analyst page displays. If your group is SAS Web Analytics Report Users, then the Reports page displays.

Create SAS Web Analytics Portal Pages

Create the Template Pages

Running the Create Portal Template Page job in SAS Data Integration Studio creates two portal template pages for these roles:

- ❑ Web Analytics Analyst. This page contains a link to the SAS Web Analytics Web application and all SAS Web Report Studio reports.



Created from Web Analytics Analyst Default Template

SAS Web Analytics Application

Click here to refresh collection.

SAS Web Analytics
SAS Web Analytics Solution
Web Analytics Analyst

Site Metric Reports

Click here to refresh collection.

Site Metrics.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Hourly Metrics.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Day of Week.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Available Data.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Referrer and Search Term Reports

Click here to refresh collection.

Search Term Referrer.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Referrer Entry Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Search Word Overview.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Search Word Effectiveness.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Goal Page Summary.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Visitor Reports

Click here to refresh collection.

Visitor Recency.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Visitor Frequency.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Traffic Heatmap - US.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Browser and Platform Reports

Click here to refresh collection.

Platforms.srx
METASERVER/Products/SAS Web Analytics/Reports/Browser and Platform Reports

Browser.srx
METASERVER/Products/SAS Web Analytics/Reports/Browser and Platform Reports

Status Code Reports

Click here to refresh collection.

Status Codes.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Status Codes Per Hour.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Error Status.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Page Usage Reports

Click here to refresh collection.

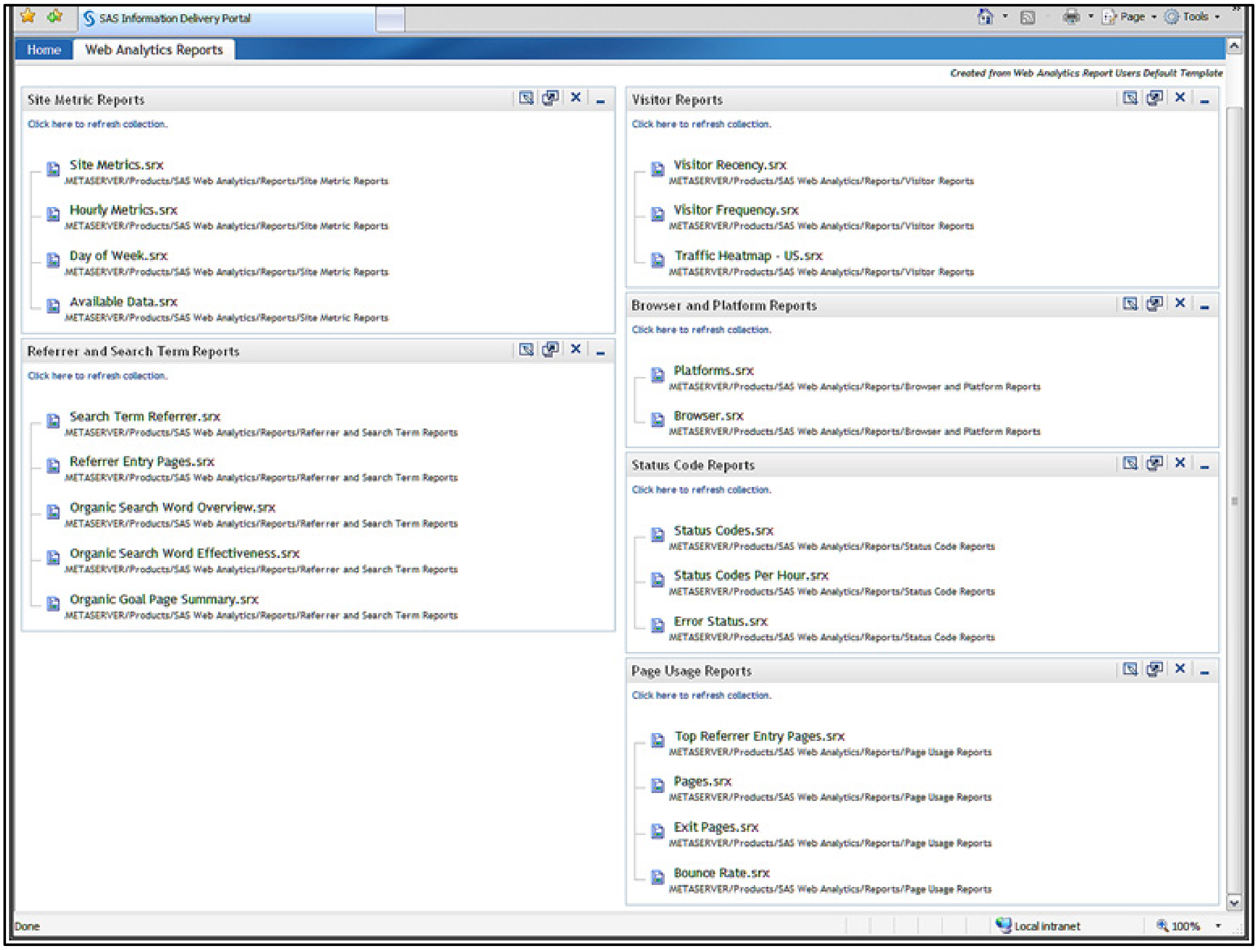
Top Referrer Entry Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Exit Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Bounce Rate.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

❑ Web Analytics Reports. This page contains a link to the SAS Web Analytics SAS Web Report Studio reports.



Created from Web Analytics Report Users Default Template

Site Metric Reports

Click here to refresh collection.

Site Metrics.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Hourly Metrics.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Day of Week.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Available Data.srx
METASERVER/Products/SAS Web Analytics/Reports/Site Metric Reports

Referrer and Search Term Reports

Click here to refresh collection.

Search Term Referrer.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Referrer Entry Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Search Word Overview.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Search Word Effectiveness.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Organic Goal Page Summary.srx
METASERVER/Products/SAS Web Analytics/Reports/Referrer and Search Term Reports

Visitor Reports

Click here to refresh collection.

Visitor Recency.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Visitor Frequency.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Traffic Heatmap - US.srx
METASERVER/Products/SAS Web Analytics/Reports/Visitor Reports

Browser and Platform Reports

Click here to refresh collection.

Platforms.srx
METASERVER/Products/SAS Web Analytics/Reports/Browser and Platform Reports

Browser.srx
METASERVER/Products/SAS Web Analytics/Reports/Browser and Platform Reports

Status Code Reports

Click here to refresh collection.

Status Codes.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Status Codes Per Hour.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Error Status.srx
METASERVER/Products/SAS Web Analytics/Reports/Status Code Reports

Page Usage Reports

Click here to refresh collection.

Top Referrer Entry Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Exit Pages.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

Bounce Rate.srx
METASERVER/Products/SAS Web Analytics/Reports/Page Usage Reports

For more information about portal template pages, see the SAS Information Delivery Portal online Help that is accessible from within the product.

Requirements

Before you run the page creation job, make sure you have satisfied these requirements:

- ❑ The SAS Information Portal must be started at least once. If the job is run before you start the portal, some of the referenced portlet templates will not be present yet and you will produce this error: `The object reference to Prototype was requested without an identifier`.
- ❑ SAS Web Analytics identity groups must be created. For a list of user identity groups and names that are created during configuration of SAS Web Analytics, see Chapter 1, “Introduction to SAS Web Analytics”.
- ❑ For new user identity groups, log on to the SAS Information Delivery Portal as a SAS Trusted User, or use another user ID, which is a portal administrator, in order to initialize the necessary metadata permissions folder structure for the portal.

Note: It is best practice to first create all user groups that will be used for accessing the SAS Information Delivery Portal, and then to log on to the SAS Portal as a SAS Trusted User.

Note: Search Engine Bid reports will not appear within the portal template page automatically if the SEBD initialization job has not been run.

Run the Portal Page Creation Job

To run the job:

1. Open SAS Data Integration Studio.
2. Go to the **/Programs/SAS Web Analytics/Sample Jobs** folder.
3. Open Create Portal Template Pages.
4. Open the weba_portal_page_creation transformation and click the **Options** tab:
   a. Select Page type:
      - ❑ Web Analytics Reports – this portal page contains a link to each of the SAS Web Report Studio reports associated with SAS Web Analytics.
      - ❑ Web Analytics Analyst – this portal page contains a link to each of the SAS Web Report Studio reports associated with SAS Web Analytics and a link to the SAS Web Analytics Web application.

   b. User group selection occurs automatically when a page type is selected. If you do not want to use the default user group, enter the user group to be used.

   c. Select share type - specifies whether the page is Default or Sticky. For more information about page share type, see the SAS Information Delivery Portal Help.
      - ❑ Default: Default group pages are automatically added to the portal Web application of all users in the group and users can remove them.
      - ❑ Sticky: Sticky group pages are automatically added to the portal Web application of all users in the group and users cannot remove them.

   d. Enter page rank: The default is 100. For more information about page rank, see the SAS Information Delivery Portal Help.

   e. Log on as a SAS Trusted User, or use a portal administrator user ID. For information about portal administrators, see the SAS Information Portal chapter in the *SAS 9.2 Intelligence Platform: Web Application Administration Guide*.

   f. Enter a password for the SAS Trusted User or portal administrator user ID.

   g. Click **OK** to close the transformation and save your selections.
5. Run the program by selecting the **Run** arrow. For more information, see the SAS Data Integration Studio Help.
6. Repeat steps 3 and 4 to create the other portal template page.

Note:

- ❑ The job will end with errors if you have created user groups but not accessed the SAS Information Delivery Portal as recommended in the requirements section.

- ❑ The following warning can be ignored: **WARNING: The quoted string currently being processed has become more than 262 characters long. You may have unbalanced quotation marks.** The warning can be turned off by setting the NOQUOTELENMAX system option.
- ❑ The job will end with errors if the portal template page already exists.

Making the SEBD Reports Available on the Portal Page

The SEBD reports are created after the SEBD Initialization job is run. If the SAS Web Analytics portal page is created before the SEBD Initialization job is run, then the SEBD reports cannot be viewed on the portal page. To view the SEBD reports on the portal page:

1 Delete the portal template page in the SAS Information Delivery Portal. (For more information, see the SAS Information Delivery Portal online Help):

   a. Log on to the SAS Information Delivery Portal as the content administrator for the group that shares the page template.

   b. Search for and delete the page template.

   c. Specify whether you want to delete all of the pages that were created from the template.

2 Rerun the Create Portal Template Page job. For more information, see “Run the Portal Page Creation Job” on page 74.

Making the SAS Information Delivery Portal Page Available

To make available the template that was created by the Create Portal Template Page job that is described on page 61:

1. Log on to the SAS Information Delivery Portal as a user.
2. Select **Options** ▸ **Tools** ▸ **Manage Page History**.
3. On the Manage Page History page, click **Clear History**.
4. Log off the SAS Information Delivery Portal and log on again as a user.

The Web Analytics Analyst tab is now available.

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. Each search engine provider offers 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 5.3.3, 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 -sysin`** <*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.

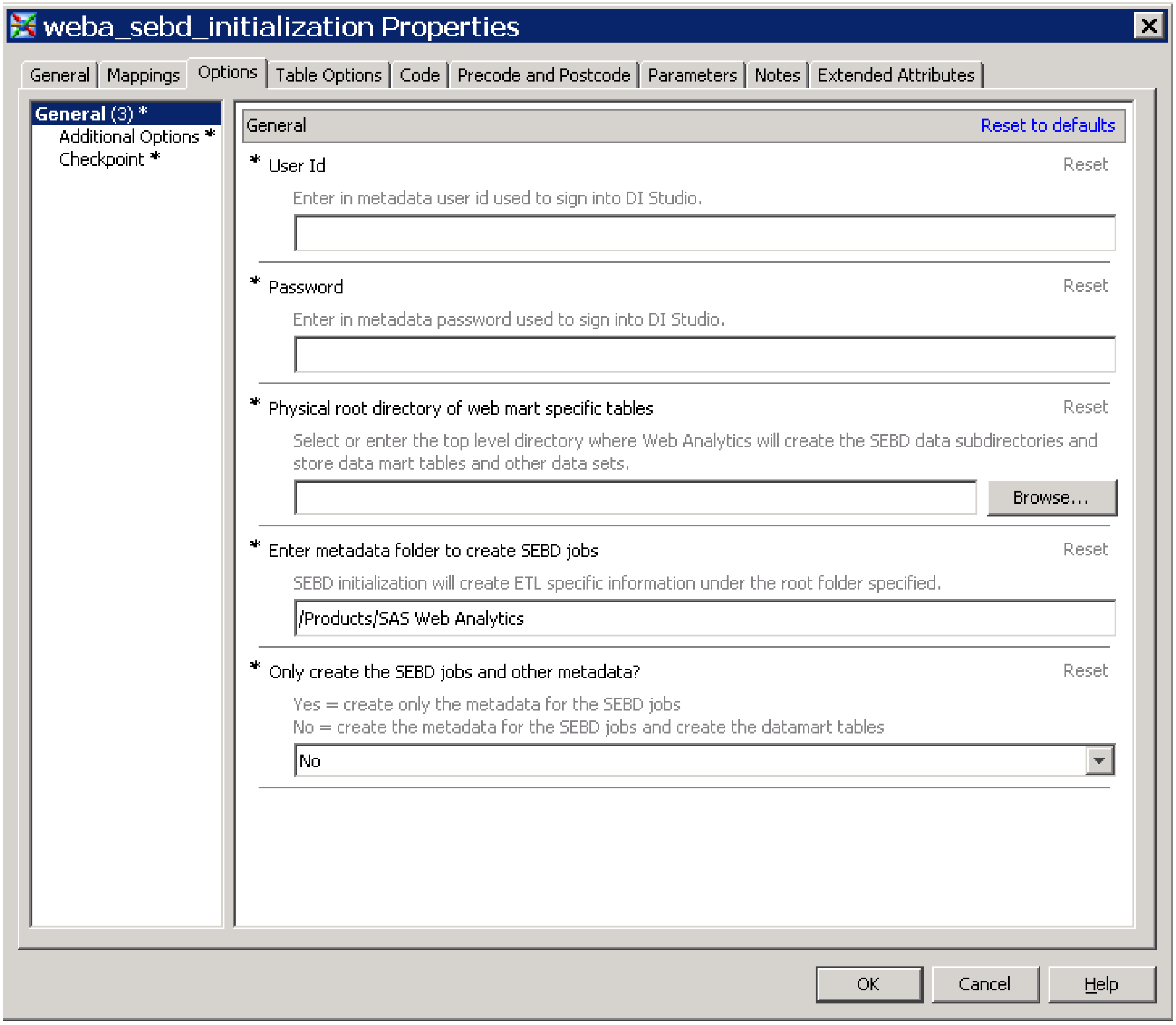
   b. **Password** – enter the password that is associated with the user ID.

   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.



* 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

f. Click **OK.**

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 79.)

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

❑ **`5.3.3 Jobs`** - see Table 5.1 on page 95 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**.

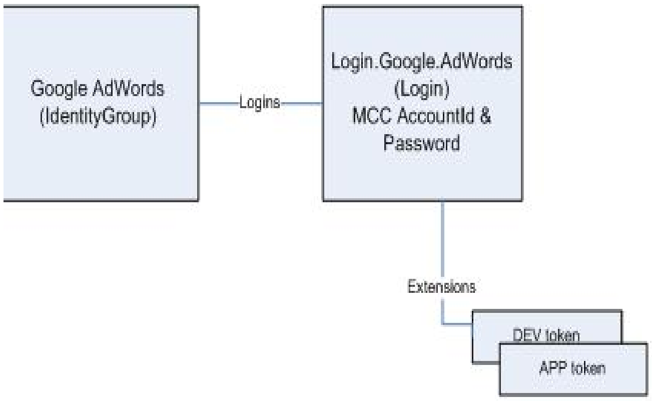
Managing Google AdWords My Client Center Account Login Information

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, 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>";
;
        %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;
    %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 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

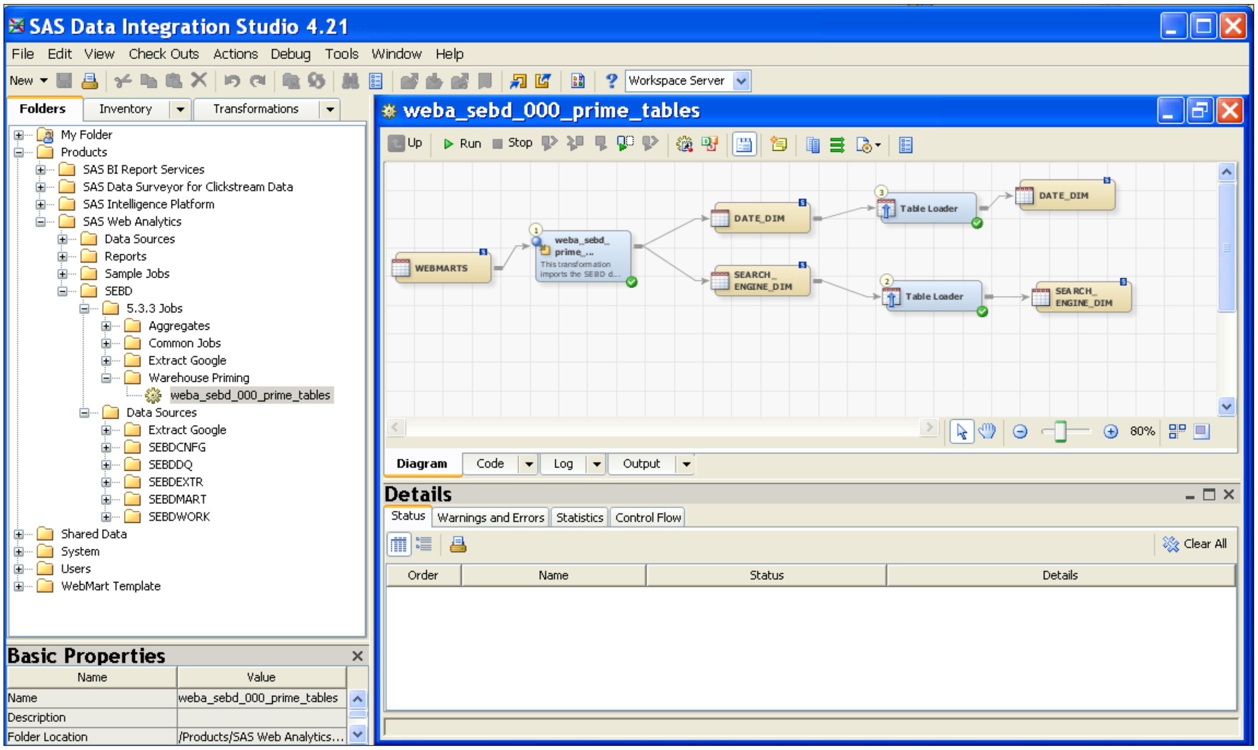
Verify that the SAS Admin user that was specified has Unrestricted metadata access. The default user is sasadm@saspw, which is the preferred user for this job.

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 79.

2 Double-click the **weba_sebd_000_prime_tables** job in **`/SEBD/5.3 Jobs/Warehouse/Priming`**).



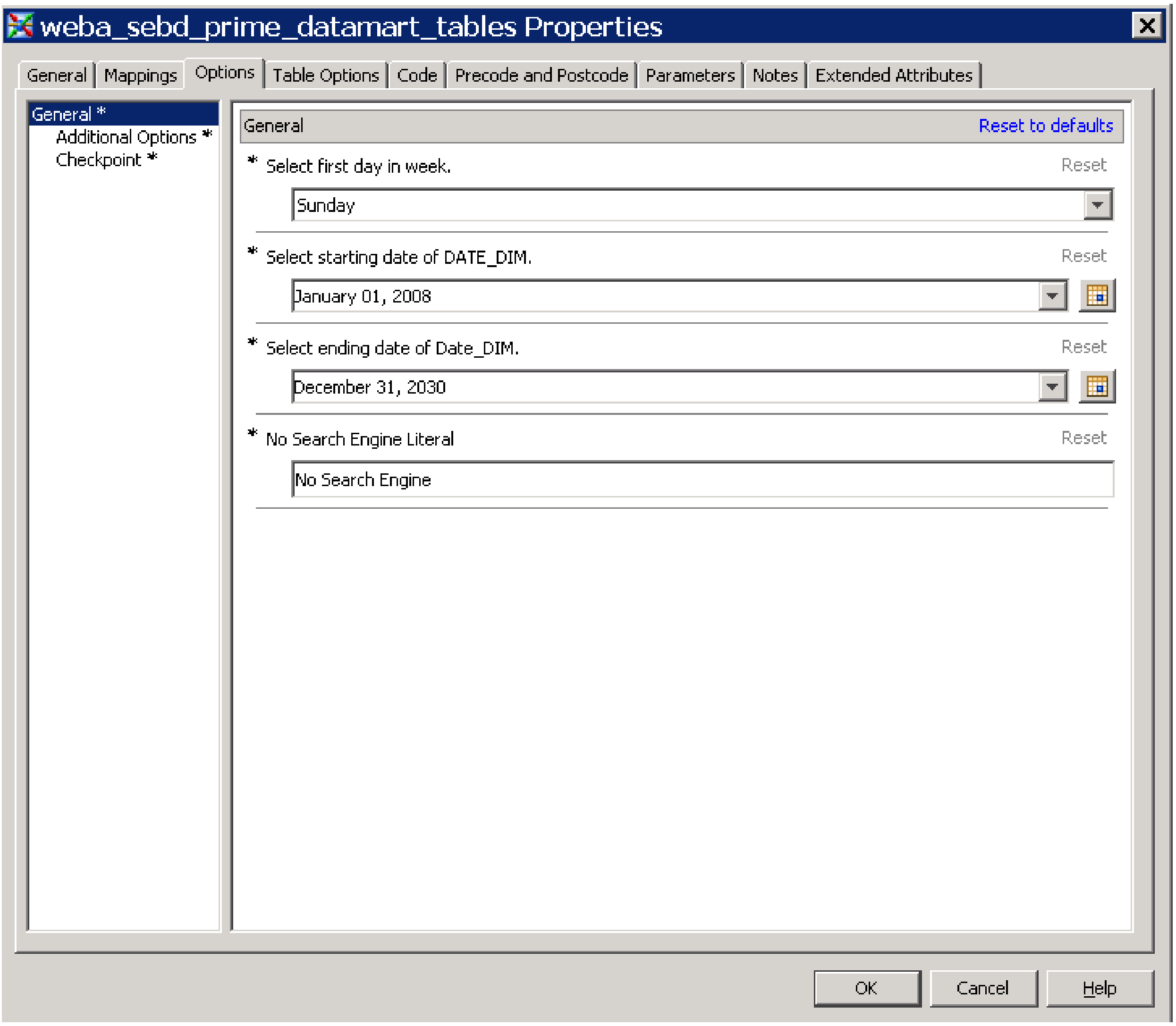
Basic Properties

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

3 Double-click the **weba_sebd_prime_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

General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes

General *
Additional Options *
Checkpoint *

General — Reset to defaults

* Select first day in week. — Reset: Sunday

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

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

* No Search Engine Literal — Reset: 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 Google Learning Center Lessons 2a, b, and c 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.

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

Set Up the SEBD Mart ETL

Overview

All SEBD ETL jobs are located in SAS Data Integration Studio in the **`/Products/SAS Web Analytics/SEBD/5.3 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 5.3.3 Jobs/Common Jobs	No	None		Creates ZZ_* versions of all SEBDMART tables to be updated if necessary.
webd_sebd_002_google_ adwords_extract	/SEBD 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	No	None	weba_sebd_029_create_zz_tables	Loads PPCWORK.Destination_Domain_Dim with new records.
weba_sebd_042_google_load_destination_page_dim	/SEBD 5.3.3 Jobs/Extract Google/ Adwords vXX	No	None	weba_sebd_041_google_load_destination_domain_dim	Loads ZZ_DESTINATION_PAGE_DIM with new records.
weba_sebd_043_google_load_campaign_dim	/SEBD 5.3.3 Jobs/Extract Google/ Adwords vXX	No	None	weba_sebd_042_google_load_destination_page_dim	Loads ZZ_CAMPAIGN_DIM with new records and updates existing records with current information.

Job Name	Folder	Optional?	Associated Job	Dependency	Description
weba_sebd_044_google_load_ ad_group	/SEBD 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	No	None	weba_sebd_044_ google_load_ad_ group_dim	Loads ZZ_AD_CREATIVE_DIM with new records and updates existing records with current information.
weba_sebd_046_google_load_ keyword_dim	/SEBD 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 Jobs/Extract Google/Adwords vXX	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 5.3.3 Jobs/Extract Google/ Adwords vXX	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 5.3.3 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 5.3.3 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 5.3.3 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 5.3.3 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, **paignManagement?**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:

1 Run the Oracle client installation.

2 Click **Next**.

3 When prompted for which type of installation, select **Administrator** and click **Next**.

4 Choose a name and path for the installation and click **Next**.

5 Follow the instructions throughout; do not add any database or test connections now.

6 Navigate to **Start ->All programs->Oracle (name) ->Configuration** and **Migration Tools ->Oracle Net Configuration Assistant**.

7 Select **Local Net Service Name Configuration** and click **Next.** Click **Add**, and then click **Next**.

8 Enter the Oracle service name, **network protocol** is TCP; and host name is <***your host machine name***>. Keep the standard ports.

9 When prompted to test the connection, change the user ID and password to oramart/oramart1. The test connection should be working.

10 Repeat with these parameters for the other database.

11 Exit the application.

12 In Windows, select **My Computer**. Right-click **Start** and select **Properties**.

13 Click the **Advanced** tab, and then select **Environment Variables**.

14 Make sure that the Path variable contains the pathname to the location where you installed the client, followed by bin. For example, a valid pathname could be C:\Oracle_client\bin.

15 Add an environment variable named ORACLE_HOME. The value should be the same as the piece added to Path, but without \bin. For example, C:\Oracle_Client.

16 Log out of the machine and log back on. Select **Start->Run.** Enter **cmd.** Then enter **sqlplus.** A logon screen is displayed, which indicates that the client is working.

Enter the user name and password. Enter **quit**. Make sure you can log on to the other database, too.

17 If you can log on, then it works for your user ID alone. Navigate to the place where the client is installed (for example, C:\Oracle_Client) and set the permissions of the root directory. Add Everyone. Administrators and Everyone should have full control.

18 Make sure other team members can log on. They should be able to open a Base SAS session, enter the following, and execute the statement to get an assigned library:

```
libname a oracle user=oramart pass=oramart1 path=<pathname>;
```

Note: If the Oracle System ID (SID) and the PATH name differ in the tnsnames.ora file, then an extended attribute for oracle_service_name needs to be added 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 and SAS Digital Marketing send identifiers for customers (also called subjects) as subject IDs, and send identifiers for campaigns as response tracking codes. The response tracking codes identify SAS Marketing Automation cell_package and business context. These identifiers are added to e-mail broadcasts that contain links to target Web sites. The e-mails are sent to customers as part of a campaign.

Note: Although the flow of information can be fully configured, this is the recommended best practice.

An e-mail recipient has several options after receiving an e-mail broadcast. If a recipient opens the e-mail and clicks the link to a Web site (known as a click-through), the subject ID and response tracking code are sent to SAS Web Analytics through one of the following methods:

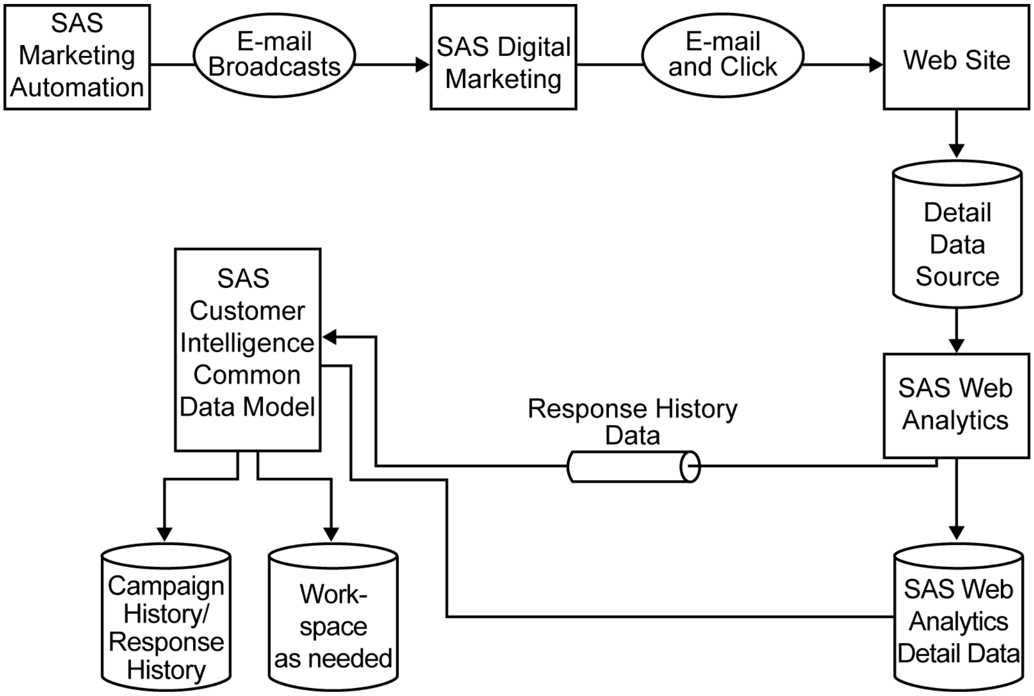
- ❑ SAS Data Surveyor for Clickstream Data , which processes the Web site's Web logs.
- ❑ If the SAS Tag Data Format is used for page tagging, a data collection server collects the output and writes the output to page tagging logs.

After an e-mail recipient arrives at a destination Web page in a target Web site, SAS Web Analytics can track online behavior details for the campaign.

Lastly, for campaigns that are reached (that is, the e-mail recipient arrives at a destination Web page) SAS Web Analytics returns the subject ID and response 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.

How Data Is Updated between the Applications

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



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.

For configuration information, see the *SAS Digital Marketing 5.3: Administrator's Guide* and the *SAS Marketing Automation 5.3: 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.

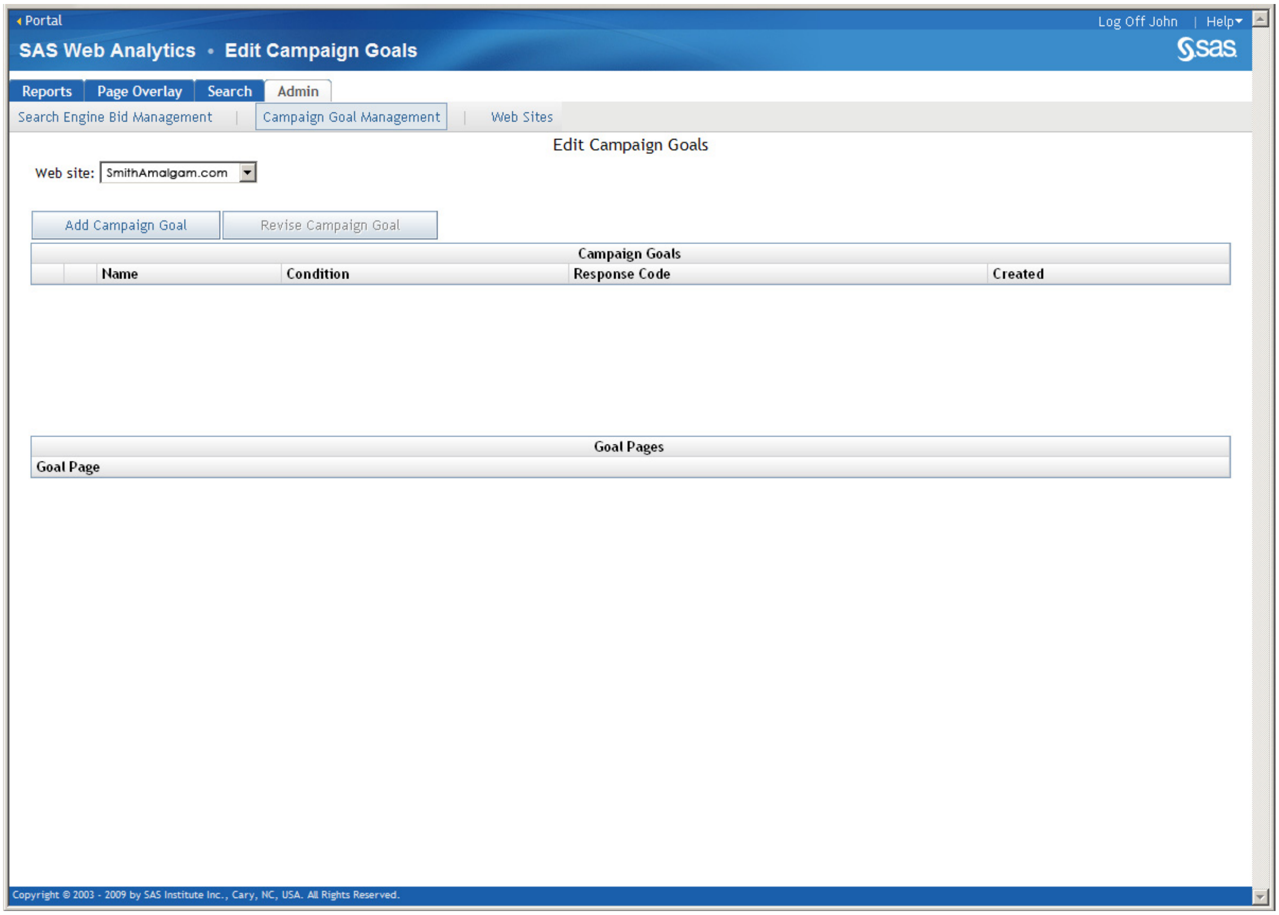
2 Define parameters for the job that updates customer response history in SAS Marketing Automation (weba_2550_ma_response_history_update).

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 5.3: User's Guide, Second Edition*.

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

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



Web site: SmithAmalgam.com

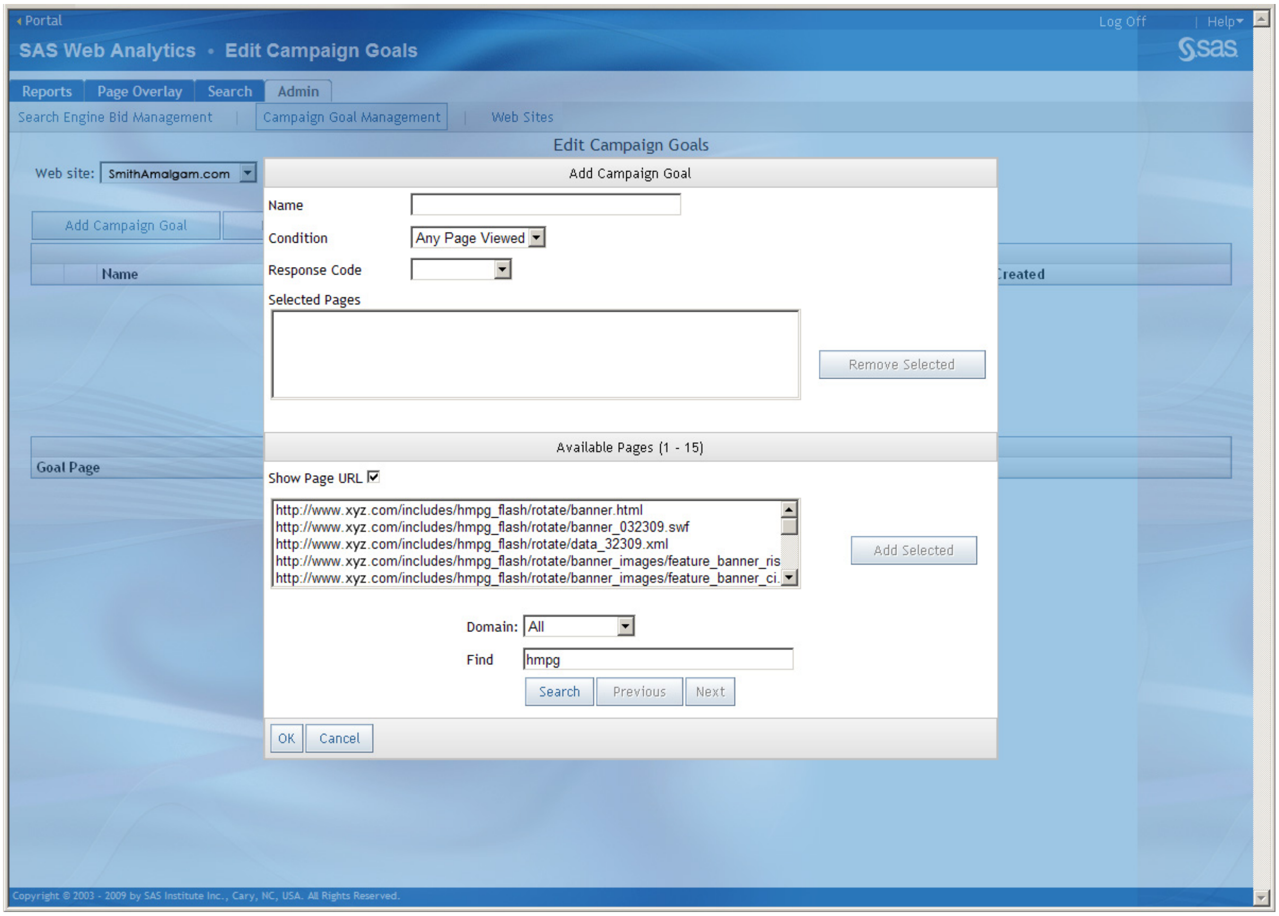
Campaign Goals

		Name	Condition	Response Code	Created

Goal Pages

Goal Page

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



Web site: SmithAmalgam.com

Name

Condition: Any Page Viewed

Response Code

Selected Pages

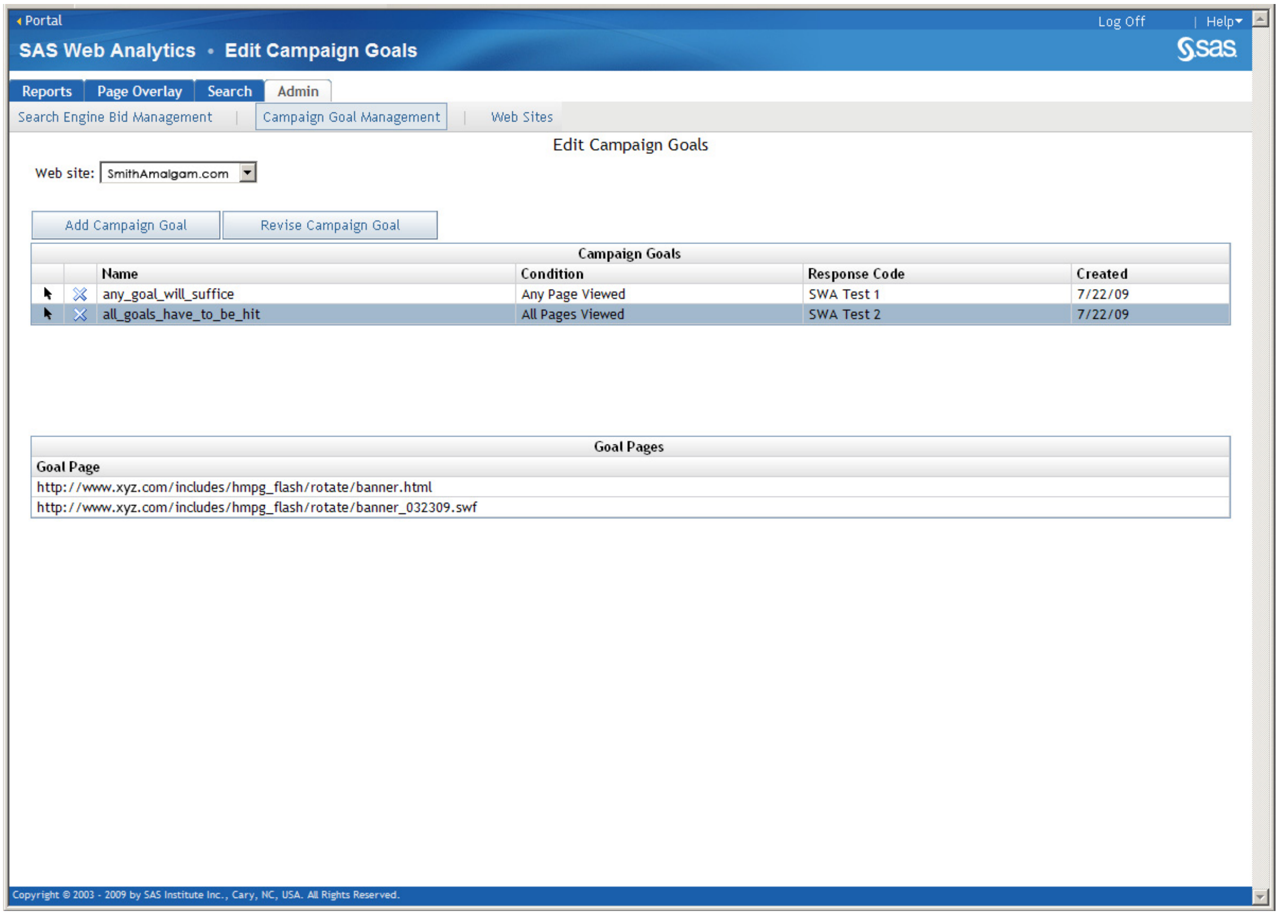
Available Pages (1 - 15)

Show Page URL ☑

```
http://www.xyz.com/includes/hmpg_flash/rotate/banner.html
http://www.xyz.com/includes/hmpg_flash/rotate/banner_032309.swf
http://www.xyz.com/includes/hmpg_flash/rotate/data_32309.xml
http://www.xyz.com/includes/hmpg_flash/rotate/banner_images/feature_banner_ris
http://www.xyz.com/includes/hmpg_flash/rotate/banner_images/feature_banner_ci.
```

Domain: All

Find: hmpg



Campaign Goals

		Name	Condition	Response Code	Created
		any_goal_will_suffice	Any Page Viewed	SWA Test 1	7/22/09
		all_goals_have_to_be_hit	All Pages Viewed	SWA Test 2	7/22/09

Goal Pages

Goal Page
http://www.xyz.com/includes/hmpg_flash/rotate/banner.html
http://www.xyz.com/includes/hmpg_flash/rotate/banner_032309.swf

Note that when the condition **Any Page Viewed** is selected, the visitor can view any of the pages selected in order or the goal to be reached. If **All Pages Viewed** 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 Click **Revise Campaign Goal**.

2 In the window, 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

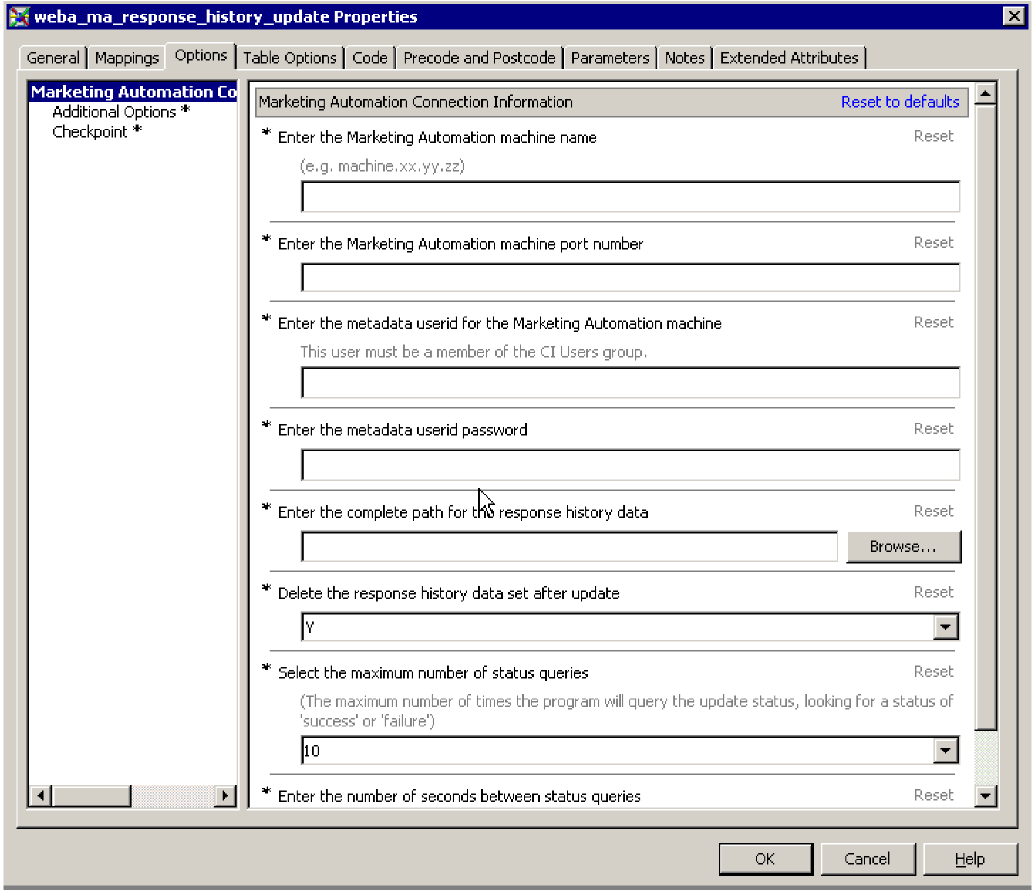
The SAS Web Analytics job weba_2550_ma_response_history_update must be modified to supply the information that is necessary to complete the updates in SAS Marketing Automation.

1 Open SAS Data Integration Studio and navigate to **`Shared Data/Web Analytics/`**`<`*Web mart*`>`**`/5.3.3 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

Marketing Automation Connection Information

* Enter the Marketing Automation machine name Reset
(e.g. machine.xx.yy.zz)

* Enter the Marketing Automation machine port number Reset

* Enter the metadata userid for the Marketing Automation machine Reset
This user must be a member of the CI Users group.

* Enter the metadata userid password Reset

* Enter the complete path for the response history data Reset

* Delete the response history data set after update Reset
Y

* Select the maximum number of status queries Reset
(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 Reset

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=Y.

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 5.3: 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 “Required ETL Jobs” in Chapter 3.

Relationship of the weba_1200_warehouse_staging_tables Job to Response History Updates

Three 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_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_source_id

at this time, the only value is SDM, for SAS Digital Marketing. This value is set during clickstream data processing.

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 when an e-mail recipient clicks on a link in an e-mail broadcast sent by SAS Digital Marketing. 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, four 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

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.

Columns Assigned by SAS Web Analytics

respTracking_cd (from Weblog, WaMart.detail_fact.entry_action_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

Columns Assigned by SAS Marketing Automation

tmtTracking_cd
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 four WaWork data sets WaWork.rh_all_goalsmet, WaWork.rh_any_goalsmet, WaWork.rh_sessions_for_all_goalsmet, and WaWork.rh_update_activegoals_processed were created successfully.

❑ The user ID that is running the program has Write 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.

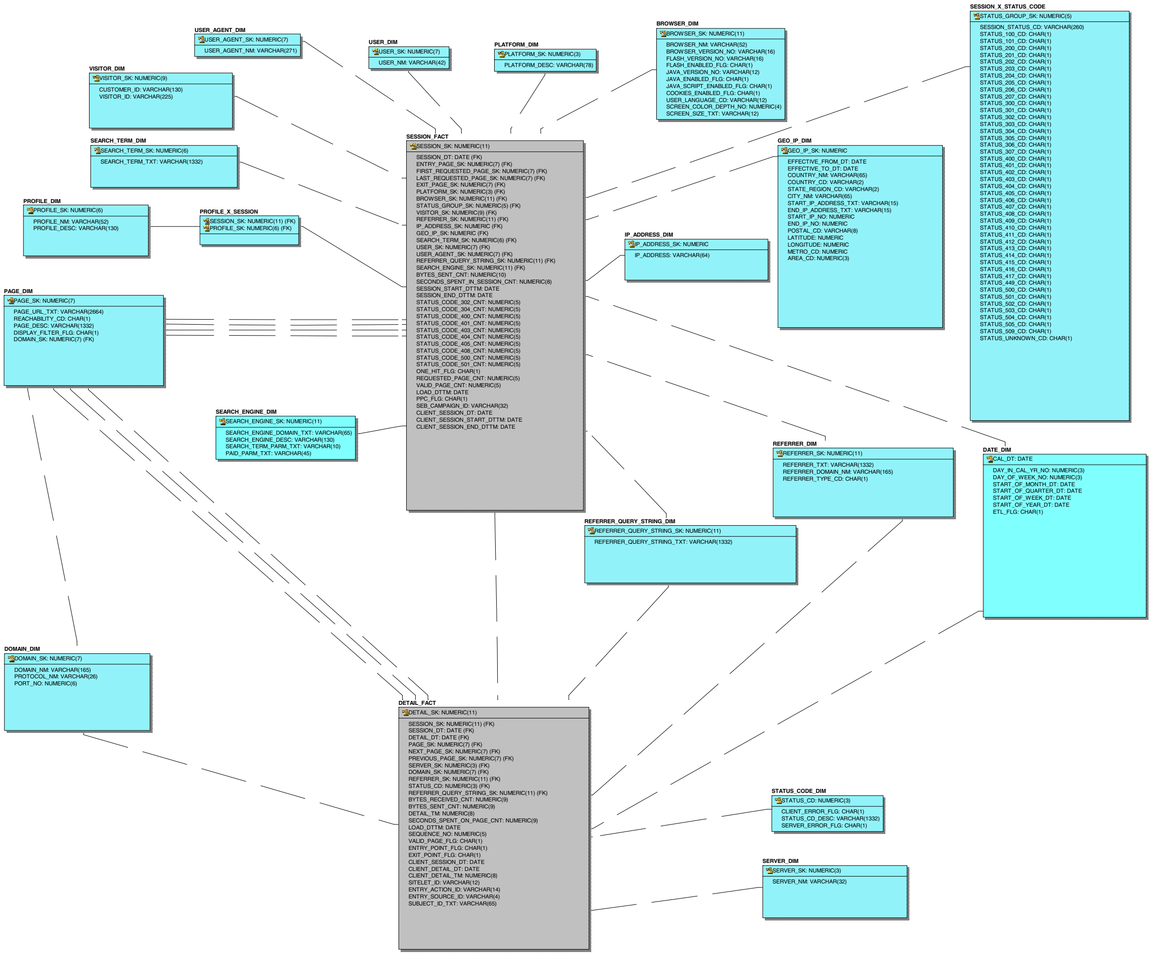
APPENDIX

1

SAS Web Analytics Web Mart Aggregates

Web Mart Data Model

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


USER_AGENT_DIM
USER_AGENT_SK: NUMERIC(7)
USER_AGENT_NM: VARCHAR(271)
USER_DIM
USER_SK: NUMERIC(7)
USER_NM: VARCHAR(42)
PLATFORM_DIM
PLATFORM_SK: NUMERIC(3)
PLATFORM_DESC: VARCHAR(78)
BROWSER_DIM
BROWSER_SK: NUMERIC(11)
BROWSER_NM: VARCHAR(52)
BROWSER_VERSION_NO: VARCHAR(16)
FLASH_VERSION_NO: VARCHAR(16)
FLASH_ENABLED_FLG: CHAR(1)
JAVA_VERSION_NO: VARCHAR(12)
JAVA_ENABLED_FLG: CHAR(1)
JAVA_SCRIPT_ENABLED_FLG: CHAR(1)
COOKIES_ENABLED_FLG: CHAR(1)
USER_LANGUAGE_CD: VARCHAR(12)
SCREEN_COLOR_DEPTH_NO: NUMERIC(4)
SCREEN_SIZE_TXT: VARCHAR(12)
SESSION_X_STATUS_CODE
STATUS_GROUP_SK: NUMERIC(5)
SESSION_STATUS_CD: VARCHAR(260)
STATUS_100_CD: CHAR(1)
STATUS_101_CD: CHAR(1)
STATUS_200_CD: CHAR(1)
STATUS_201_CD: CHAR(1)
STATUS_202_CD: CHAR(1)
STATUS_203_CD: CHAR(1)
STATUS_204_CD: CHAR(1)
STATUS_205_CD: CHAR(1)
STATUS_206_CD: CHAR(1)
STATUS_207_CD: CHAR(1)
STATUS_300_CD: CHAR(1)
STATUS_301_CD: CHAR(1)
STATUS_302_CD: CHAR(1)
STATUS_303_CD: CHAR(1)
STATUS_304_CD: CHAR(1)
STATUS_305_CD: CHAR(1)
STATUS_306_CD: CHAR(1)
STATUS_307_CD: CHAR(1)
STATUS_400_CD: CHAR(1)
STATUS_401_CD: CHAR(1)
STATUS_402_CD: CHAR(1)
STATUS_403_CD: CHAR(1)
STATUS_404_CD: CHAR(1)
STATUS_405_CD: CHAR(1)
STATUS_406_CD: CHAR(1)
STATUS_407_CD: CHAR(1)
STATUS_408_CD: CHAR(1)
STATUS_409_CD: CHAR(1)
STATUS_410_CD: CHAR(1)
STATUS_411_CD: CHAR(1)
STATUS_412_CD: CHAR(1)
STATUS_413_CD: CHAR(1)
STATUS_414_CD: CHAR(1)
STATUS_415_CD: CHAR(1)
STATUS_416_CD: CHAR(1)
STATUS_417_CD: CHAR(1)
STATUS_449_CD: CHAR(1)
STATUS_500_CD: CHAR(1)
STATUS_501_CD: CHAR(1)
STATUS_502_CD: CHAR(1)
STATUS_503_CD: CHAR(1)
STATUS_504_CD: CHAR(1)
STATUS_505_CD: CHAR(1)
STATUS_509_CD: CHAR(1)
STATUS_UNKNOWN_CD: CHAR(1)
VISITOR_DIM
VISITOR_SK: NUMERIC(9)
CUSTOMER_ID: VARCHAR(130)
VISITOR_ID: VARCHAR(225)
SEARCH_TERM_DIM
SEARCH_TERM_SK: NUMERIC(6)
SEARCH_TERM_TXT: VARCHAR(1332)
SESSION_FACT
SESSION_SK: NUMERIC(11)
SESSION_DT: DATE (FK)
ENTRY_PAGE_SK: NUMERIC(7) (FK)
FIRST_REQUESTED_PAGE_SK: NUMERIC(7) (FK)
LAST_REQUESTED_PAGE_SK: NUMERIC(7) (FK)
EXIT_PAGE_SK: NUMERIC(7) (FK)
PLATFORM_SK: NUMERIC(3) (FK)
BROWSER_SK: NUMERIC(11) (FK)
STATUS_GROUP_SK: NUMERIC(5) (FK)
VISITOR_SK: NUMERIC(9) (FK)
REFERRER_SK: NUMERIC(11) (FK)
IP_ADDRESS_SK: NUMERIC (FK)
GEO_IP_SK: NUMERIC (FK)
SEARCH_TERM_SK: NUMERIC(6) (FK)
USER_SK: NUMERIC(7) (FK)
USER_AGENT_SK: NUMERIC(7) (FK)
REFERRER_QUERY_STRING_SK: NUMERIC(11) (FK)
SEARCH_ENGINE_SK: NUMERIC(11) (FK)
BYTES_SENT_CNT: NUMERIC(10)
SECONDS_SPENT_IN_SESSION_CNT: NUMERIC(8)
SESSION_START_DTTM: DATE
SESSION_END_DTTM: DATE
STATUS_CODE_302_CNT: NUMERIC(5)
STATUS_CODE_304_CNT: NUMERIC(5)
STATUS_CODE_400_CNT: NUMERIC(5)
STATUS_CODE_401_CNT: NUMERIC(5)
STATUS_CODE_403_CNT: NUMERIC(5)
STATUS_CODE_404_CNT: NUMERIC(5)
STATUS_CODE_405_CNT: NUMERIC(5)
STATUS_CODE_408_CNT: NUMERIC(5)
STATUS_CODE_500_CNT: NUMERIC(5)
STATUS_CODE_501_CNT: NUMERIC(5)
ONE_HIT_FLG: CHAR(1)
REQUESTED_PAGE_CNT: NUMERIC(5)
VALID_PAGE_CNT: NUMERIC(5)
LOAD_DTTM: DATE
PPC_FLG: CHAR(1)
SEB_CAMPAIGN_ID: VARCHAR(32)
CLIENT_SESSION_DT: DATE
CLIENT_SESSION_START_DTTM: DATE
CLIENT_SESSION_END_DTTM: DATE
GEO_IP_DIM
GEO_IP_SK: NUMERIC
EFFECTIVE_FROM_DT: DATE
EFFECTIVE_TO_DT: DATE
COUNTRY_NM: VARCHAR(65)
COUNTRY_CD: VARCHAR(2)
STATE_REGION_CD: VARCHAR(2)
CITY_NM: VARCHAR(65)
START_IP_ADDRESS_TXT: VARCHAR(15)
END_IP_ADDRESS_TXT: VARCHAR(15)
START_IP_NO: NUMERIC
END_IP_NO: NUMERIC
POSTAL_CD: VARCHAR(8)
LATITUDE: NUMERIC
LONGITUDE: NUMERIC
METRO_CD: NUMERIC
AREA_CD: NUMERIC(3)
PROFILE_DIM
PROFILE_SK: NUMERIC(6)
PROFILE_NM: VARCHAR(52)
PROFILE_DESC: VARCHAR(130)
PROFILE_X_SESSION
SESSION_SK: NUMERIC(11) (FK)
PROFILE_SK: NUMERIC(6) (FK)
IP_ADDRESS_DIM
IP_ADDRESS_SK: NUMERIC
IP_ADDRESS: VARCHAR(64)
PAGE_DIM
PAGE_SK: NUMERIC(7)
PAGE_URL_TXT: VARCHAR(2664)
REACHABILITY_CD: CHAR(1)
PAGE_DESC: VARCHAR(1332)
DISPLAY_FILTER_FLG: CHAR(1)
DOMAIN_SK: NUMERIC(7) (FK)
SEARCH_ENGINE_DIM
SEARCH_ENGINE_SK: NUMERIC(11)
SEARCH_ENGINE_DOMAIN_TXT: VARCHAR(65)
SEARCH_ENGINE_DESC: VARCHAR(130)
SEARCH_TERM_PARM_TXT: VARCHAR(10)
PAID_PARM_TXT: VARCHAR(45)
REFERRER_DIM
REFERRER_SK: NUMERIC(11)
REFERRER_TXT: VARCHAR(1332)
REFERRER_DOMAIN_NM: VARCHAR(165)
REFERRER_TYPE_CD: CHAR(1)
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)
REFERRER_QUERY_STRING_DIM
REFERRER_QUERY_STRING_SK: NUMERIC(11)
REFERRER_QUERY_STRING_TXT: VARCHAR(1332)
DOMAIN_DIM
DOMAIN_SK: NUMERIC(7)
DOMAIN_NM: VARCHAR(165)
PROTOCOL_NM: VARCHAR(26)
PORT_NO: NUMERIC(6)
DETAIL_FACT
DETAIL_SK: NUMERIC(11)
SESSION_SK: NUMERIC(11) (FK)
SESSION_DT: DATE (FK)
DETAIL_DT: DATE (FK)
PAGE_SK: NUMERIC(7) (FK)
NEXT_PAGE_SK: NUMERIC(7) (FK)
PREVIOUS_PAGE_SK: NUMERIC(7) (FK)
SERVER_SK: NUMERIC(3) (FK)
DOMAIN_SK: NUMERIC(7) (FK)
REFERRER_SK: NUMERIC(11) (FK)
STATUS_CD: NUMERIC(3) (FK)
REFERRER_QUERY_STRING_SK: NUMERIC(11) (FK)
BYTES_RECEIVED_CNT: NUMERIC(9)
BYTES_SENT_CNT: NUMERIC(9)
DETAIL_TM: NUMERIC(8)
SECONDS_SPENT_ON_PAGE_CNT: NUMERIC(9)
LOAD_DTTM: DATE
SEQUENCE_NO: NUMERIC(5)
VALID_PAGE_FLG: CHAR(1)
ENTRY_POINT_FLG: CHAR(1)
EXIT_POINT_FLG: CHAR(1)
CLIENT_SESSION_DT: DATE
CLIENT_DETAIL_DT: DATE
CLIENT_DETAIL_TM: NUMERIC(8)
SITELET_ID: VARCHAR(12)
ENTRY_ACTION_ID: VARCHAR(14)
ENTRY_SOURCE_ID: VARCHAR(4)
SUBJECT_ID_TXT: VARCHAR(65)
STATUS_CODE_DIM
STATUS_CD: NUMERIC(3)
CLIENT_ERROR_FLG: CHAR(1)
STATUS_CD_DESC: VARCHAR(1332)
SERVER_ERROR_FLG: CHAR(1)
SERVER_DIM
SERVER_SK: NUMERIC(3)
SERVER_NM: VARCHAR(32)

Aggregates

Table A1.1: AGGR_DAILY_TOTAL_DAY

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		Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	NLNUM15.
page_count		Num	8	NLNUM15.	NLNUM15.
status_code_302_count		Num	8	BEST12.	12.
status_code_304_count		Num	8	BEST12.	12.
status_code_400_count		Num	8	BEST12.	12.
status_code_401_count		Num	8	BEST12.	12.
status_code_403_count		Num	8	BEST12.	12.
status_code_404_count		Num	8	BEST12.	12.
status_code_405_count		Num	8	BEST12.	12.
status_code_408_count		Num	8	BEST12.	12.
status_code_500_count		Num	8	BEST12.	12.
status_code_501_count		Num	8	BEST12.	12.
duration	Duration	Num	8	NLNUM15.	NLNUM15.
file_count	Hit Count	Num	8	NLNUM15.	NLNUM15.
one_hit_session_count	Bounces	Num	8	BEST12.	12.
total_bytes_sent	Total Kb Sent	Num	8	NLNUM15.	NLNUM15.
hour		Num	8	BEST12.	12.
day_of_week		Num	8		
daily_sessions		Num	8	BEST12.	12.

Table A1.2: AGGR_DAILY_TOTAL_MONTH

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8.	8.
repeat_visitors		Num	8	8.	8.
page_view_med_session_cnt		Num	8	8.	8.
page_view_low_session_cnt		Num	8	8.	8.
page_view_high_session_cnt		Num	8	8.	8.
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
status_code_302_count		Num	8	BEST12.	12.
status_code_304_count		Num	8	BEST12.	12.
status_code_400_count		Num	8	BEST12.	12.
status_code_401_count		Num	8	BEST12.	12.
status_code_403_count		Num	8	BEST12.	12.
status_code_404_count		Num	8	BEST12.	12.
status_code_405_count		Num	8	BEST12.	12.
status_code_408_count		Num	8	BEST12.	12.
status_code_500_count		Num	8	BEST12.	12.
status_code_501_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
file_count		Num	8	BEST12.	12.
one_hit_session_count		Num	8	BEST12.	12.
total_bytes_sent		Num	8	BEST12.	12.
hour		Num	8	BEST12.	12.
daily_sessions		Num	8	BEST12.	12.

Table A1.3: AGGR_DAILY_TOTAL_QUARTER

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8.	8.
repeat_visitors		Num	8	8.	8.
page_view_med_session_cnt		Num	8	8.	8.
page_view_low_session_cnt		Num	8	8.	8.
page_view_high_session_cnt		Num	8	8.	8.
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
status_code_302_count		Num	8	BEST12.	12.
status_code_304_count		Num	8	BEST12.	12.
status_code_400_count		Num	8	BEST12.	12.
status_code_401_count		Num	8	BEST12.	12.
status_code_403_count		Num	8	BEST12.	12.
status_code_404_count		Num	8	BEST12.	12.
status_code_405_count		Num	8	BEST12.	12.
status_code_408_count		Num	8	BEST12.	12.
status_code_500_count		Num	8	BEST12.	12.
status_code_501_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
file_count		Num	8	BEST12.	12.
one_hit_session_count		Num	8	BEST12.	12.
total_bytes_sent		Num	8	BEST12.	12.
hour		Num	8	BEST12.	12.
daily_sessions		Num	8	BEST12.	12.

Table A1.4: AGGR_DAILY_TOTAL_WEEK

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8.	8.
repeat_visitors		Num	8	8.	8.
page_view_med_session_cnt		Num	8	8.	8.
page_view_low_session_cnt		Num	8	8.	8.
page_view_high_session_cnt		Num	8	8.	8.
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
status_code_302_count		Num	8	BEST12.	12.
status_code_304_count		Num	8	BEST12.	12.
status_code_400_count		Num	8	BEST12.	12.
status_code_401_count		Num	8	BEST12.	12.
status_code_403_count		Num	8	BEST12.	12.
status_code_404_count		Num	8	BEST12.	12.
status_code_405_count		Num	8	BEST12.	12.
status_code_408_count		Num	8	BEST12.	12.
status_code_500_count		Num	8	BEST12.	12.
status_code_501_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
file_count		Num	8	BEST12.	12.
one_hit_session_count		Num	8	BEST12.	12.
total_bytes_sent		Num	8	BEST12.	12.
hour		Num	8	BEST12.	12.
daily_sessions		Num	8	BEST12.	12.

Table A1.5: AGGR_DAILY_TOTAL_YEAR

Field	Label	Type	Length	Format	Informat
unique_visitors		Num	8	8.	8.
repeat_visitors		Num	8	8.	8.
page_view_med_session_cnt		Num	8	8.	8.
page_view_low_session_cnt		Num	8	8.	8.
page_view_high_session_cnt		Num	8	8.	8.
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
status_code_302_count		Num	8	BEST12.	12.
status_code_304_count		Num	8	BEST12.	12.
status_code_400_count		Num	8	BEST12.	12.
status_code_401_count		Num	8	BEST12.	12.
status_code_403_count		Num	8	BEST12.	12.
status_code_404_count		Num	8	BEST12.	12.
status_code_405_count		Num	8	BEST12.	12.
status_code_408_count		Num	8	BEST12.	12.
status_code_500_count		Num	8	BEST12.	12.
status_code_501_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
file_count		Num	8	BEST12.	12.
one_hit_session_count		Num	8	BEST12.	12.
total_bytes_sent		Num	8	BEST12.	12.
hour		Num	8	BEST12.	12.
daily_sessions		Num	8	BEST12.	12.

Table A1.6: AGGR_GEO

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE9.	DATE9.
page_cnt		Num	8		
session_cnt		Num	8		
country_nm		Char	50		
state_region_cd		Char	4		
city_nm		Char	50		
start_of_week_dt		Num	8	DATE9.	DATE9.
start_of_month_dt		Num	8	DATE9.	DATE9.
start_of_quarter_dt		Num	8	DATE9.	DATE9.
start_of_year_dt		Num	8	DATE9.	DATE9.

Table A1.7: AGGR_GEO_FULL

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE9.	DATE9.
page_cnt		Num	8		
session_cnt		Num	8		
country_nm		Char	50		
state_region_cd		Char	4		
city_nm		Char	50		
start_of_week_dt		Num	8	DATE9.	DATE9.
start_of_month_dt		Num	8	DATE9.	DATE9.
start_of_quarter_dt		Num	8	DATE9.	DATE9.
start_of_year_dt		Num	8	DATE9.	DATE9.

Table A1.8: AGGR_HOURLY_STATUS_DAILY

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	15.
page_count	Page Count	Num	8	NLNUM15.	15.
file_count	Hit Count	Num	8	NLNUM15.	15.
hour		Num	8	BEST12.	12.
status_cd		Num	3		
daily_sessions		Num	8	BEST12.	12.

Table A1.9: AGGR_HOURLY_STATUS_MONTH

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
file_count		Num	8	BEST12.	12.
hour		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
daily_sessions		Num	8	BEST12.	12.

Table A1.10: AGGR_HOURLY_STATUS_QUARTER

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
page_count		Num	8		
file_count		Num	8		
hour		Num	8		
status_cd		Num	3		

Table A1.11: AGGR_HOURLY_STATUS_WEEK

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
page_count		Num	8		
file_count		Num	8		
hour		Num	8		
status_cd		Num	3		

Table A1.12: AGGR_HOURLY_STATUS_YEAR

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
page_count		Num	8		
file_count		Num	8		
hour		Num	8		
status_cd		Num	3		

Table A1.13: AGGR_PAGE_DAY

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	15.
page_count		Num	8	NLNUM15.	15.
entry_page_count		Num	8	BEST12.	12.
exit_page_count		Num	8	BEST12.	12.
entry_total		Num	8	BEST12.	12.
exit_total		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
page_total		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
first_requested_page_flg		Char	1	$1.	$1.
valid_page_flg		Char	1	$1.	$1.
page_sk		Num	7	BEST12.	12.
prosp_bounce		Num	8	NLNUM15.	15.

Table A1.14: AGGR_PAGE_MONTH

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
entry_page_count		Num	8	BEST12.	12.
exit_page_count		Num	8	BEST12.	12.
entry_total		Num	8	BEST12.	12.
exit_total		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
page_total		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
first_requested_page_flg		Char	1	$1.	$1.
valid_page_flg		Char	1	$1.	$1.
page_sk		Num	7	BEST12.	12.
prosp_bounce		Num	8	NLNUM12.	12.
one_hit_session_count		Num	8	BEST12.	12.

Table A1.15: AGGR_PAGE_QUARTER

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
entry_page_count		Num	8	BEST12.	12.
exit_page_count		Num	8	BEST12.	12.
entry_total		Num	8	BEST12.	12.
exit_total		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
page_total		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
first_requested_page_flg		Char	1	$1.	$1.
valid_page_flg		Char	1	$1.	$1.
page_sk		Num	7	BEST12.	12.
prosp_bounce		Num	8	NLNUM12.	12.
one_hit_session_count		Num	8	BEST12.	12.

Table A1.16: AGGR_PAGE_WEEK

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
entry_page_count		Num	8	BEST12.	12.
exit_page_count		Num	8	BEST12.	12.
entry_total		Num	8	BEST12.	12.
exit_total		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
page_total		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
first_requested_page_flg		Char	1	$1.	$1.
valid_page_flg		Char	1	$1.	$1.
page_sk		Num	7	BEST12.	12.
prosp_bounce		Num	8	NLNUM12.	12.
one_hit_session_count		Num	8	BEST12.	12.

Table A1.17: AGGR_PAGE_YEAR

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
entry_page_count		Num	8	BEST12.	12.
exit_page_count		Num	8	BEST12.	12.
entry_total		Num	8	BEST12.	12.
exit_total		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
page_total		Num	8	BEST12.	12.
status_cd		Num	3	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
first_requested_page_flg		Char	1	$1.	$1.
valid_page_flg		Char	1	$1.	$1.
page_sk		Num	7	BEST12.	12.
prosp_bounce		Num	8	NLNUM12.	12.
one_hit_session_count		Num	8	BEST12.	12.

Table A1.18: AGGR_PLATFORM_DAY

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	15.
page_count	Page Count	Num	8	NLNUM15.	15.
session_sum		Num	8	BEST12.	12.
page_sum		Num	8	BEST12.	12.
browser_nm		Char	52	$52.	$52.
browser_version_no		Char	16	$16.	$16.
platform_desc		Char	78	&78.	$78.

Table A1.19: AGGR_PLATFORM_MONTH

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
session_sum		Num	8	BEST12.	12.
page_sum		Num	8	BEST12.	12.
browser_nm		Char	52	$52.	$52.
browser_version_no		Char	16	$16.	$16.
platform_desc		Char	78	$78.	$78.

Table A1.20: AGGR_PLATFORM_QUARTER

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
session_sum		Num	8	BEST12.	12.
page_sum		Num	8	BEST12.	12.
browser_nm		Char	52	$52.	$52.
browser_version_no		Char	16	$16.	$16.
platform_desc		Char	78	$78.	$78.

Table A1.21: AGGR_PLATFORM_WEEK

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
session_sum		Num	8	BEST12.	12.
page_sum		Num	8	BEST12.	12.
browser_nm		Char	52	$52.	$52.
browser_version_no		Char	16	$16.	$16.
platform_desc		Char	78	$78.	$78.

Table A1.22: AGGR_PLATFORM_YEAR

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DAT7E.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
session_sum		Num	8	BEST12.	12.
page_sum		Num	8	BEST12.	12.
browser_nm		Char	52	$52.	$52.
browser_version_no		Char	16	$16.	$16.
platform_desc		Char	78	$78.	$78.

Table A1.23: AGGR_RECENCY

Field	Label	Type	Length	Format	Informat
visit_count		Num	8		
duration		Num	8	5.2	
WeeksApart		Num	8		
visit_pct		Num	8	PERCENT5.2	

Table A1.24: AGGR_REF_TERM_DAY

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	15.
session_total		Num	8	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
search_term_sk		Num	6	BEST12.	12.
search_engine_sk		Num	8	BEST12.	12.

Table A1.25: AGGR_REF_TERM_MONTH

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
session_total		Num	8	BEST12.	12.
referrer_sk		Num	8	BEST12.	12.
search_term_sk		Num	6	BEST12.	12.

Table A1.26: AGGR_REFERRER_SEARCH_TERM_QTR

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
session_total		Num	8		
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.27: AGGR_REFERRER_SEARCH_TERM_WEEK

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
session_total		Num	8		
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.28: AGGR_REFERRER_SEARCH_TERM_YEAR

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE.	DATE9.
session_count		Num	8		
session_total		Num	8		
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.29: AGGR_REF_ANNUALLY

Field	Label	Type	Length	Format	Informat
session_count		Num	8		
session_total		Num	8		
session_dt		Num	8	DATE9.	DATE9.
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.30: AGGR_REF_DAILY

Field	Label	Type	Length	Format	Informat
session_count	Session Count	Num	8	NLNUM15.	
session_total		Num	8		
session_dt		Num	8	DATE9.	DATE9.
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.31: AGGR_REF_MONTHLY

Field	Label	Type	Length	Format	Informat
session_count		Num	8		
session_total		Num	8		
session_dt		Num	8	DATE9.	DATE9.
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.32: AGGR_REF_QUARTERLY

Field	Label	Type	Length	Format	Informat
session_count		Num	8		
session_total		Num	8		
session_dt		Num	8	DATE9.	DATE9.
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.33: AGGR_REF_WEEKLY

Field	Label	Type	Length	Format	Informat
session_count		Num	8		
session_total		Num	8		
session_dt		Num	8	DATE9.	DATE9.
referrer_sk		Num	8		
search_term_sk		Num	6		

Table A1.34: AGGR_VISITOR_ANNUALLY

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
page_view_high_session_cnt		Num	8	BEST12.	12.
page_view_low_session_cnt		Num	8	BEST12.	12.
page_view_med_session_cnt		Num	8	BEST12.	12.
visitor_sk		Num	8	BEST12.	12.
start_dttm		Num	8	DATETIME20.	20.
end_dttm		Num	8	DATETIME20.	20.

Table A1.35: AGGR_VISITOR_DAILY

Field	Label	Type	Length	Format	Informat
session_dt	day	Num	8	DATE7.	DATE9.
session_count	Session Count	Num	8	NLNUM15.	15.
page_count		Num	8	NLNUM15.	15.
duration	Duration	Num	8	NLNUM15.	15.
page_view_high_session_cnt	High (> 8) Page Session Count	Num	8	NLNUM15.	15.
page_view_low_session_cnt	Low (<= 4) Page Session Count	Num	8	NLNUM15.	15.
page_view_med_session_cnt	Medium (&med_label) Page Session Count	Num	8	NLNUM15.	15.
visitor_sk		Num	8	BEST12.	12.
start_dttm		Num	8	DATETIME20.	NLDATM21.
end_dttm	start_dttm	Num	8	DATETIME20.	NLDATM21.

Table A1.36: AGGR_VISITOR_MONTHLY

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
page_view_high_session_cnt		Num	8	BEST12.	12.
page_view_low_session_cnt		Num	8	BEST12.	12.
page_view_med_session_cnt		Num	8	BEST12.	12.
visitor_sk		Num	8	BEST12.	12.
start_dttm		Num	8	DATETIME20.	20.
end_dttm		Num	8	DATETIME20.	20.

Table A1.37: AGGR_VISITOR_QUARTERLY

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
page_view_high_session_cnt		Num	8	BEST12.	12.
page_view_low_session_cnt		Num	8	BEST12.	12.
page_view_med_session_cnt		Num	8	BEST12.	12.
visitor_sk		Num	8	BEST12.	12.
start_dttm	start_dttm	Num	8	DATETIME20.	20.
end_dttm	end_dttm	Num	8	DATETIME20.	20.

Table A1.38: AGGR_VISITOR_WEEKLY

Field	Label	Type	Length	Format	Informat
session_dt		Num	8	DATE7.	DATE9.
session_count		Num	8	BEST12.	12.
page_count		Num	8	BEST12.	12.
duration		Num	8	BEST12.	12.
page_view_high_session_cnt		Num	8	BEST12.	12.
page_view_low_session_cnt		Num	8	BEST12.	12.
page_view_med_session_cnt		Num	8	BEST12.	12.
visitor_sk		Num	8	BEST12.	12.
start_dttm		Num	8	DATETIME20.	20.
end_dttm		Num	8	DATETIME20.	20.

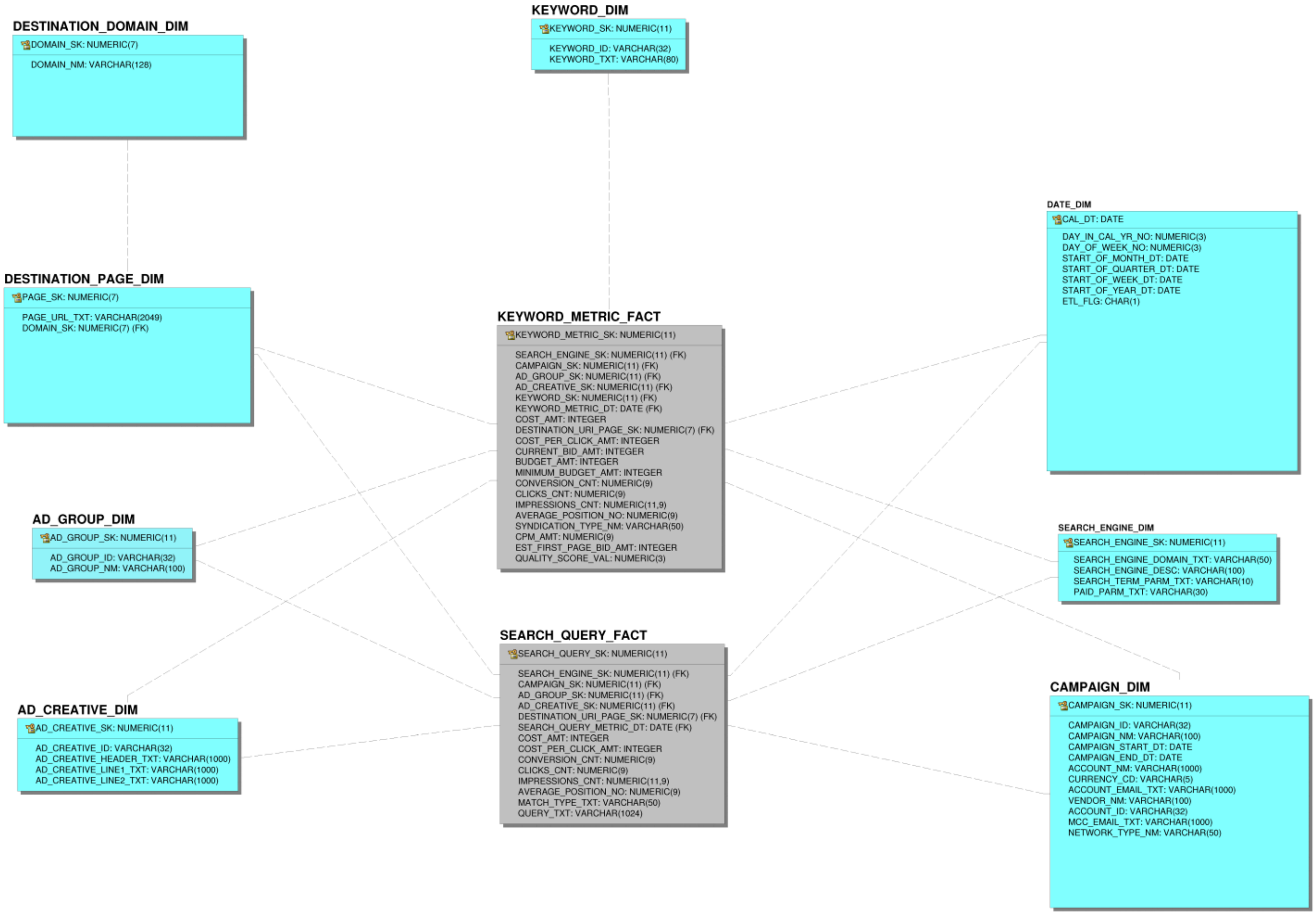
APPENDIX

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)
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)
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)

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	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_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.G OOGLE_SQ to dimensional table. Need to join dest_domain with Destination_Do main_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		
SEBDWORK. GOOGLE_SQ	Clicks	Num	8		Clicks_cnt	Num	8		

Source Table	Source Field	Type	Length	Transformation Notes	Table Field	Type	Length	Key Type	Notes
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

Quick Start Steps for Creating a Web Mart

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 prime 0000 job.

4 Run all the remaining ETL jobs for the new Web mart.

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 1100_ 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.

If while running the site initialization program you receive an “ERROR: Insufficient memory available for java” message, examine the JREOPTIONS parameter in the same SAS session context in which the site initialization was run. Be sure to use the same SAS program, arguments, and so on. Use the following SAS code:

```
proc options option=jreoptions;
run;
```

Check for the parameter -Xmx128m. If this argument is not present in the JREOPTIONS, then add it. It can be added to the appropriate sasv9.cfg file or entered on the command line. The value 128m is recommended, but on some platforms such as Sun Solaris for SPARC 64-bit, a higher value might be required (for example, 150m).

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 found in <*your-server-installation-location*>/config/Lev1/SASApp. For example, run <*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.

   You can use these plug-ins to help you determine the page referrer:

   - ❑ Firefox - HTTP Live Headers
   - ❑ Internet Explorer ieHTTPHeaders

3 For each search engine that will be added to the SEARCH_ENGINE_DIM table, gather the following information for these fields:

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*>/5.3.3 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.

Removing a Site

You can remove a Web site by running a macro to delete its metadata, or by deleting its metadata manually. To run the macro, enter the following code where `webmart_name` is name of the Web mart for which you are removing the associated metadata:

```
%weba_webmart_metadata_cleanup(webmart_name=)
```

To remove a Web site's metadata manually:

1. Open SAS Management Console and log on as the SAS Administrator or as a SAS Web Analytics Administrator.
2. Click the **Folders** tab.
3. Navigate to **`/System/Applications/SAS Web Analytics/Web Analytics 5.3.3/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.
4. 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
5. Exit SAS Management Console.

To complete the removal, delete the data mart that is associated with the site.

1. Navigate to the directory where the site's data mart exists.
2. 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.sas 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
```

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*>\5.3.3 Jobs\ETL Jobs\Webmart\weba_2500_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_2500_prep_insight_monitor job, and then run the job.

APPENDIX

5

How the Aggregate Jobs Work

The Aggregate Transformation

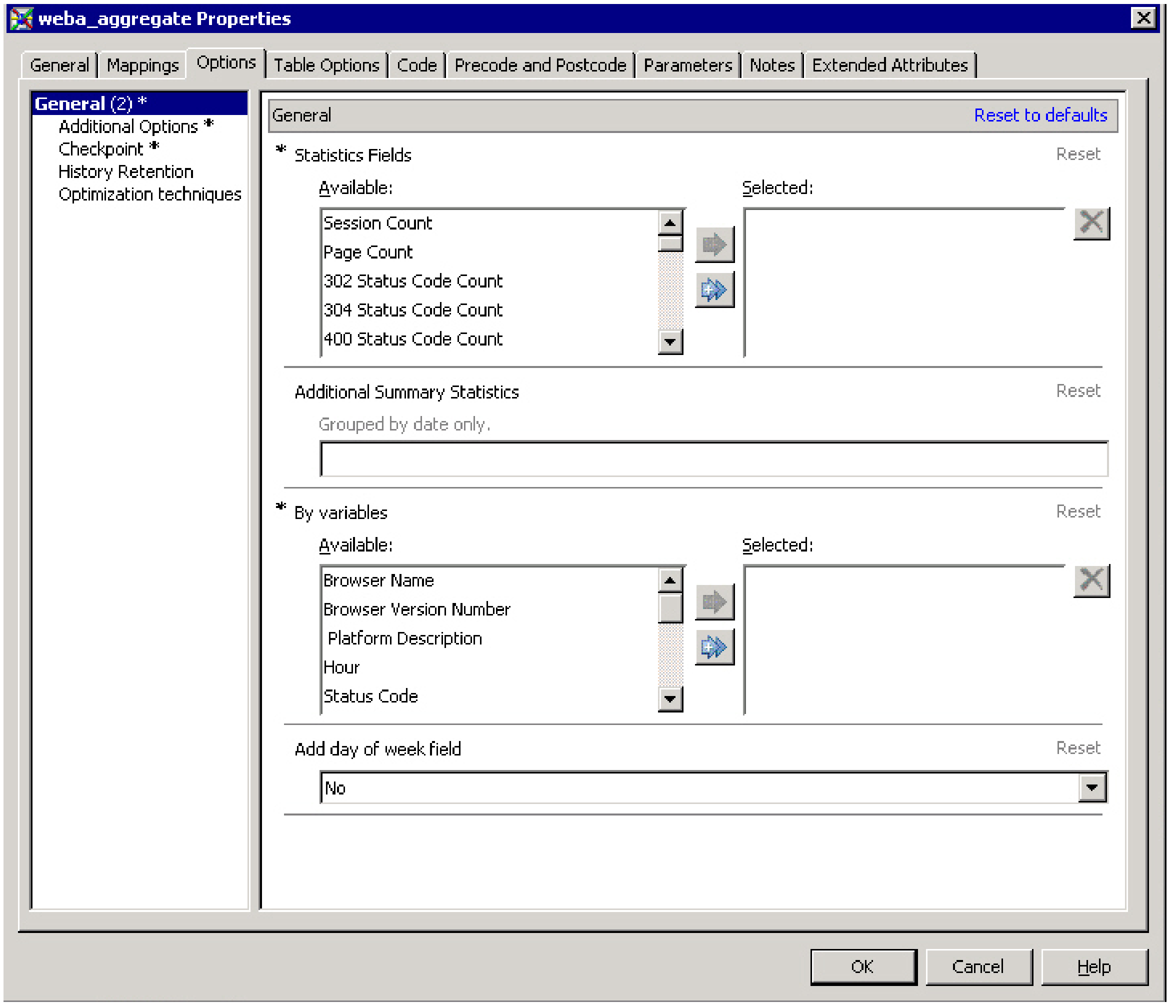
The transformation uses two input tables:

- Input table 1 contains an iteration of summarized data.
- Input table 2 references WAWORK.WEBMART to obtain 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 will not backload past history. Therefore, if the options settings change in the Aggregates object, then the user must ensure that the changes will not negatively affect the correctness of the data or incremental loading. For example, if an administrator changes a grouping element or adds or removes a metric after running aggregate loads for a period of time, the administrator must add the appropriate column to the upstream aggregate tables and account for past history.
- Input and output tables must contain the fields that are referenced in the transformation's selected columns in order to function properly.
- When adding output tables, delete the automatically included table loader process for each job. The transformation re-creates tables dynamically, and this extra code is not necessary and can compromise the accuracy of the process.
- Do not add a date index to the output tables or you will force SAS Data Integration Studio to create a date index, which already exists.
- Business rules are generated from the user's selections and entries on the **Options** tab of the transformation. To view the options, right-click on the 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:

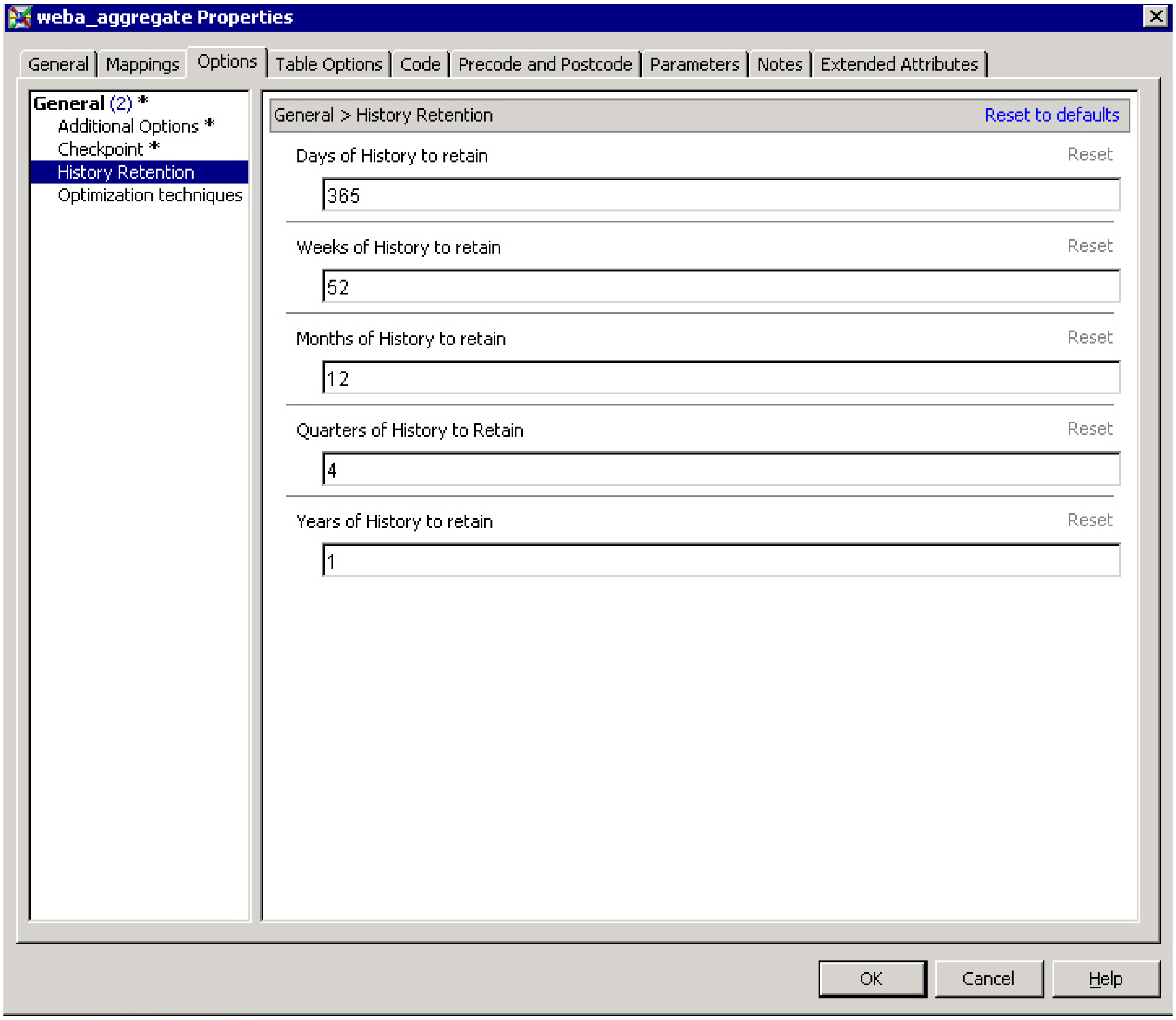
Browser Name
Browser Version Number
Platform Description
Hour
Status Code

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.



weba_aggregate Properties

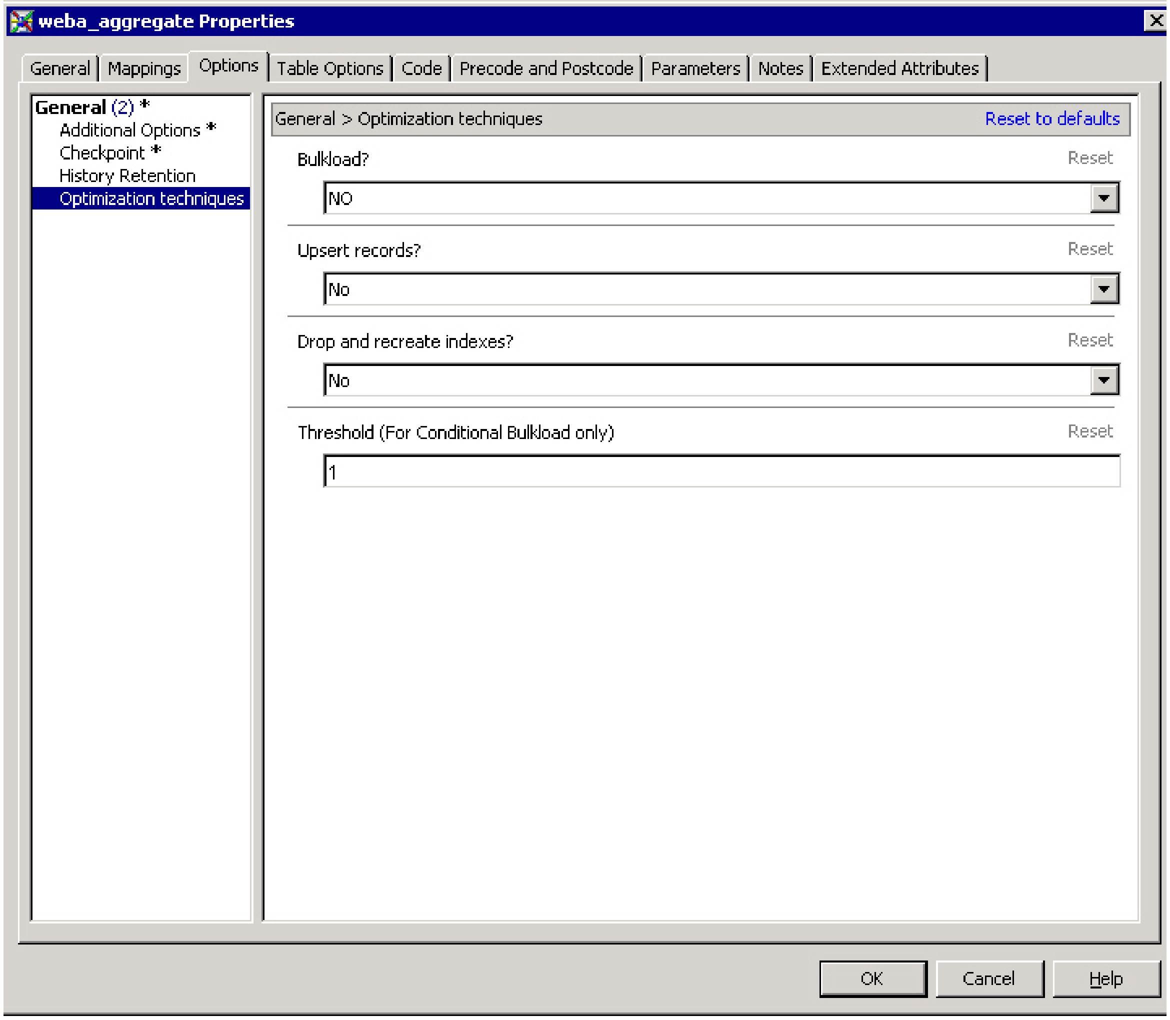
Days of History to retain: 365

Weeks of History to retain: 52

Months of History to retain: 12

Quarters of History to Retain: 4

Years of History to retain: 1



Bulkload? NO

Upsert records? No

Drop and recreate indexes? No

Threshold (For Conditional Bulkload only) 1

Regenerate Aggregates

Introduction

You can edit the transformation to reload aggregate tables with all of the history that is currently stored in the warehouse tables. Complete the steps for the Web mart that you are loading. The process varies depending on the type of data that is in the tables.

SAS Data

1. Within a Web mart, back up the 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 aggregate tables from the WaMart library.

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 2099 inclusively, and have _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.

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 Reschedule the ETL.

Oracle Data

1 Within a Web mart, back up the 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 aggregate tables from the WaMart library that you backed up in step 1.

The aggregate jobs contain the numbers 1500 through 2099 inclusively, and have _agg after the number. For each aggregate job to be reloaded, complete the following steps:

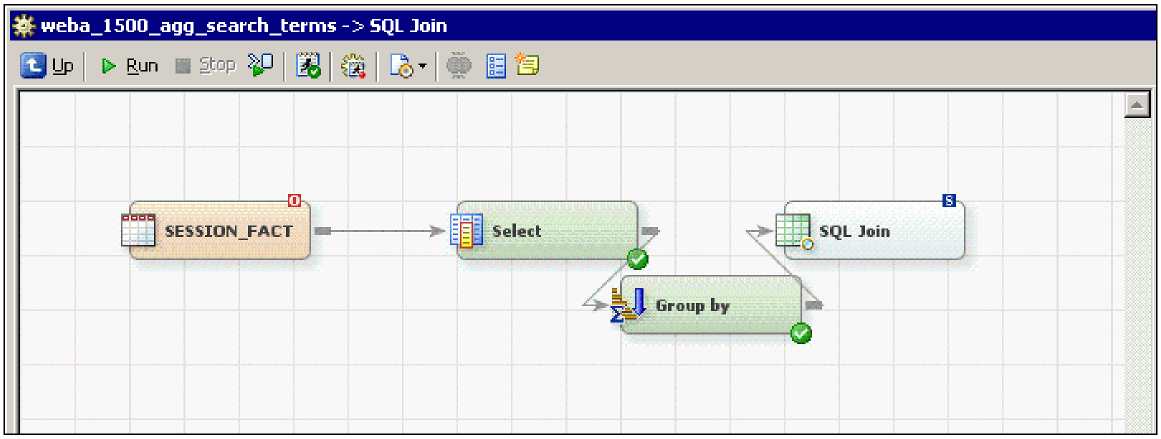
1 In SAS Data Integration Studio, navigate to the <*Web mart*>\\`5.3.3 Jobs\ETL Jobs` folder and open the aggregate job that is to be run.

2 Each aggregate has either a SESSION_FACT table, a DETAIL_FACT table, or both. Complete these steps on each table:

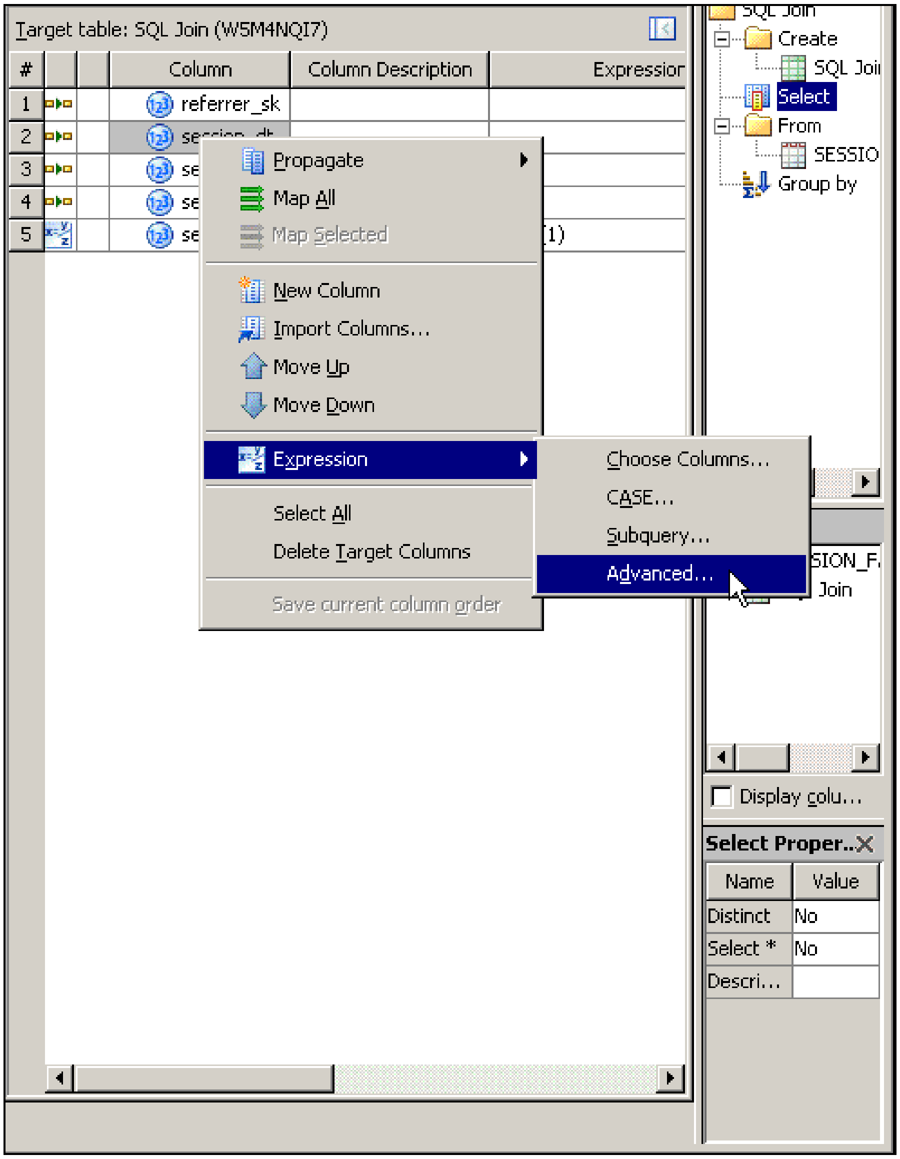
   a. Right-click the **SESSION_FACT** or **DETAIL_FACT** table and select **Properties**. On the **Physical Storage** tab next to the Library box, click the **...** button. Navigate to the <*Web mart*>\\`WaMart` folder and select the mart's WaMart library object. Click **OK** and close the Properties window. The red O in the upper right corner of the table object indicates an Oracle table.

   b. Right-click the **SQL Join** object that is connected to the table. Click **Open**.

   c. In the resulting diagram, double-click the **Select** object.



d. On the right side of the mapping, select the **session_dt** or **detail_fact** variable under the **Target Table** column listing.


Target table: SQL Join (W5M4NQI7)

#			Column	Column Description	Expression
1			referrer_sk		
2			session_dt		
3			se		
4			se		
5			se		(1)

Select Proper..

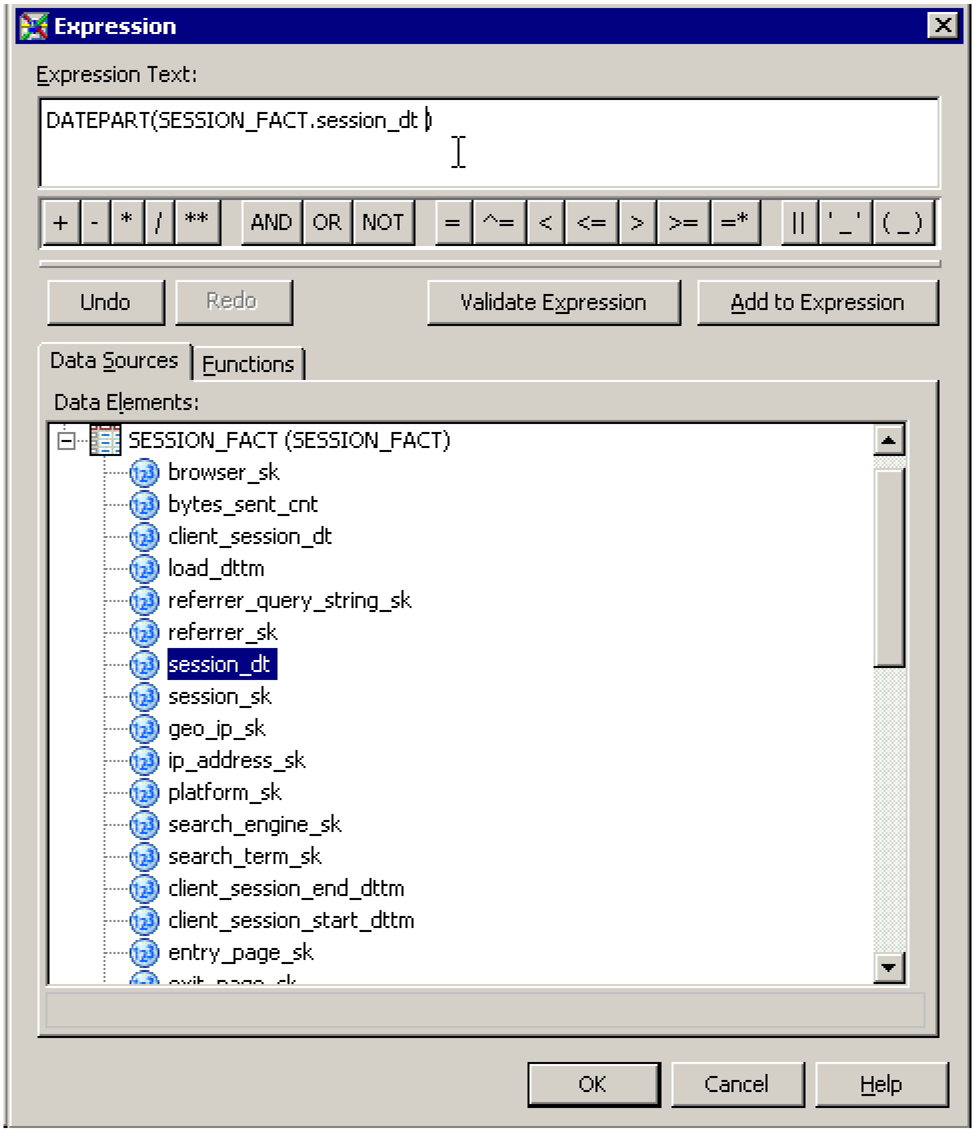
Name	Value
Distinct	No
Select *	No
Descri...	

e. Right-click the selected column and select **Expression** ▸ **Advanced**.

f. 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 double-click the **DATEPART** function. The function is entered into the **Expression Text** box.

-Click the **Data Sources** tab. Expand the **SESSION_FACT** or **DETAIL_FACT** table, and then select the **session_dt** or **detail_fact** column, respectively. The column is entered into the **Expression Text** box. Close the Expression Builder window.



Expression Text:

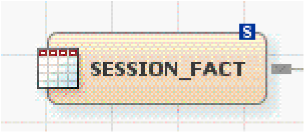
```
DATEPART(SESSION_FACT.session_dt )
```

-Click **Up** from the menu bar to return to the full diagram in the Job Editor window.

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.



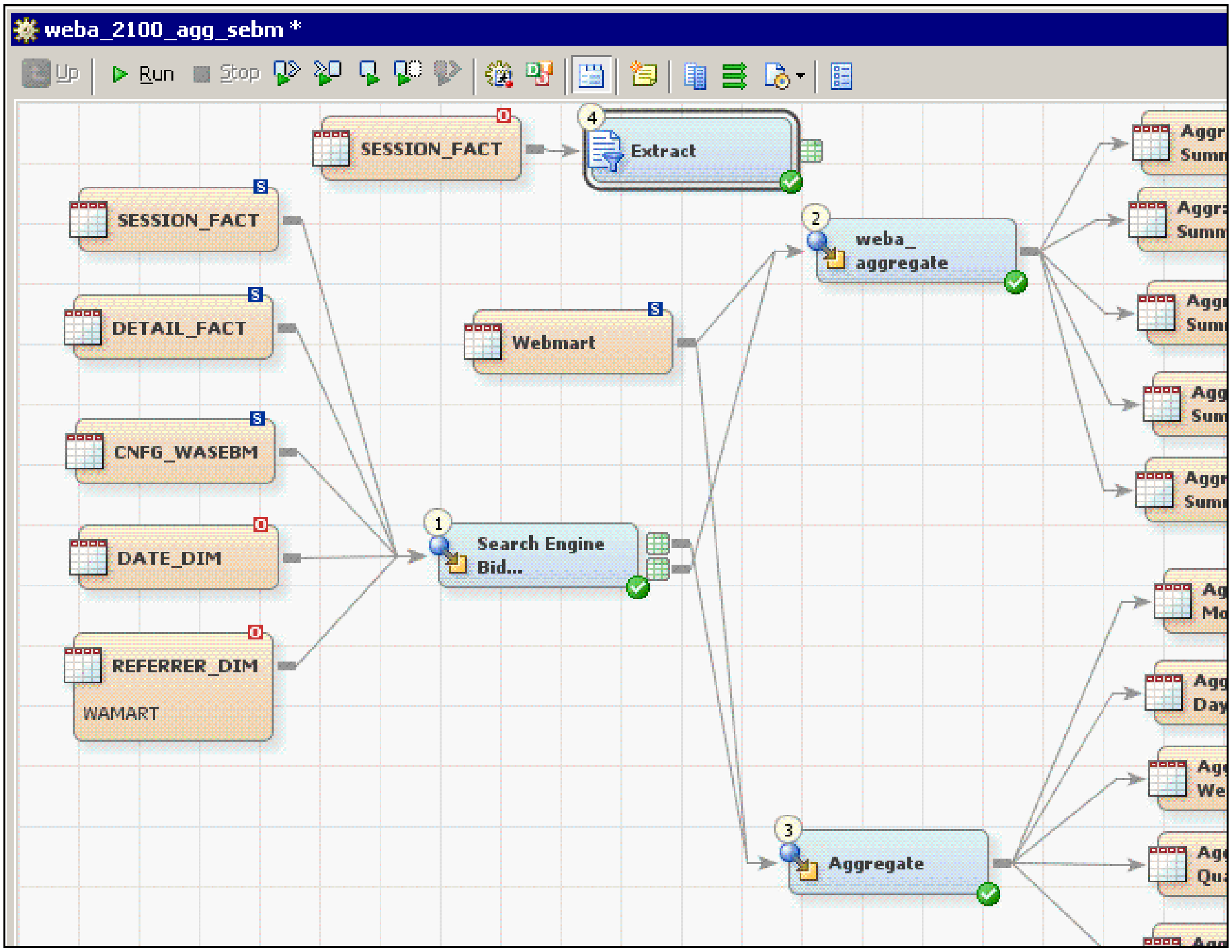
b. If you did not save the modified job, you can close the job without saving it and the case expression modification will not be saved. If you did save the job, then delete the expression. Right-click the **SQL Join** object that is connected to the table, and then click **Open**. In the diagram, double-click the **Select** object. On the right side of the mapping, select the **session_dt** or **detail_dt** variable under the **Target Table** column listing. Scroll to the right to view the **Expression** column, click on the expression, and then delete it. Click **Up** from the menu bar to return to the full diagram in the Job Editor window. Save the job and close it.

5 For the weba_2100_agg_sebm job and the weba_2150_seb_campaigns jobs, complete the following steps:

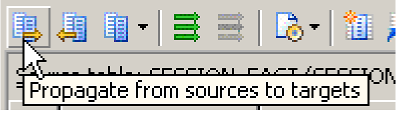
a. In SAS Data Integration Studio, navigate to the <*Web mart*> **`5.3.3 Jobs\ETL Jobs`** folder and open the aggregate job that is to be run.

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.


Propagate from sources to targets
Source table: SESSION_FACT (SESSION

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 box. 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**.

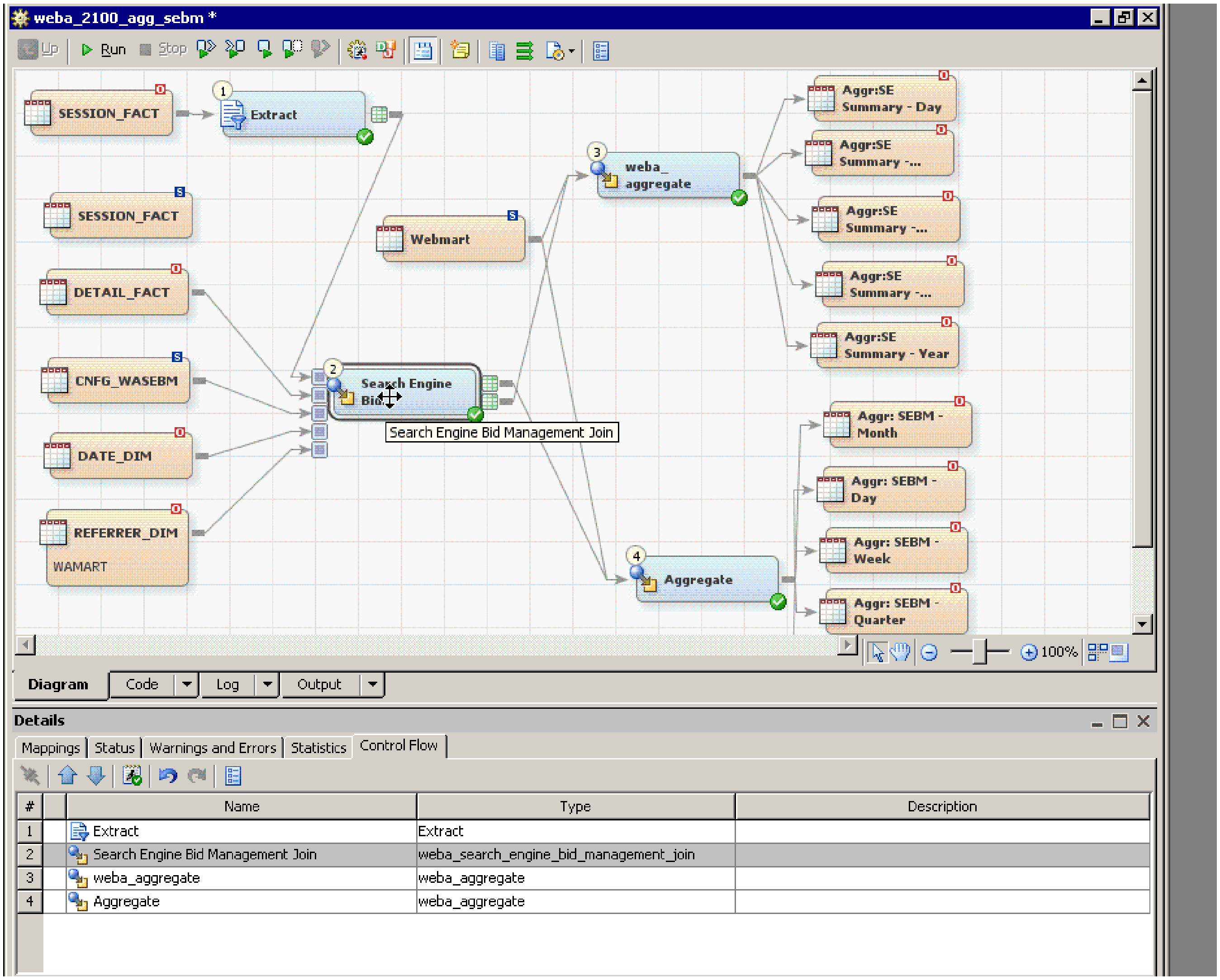
   Note: For the 2100_agg_scbm job, use the Search Engine Bid Management join.

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_scbm 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 successfully created, restore the original settings to the job 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.
- If you did not save the modified job, 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 the administrator to identify Class variables. Date is always used as a Class variable, so leaving this value blank will aggregate 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 along with the date field. Statistics are calculated (grouped) by the fields noted here along 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 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. Only daily grain records have a populated value for this field. Other tables (week, day, month, quarter, year) do not contain this field. However, if administrators have chosen a single-table option that causes all grains of history to be stored in a single table, then tables that have Interval set to Day will have populated values. All other intervals will have missing values for this field. The default setting is NO.

Days/Weeks/Months/Quarters/Years of History

specify 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 on the RDBMS table before appending, and will re-create them upon completion of the load. If the setting is NO, the indexes remain intact throughout the load process. The default setting is NO. This setting has no impact if the repository is not Oracle.

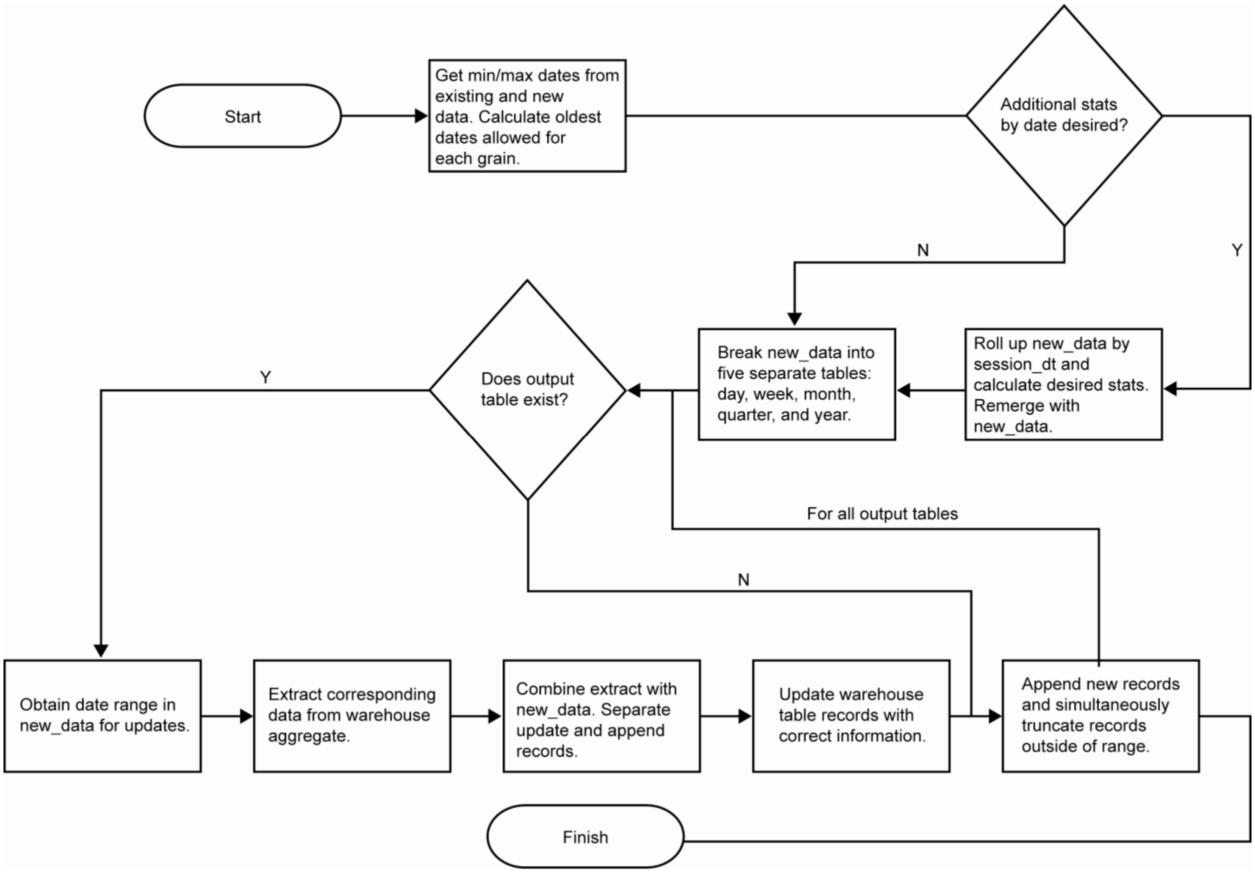
Oracle Upsert

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 bulkloaded. Loads that have less than the THRESHOLD value and that have been given a Conditional value will be inserted.

Figure 5.1: Aggregate Transformation Process


Start
Get min/max dates from existing and new data. Calculate oldest dates allowed for each grain.
Additional stats by date desired?
N
Y
Does output table exist?
Break new_data into five separate tables: day, week, month, quarter, and year.
Roll up new_data by session_dt and calculate desired stats. Remerge with new_data.
Y
For all output tables
N
Obtain date range in new_data for updates.
Extract corresponding data from warehouse aggregate.
Combine extract with new_data. Separate update and append records.
Update warehouse table records with correct information.
Append new records and simultaneously truncate records outside of range.
Finish

It is possible to 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.

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

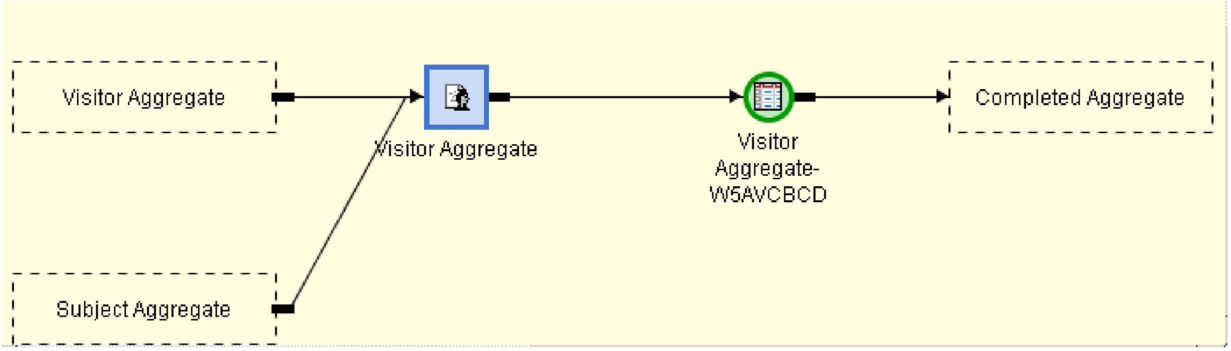
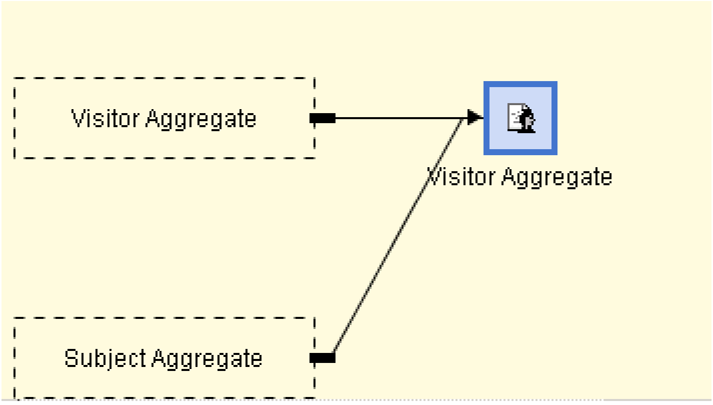


Figure 5.3: Visitor Aggregation Flow



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 due to 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 5.3.3

Prepare to Migrate

If you are upgrading to SAS Web Analytics 5.3.3 from 5.3 or 5.3.1, you need to 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 SAS Unicode server will be implemented in 5.3.3 and take appropriate action.

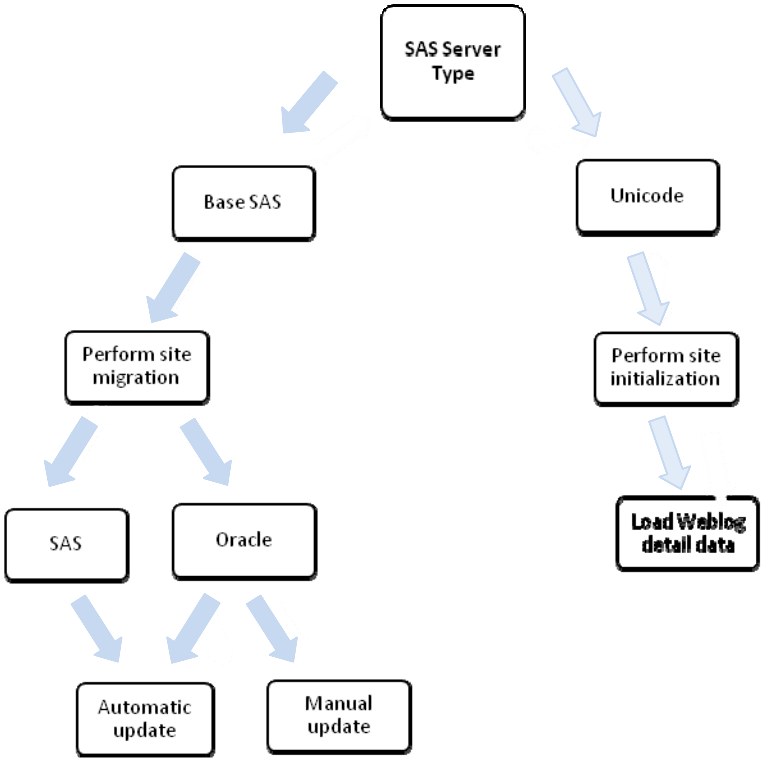
5 (Oracle only) Analyze your tablespace requirements. For more information, see “Oracle Considerations” on page 176.

How SAS Unicode Affects Your Migration Path

Your migration path is determined by whether SAS Unicode server will be implemented. It is recommended that you implement 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 SAS Unicode server is implemented, then you need to reinitialize and reload all existing Web marts. If 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 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. It is recommended that you 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 automatically by the migration job or whether the update should be manual.

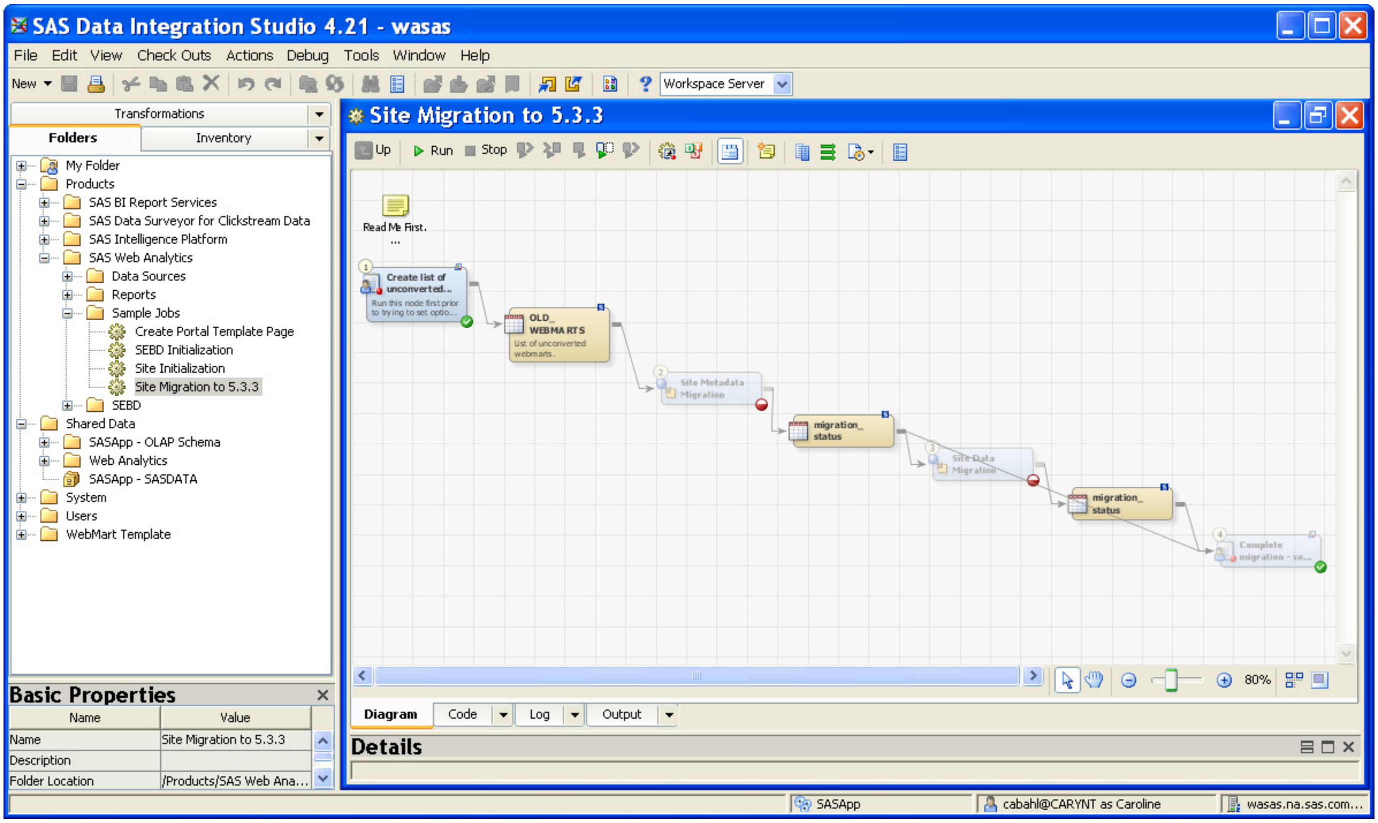
Site Migration Steps

1 Ensure that the last ETL scheduled for each site that is affected by the migration has completed successfully.

Note: Before you continue with the migration, make sure you have completed the steps in “Prepare to Migrate” on page 175.

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

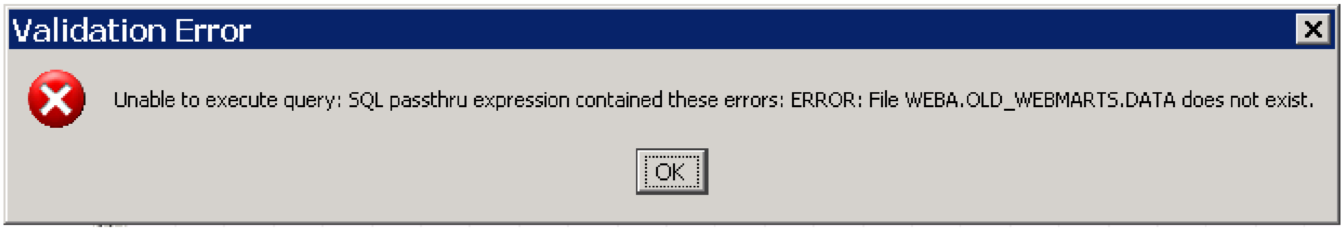
3 Open the **Site Migration to 5.3.3** job and read the Read Me First note.



Basic Properties

Name	Value
Name	Site Migration to 5.3.3
Description	
Folder Location	/Products/SAS Web Ana...

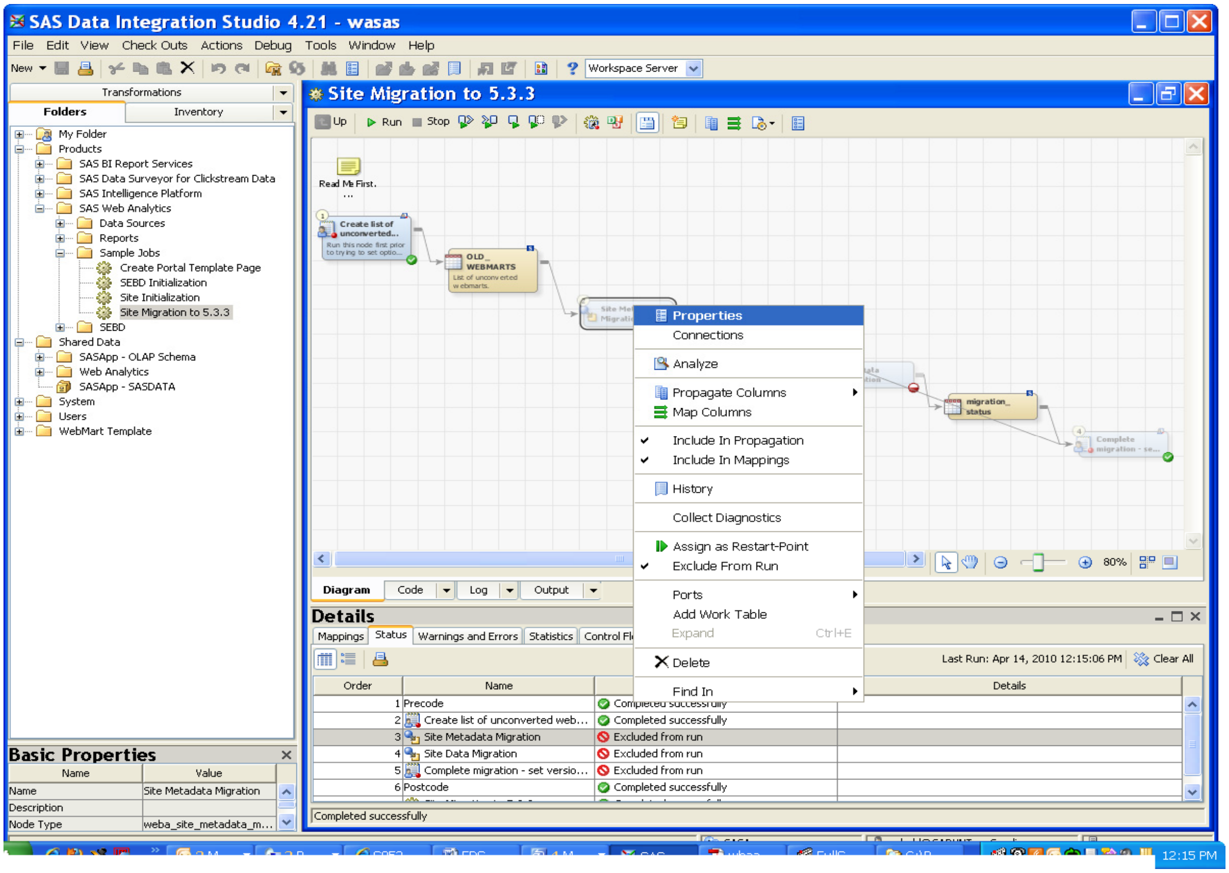
4 Run the **Site Migration to 5.3.3 Job** without modifications to generate the Old_Webmarts view. This view is used to populate a menu within the Site Metadata Migration transformation (node 2). If this step is omitted, an error message is displayed when the Site Metadata Migration transformation properties are opened.


Validation Error

Unable to execute query: SQL passthru expression contained these errors: ERROR: File WEBA.OLD_WEBMARTS.DATA does not exist.

5 Right-click on each **Excluded** transformation and deselect **Exclude From Run**.

Note: There are three transformations that need to be updated.



Last Run: Apr 14, 2010 12:15:06 PM

Order	Name		Details
1	Precode	Completed successfully	
2	Create list of unconverted web…	Completed successfully	
3	Site Metadata Migration	Excluded from run	
4	Site Data Migration	Excluded from run	
5	Complete migration - set versio…	Excluded from run	
6	Postcode	Completed successfully	

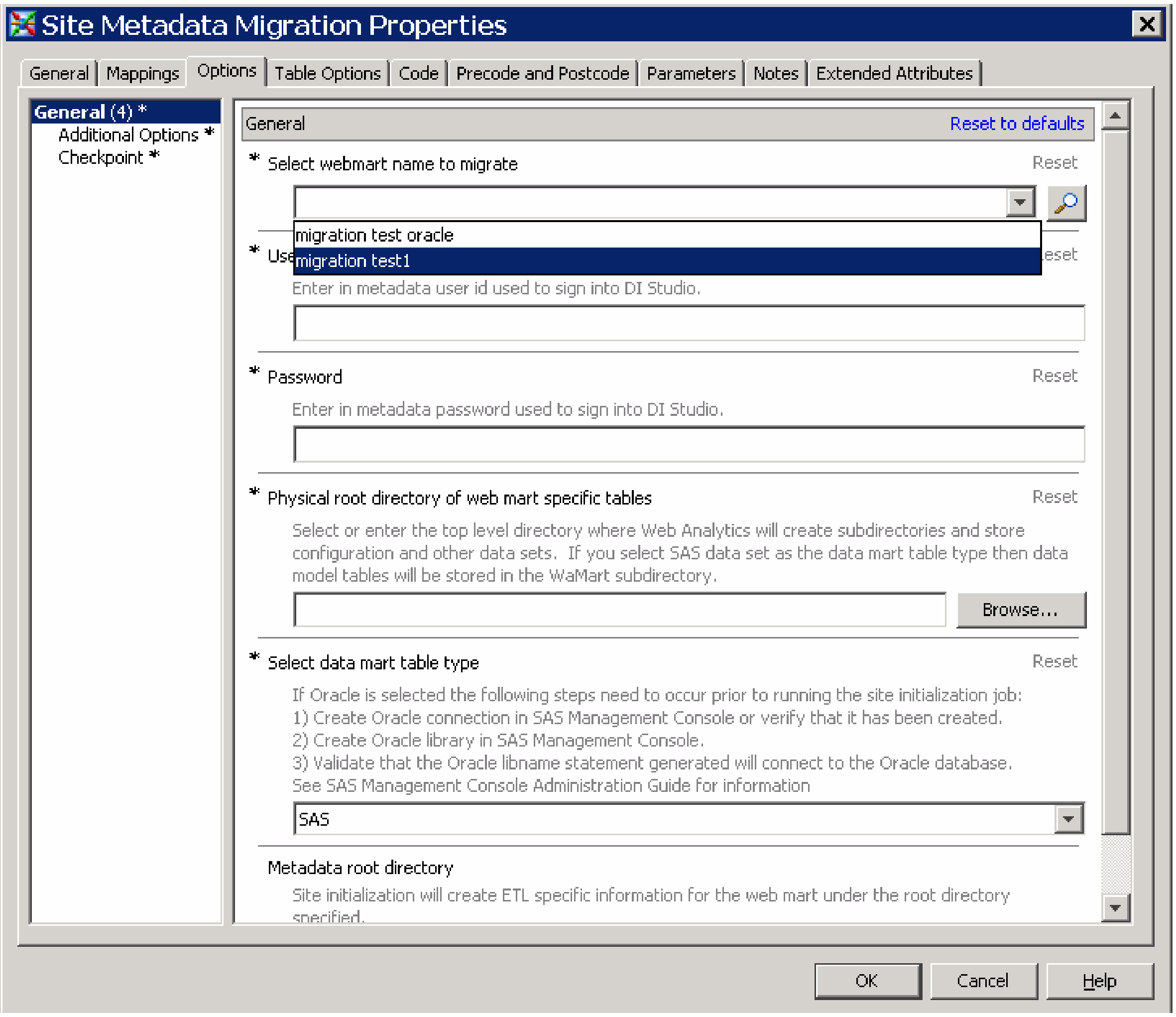
Completed successfully

Basic Properties

Name	Value
Name	Site Metadata Migration
Description	
Node Type	weba_site_metadata_m…

6 Double-click **Site Metadata Migration**.

a. Click the **Options** tab.



* Select webmart name to migrate

migration test oracle

migration test1

* User

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 subdirectories and store configuration and other data sets. If you select SAS data set as the data mart table type then data model tables will be stored in the WaMart subdirectory.

* Select data mart table type

If Oracle is selected the following steps need to occur prior to running the site initialization job:
1) Create Oracle connection in SAS Management Console or verify that it has been created.
2) Create Oracle library in SAS Management Console.
3) Validate that the Oracle libname statement generated will connect to the Oracle database.
See SAS Management Console Administration Guide for information

SAS

Metadata root directory

Site initialization will create ETL specific information for the web mart under the root directory specified.

b. Select the Web mart to migrate.

c. Enter the metadata logon information.

d. Enter the root drive where the data for the Web mart data is stored (for example, **`c:\data`**). If the name of the Web mart directory is not identical to the Web mart name in the selection list, then change the directory name to match the Web mart name.

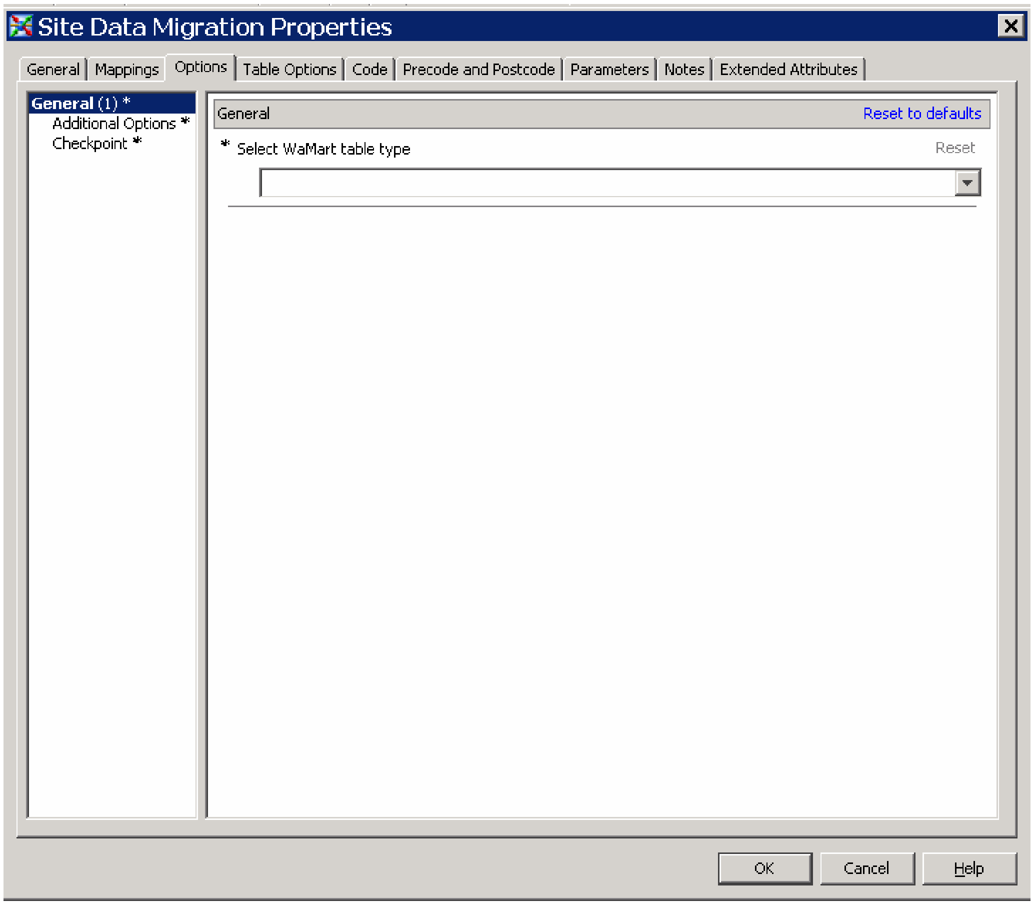
Note: If the Web mart name contains spaces, they are converted to underscores (_) in the directory name.

e. Specify whether the WaMart library is SAS or Oracle. For information about site initialization and how libraries are set up, see Chapter 2.

f. Click **OK**.

7 Double-click **Site Data Migration**.

a. Click the **Options** tab.

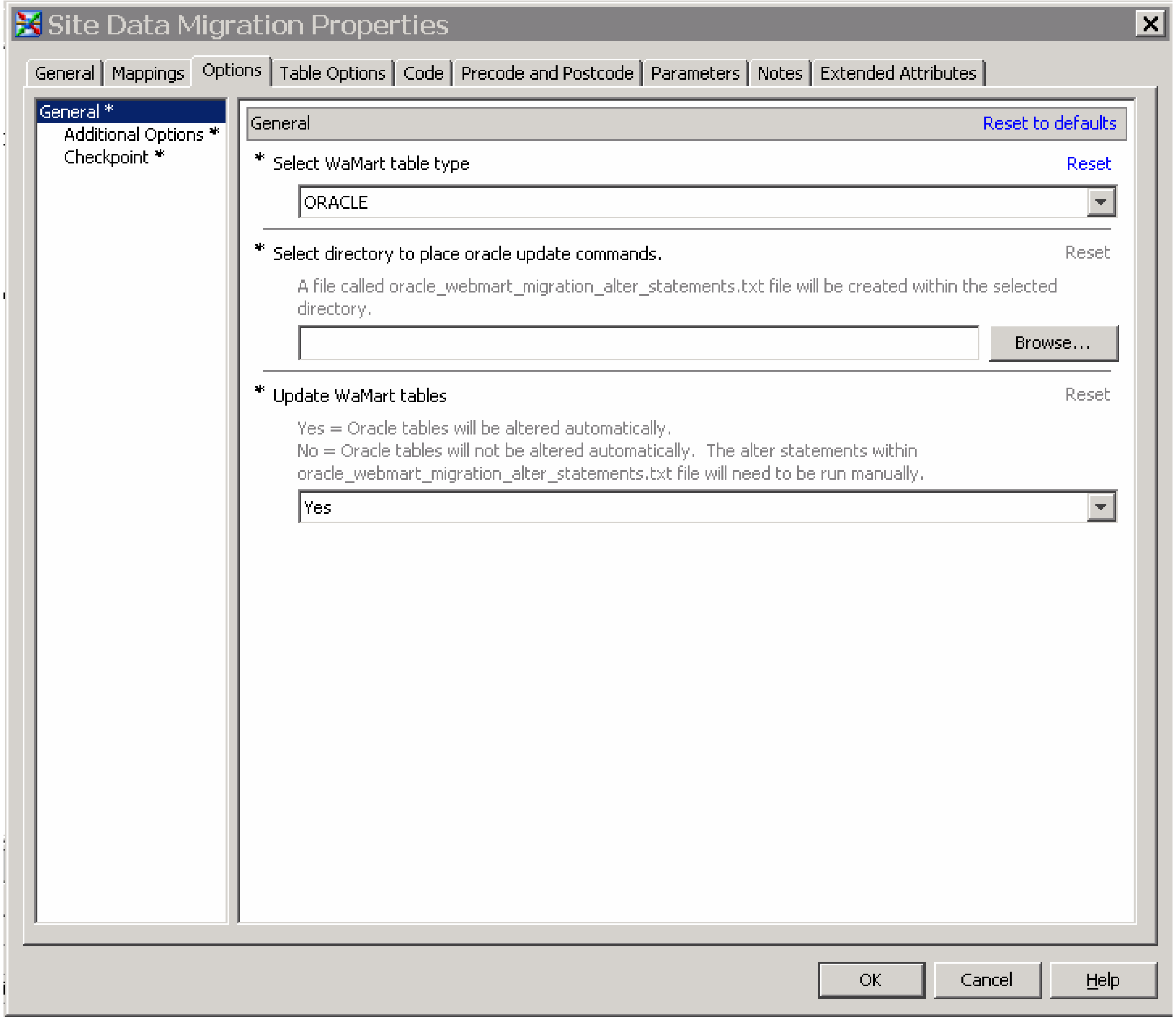

Site Data Migration Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
General (1) *
Additional Options *
Checkpoint *
General
Reset to defaults
Reset
OK
Cancel
Help

* Select WaMart table type

b. Select **WaMart Table Type**. If the data is stored in Oracle, then provide information for the following fields:

Directory - select a directory where the Oracle SQL alter statements will be saved.

Updating the Oracle tables - by default, the migration job automatically updates the tables. If you do not want to update the tables, select **No**.


Site Data Migration Properties
General | Mappings | Options | Table Options | Code | Precode and Postcode | Parameters | Notes | Extended Attributes
General *
Additional Options *
Checkpoint *
General
Reset to defaults
Reset
Browse...
OK
Cancel
Help

* Select WaMart table type: ORACLE

* Select directory to place oracle update commands.

A file called oracle_webmart_migration_alter_statements.txt file will be created within the selected directory.

* Update WaMart tables

Yes = Oracle tables will be altered automatically.
No = Oracle tables will not be altered automatically. The alter statements within oracle_webmart_migration_alter_statements.txt file will need to be run manually.

Yes

c. Click **OK**.

8 If your SAS Server tier runs on Windows, you can run the job in SAS Data Integration Studio. If your SAS Server tier runs on UNIX/LINUX, you will need to deploy the migration job to the UNIX machine and run it following the procedures for the Site Initialization job. For more information, see Chapter 2.

9 Run the job.

Before you run the first 5.3.3 ETL schedule, verify that the migration job ran with no errors.

1 If you are migrating from 5.3, delete the YY_* versions of the tables within the Web mart WaMart library.

2 Delete or rename the WACNFG.CNFG_INPUT_HISTORY data set.

3 If you are migrating from 5.3, regenerate the aggregates. For more information, see Appendix 5.

4 Redeploy all of your jobs from the `5.3.3 jobs` folder.

Migrating Search Engine Bid Data

Conversion to SAS Unicode Server

To convert the SEBD data mart to SAS Unicode:

1. Complete the SAS Unicode server setup. For more information, see the SAS Unicode topics in Chapter 1.
2. Archive the current SEBD directory structure. After archiving is complete, 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 about initialization steps, see Chapter 2.

Base SAS Changes

Make the following updates after you install and configure SAS Web Analytics 5.3.3, 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 Chapter 5.

Troubleshooting

Site Metadata Migration

- ❑ The message "Site already migrated" is displayed.

  This message indicates that the metadata has already been migrated. Verify that the following folders exist and are populated. If that is not the case, restore the metadata repository by using the backup made before the migration process.

/System/Applications/SAS Web Analytics/Web Analytics 5.3.3/Site/<*Web mart*>. Also ensure that the Web mart object and Support folder appear within this folder.

/Shared Data/Web Analytics/<*Web site*> folder

- 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 has the ability to modify and create new files within the directory.
- If there are problems creating the WaMart table, look at the PROC COMPARE output, and then fix the problem.
- If there are problems restarting the Site Migration job, edit the Site Migration job so that the first two nodes are **Excluded From Run**, and then run the job. The table migration node was designed to continue from the last successful WAMART table migration.

Deleting Aggregate Tables

If users do not have proper authorization, complete these steps:

1. Fix authorization problems (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 5.3.3 job.

-There were no error messages in the migration program log.

If both items are verified:

1 Open SAS Management Console.

2 Navigate to the **`/System/Applications/SAS Web Analytics/Web Analytics 5.3.3/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

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 at:

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

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.

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.

click-through
the act of navigating to a Web page by selecting a link in an e-mail message. See also page-open.

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 (extract, transform, load) process
the process of extracting data from a data source, transforming the data based on your business rules, and loading the data into your data warehouse.

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.

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
a data item whose values are aggregated (unless otherwise specified) and which 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. See also file count and hit.

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.

request
an attempt to access a Web page or a resource on a Web server. A request can be either a page request or a hit. See also page request and hit.

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 period of activity that starts when a visitor first accesses a particular Web site and that ends when the visitor has not performed any actions at that Web site within a specified time interval (usually 30 minutes). A session ID is associated with each session, and the activity that occurs during the session is recorded in a Web server log file.

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. See also session and entry point.

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. A status code between 200 and 299 indicates that the request was successful. A status code of 400 or greater indicates a bad request, an unauthorized request, a page not found, or some other type of failure.

stored process
a SAS program that is stored on a server and which can be executed as requested by client applications. There are two types of stored processes: IOM Direct Interface Stored Processes and SAS Stored Processes.

Uniform Resource Identifier
a string that identifies resources such as files, images, and services on the World Wide Web. A URL is a type of URI. See also Uniform Resource Locator.

Uniform Resource Locator
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.

unique visitor
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. See also session.

URI
See Uniform Resource Identifier. See also Uniform Resource Locator.

URL
See Uniform Resource Locator.

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 mart
a shortened form of the term Web site data mart, which refers to a data mart that contains information about a Web site’s visits and related customer intelligence.

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.

Index

Your Turn

We welcome your feedback.

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