

Gentran Integration Suite™

Reporting Services

Version 4.3

Sterling Commerce
An IBM Company

© Copyright 2007-2009 Sterling Commerce, Inc. All rights reserved.
Additional copyright information is located on the Gentran Integration Suite Documentation Library:
<http://www.sterlingcommerce.com/Documentation/GIS43/homepage.htm>

Contents

Reporting Services Best Practices	5
Databases	5
Eclipse Plug-ins	5
Fact Models	5
Report Definitions	6
Maps/Data Extraction	6
Data Retention	6
Testing	6
About Reporting Services	7
Reporting Services Components	7
Reporting Services Security Considerations	8
Fact Models Included in Reporting Services	8
Key Terms Used With Reporting Services	9
Reporting Services Checklist	11
Reporting Services Database Considerations	13
Specifying a Different Database or Pool for Reporting Services	14
Modeling the Data for Reporting Services	17
Reporting Services Planning	18
What Are the Questions to Answer?	18
How the Fact Model Affects Viewing Data	19
Building a Fact Model	20
Defining a Fact Set	20
Defining a Dimension	20
Defining a Mapped Extraction	21
Defining a Fact Pattern	21
Defining Properties	21
Validating the Fact Model	22
About Versioning in Reporting Services	23
Reporting Services Resources	23
How Editing the Fact Model Impacts the Fact Repository	23
Impact to Reporting When Fact Model is Changed	24
Importing and Exporting Reporting Services Resources	25
Extracting Reporting Services Data	26
Mapped Extraction Setup Checklist	27
Creating, Editing, and Formatting Reporting Services Reports	29
Default Reports	29
Parts of a Report Definition	30
Customizing the Appearance of Reporting Services Reports	31
Editing a Report Format	32

Reporting Services Frequently Asked Questions	34
About the Reporting Services Listeners	35
Setting Up Remote Listeners	35
Starting and Stopping the Reporting Services Listeners.....	35
About the startlisteners.properties.in File	36
Fact Model Components	37
Properties	37
Fact Set.....	38
Dimension	40
Mapped Extraction	41
Fact Pattern.....	42
Fact Model Naming Considerations	43
Checking In a Fact Model	44
Checking in a Fact Model for the First Time	44
Editing a Fact Model	44
Checking in an Edited Fact Model	44
Changing the Default Version of a Resource	45
Viewing Your Reporting Services Data	46
Default Reports	46
Reporting Services Properties File	47
Index	49

Reporting Services Best Practices

This section contains best practices for developing and using Reporting Services in Application.

Databases

- ◆ Before you begin using Reporting Services, decide where your fact repository should reside. As a best practice, it should not be in the same database as your Application database.

Here are some issues to consider when deciding which database to use for your Reporting Services fact repository:

What type of usage will the fact repository have? Will there be continuous, high volumes of data being extracted and stored to the fact repository, or will it be sporadic? Are there certain times of the day that will have high volume and others that will be slower? Will the data flow for Reporting Services impact the day-to-day operations of Application by using resources from the same pool?

Plan your fact model with the database you will use in mind. Ensure that you allow for any database restrictions on field or column width in the database you will use. Some versions of DB2 have a maximum column name size of 18 characters. So for DB2, if your fact names have more than 18 characters, the fact names will be truncated when the columns are created.

- ◆ Do your initial Reporting Services configurations and testing on a test system, then move the final version of your fact models and report definitions and formats to the production system.
- ◆ Configure your test system to drop tables each time you check in a new version of your fact model. This keeps your fact repository from getting too crowded with columns (each fact that is added at any point, even if deleted from a fact model, is retained in the fact set's table in the database. This can cause tables to become unnecessarily large).

Eclipse Plug-ins

Create a single workspace for each fact model and its associated reports and custom report formats. This will make it easier to keep track of which reports are associated with the fact model when in Eclipse.

Customize the Eclipse perspective to display the Reporting Services and MESA Developer Studio plug-in options on the menus.

Fact Models

There are often many ways to build a fact model, but just as often, there will be a “best” way to construct the fact model for your needs. Design the fact model with reporting in mind. For example, each fact set will generate a detail and summary report if the fact set is not linked to any other fact set. Each dimension's level generates a group in the reports. Each fact will become a report field. By visualizing how you want to view the data results, you can design the fact model to deliver the results that you want.

Report Definitions

Using Default Reports as Templates

Before creating any new report definitions, review the autogenerated report definitions to see if they meet your needs. Many times, instead of creating brand new report definitions, you can just edit these report definitions in the Report Editor to suit your needs, or customize them in Application by applying different filters and saving them as new report configurations.

Editing Default Reports

You can edit default reports, save them as new report definitions (or as new versions of the default report definitions). Edit report definitions to remove all unwanted fields and groups. When default report definitions are created, all possible groupings are created (one group per level in the fact model).

Make copies of your report definitions and work on the copies till you are satisfied with them. Do any customization of the report definitions on these copies. Once you have the report definitions finalized, you can create and save any report configurations that you need, based on the report definition copies.

Maps/Data Extraction

Set up your mapped extractions to use XML whenever possible, rather than having the data extracted using JDBC. When using XML, the output side of your map is ready to build – simply use the XML schema created when you check in your fact model as the basis for the output side of your map.

Data Retention

Set data retention for the fact repository by creating retention items in the fact model for the fact sets and fact patterns. The default is 60 days.

Testing

The Reporting Services log, BizIntell.log, can provide useful troubleshooting and debugging information while testing your Reporting Services installation. Turn the logging level to **On** for BizIntell.log while testing. To get to the log settings, go to **Operations > System > Logs**, go to the Reporting Services pane, and click the Edit icon.

Test translation maps separately, prior to using in Reporting Services. You can do this by setting up a simple business process that contains only the Start and End BPML activities and the Translation service. On the Translation service in the GPM, set the parameter Output to Process Data to **Yes**. This enables you to view the results of the translation in process data.

About Reporting Services

Reporting Services is a set of Application tools that enables you to define and see your data in ways that support your business.

Using Reporting Services consists of four parts:

◆ Planning

During this phase, you decide what information you want to ultimately see. What are the questions that Reporting Services needs to answer? What data do you want to see, and in what format? Do you need sales by year, month, and day? Do you need sales by region, salesperson, or product? Are you interested in seeing revenue vs. expenses by month, by division, and by department? Once you decide on all the questions to be answered, you can start to define the information.

◆ Defining and Building Data

In Reporting Services, data is defined and built using the Fact Model Editor. You define the data and relationships that will enable you to view your business results and facts in the most useful manner for you.

◆ Extracting Data

Once you define and build a fact model, you are ready to begin pulling the data you need from the datastream. By creating business processes that include the Mapped Extraction service, you are able to extract your data into a Reporting Service database called the *fact repository* that is created for your fact model.

◆ Viewing Data

As data is extracted into the fact repository, you can begin reviewing the default reports that are created automatically for your fact model in Application, to edit those reports, or use them as templates for new reports. If you use JPivot and Mondrian, you can export the Mondrian schema that is automatically created for your fact model in Application and view your data as a multi-dimensional cube.

Reporting Services Components

- ◆ Reporting Services Fact Repository – Reporting Services pulls facts from your data stream (systems, databases, business processes) and stores it in a database called the Reporting Services Fact Repository. This database should be separate from the Application database.

Reporting Services uses three Eclipse plug-ins that are accessed through the MESA Developer Studio:

Note: MESA Developer Studio does not work with Eclipse 3.2. You will need to use Eclipse 3.1.

- ◆ Fact Model Editor – Use to create a fact model that contains the facts and the relationships among the facts needed for you to see your data.
- ◆ Report Editor – Use to edit the two reports created by default for each fact model (Detail and Summary) and to create new reports based on your fact models.
- ◆ Format Editor – Use to define the appearance of your reports. A default report format is included with the Format Editor. You can create additional formats as needed.

Reporting Services also uses these Application features:

- ◆ Schema check-in and check-out – Fact models and its associated components (reports, Mondrian cubes) are checked in to Application using the Schema check in function.
- ◆ Reporting – Reports created for your fact models are accessed through the Application Reporting function.
- ◆ Map Editor – You can use one or more any-to-JDBC or any-to-XML maps to extract data to the fact repository.
- ◆ Business Processes and Services – Reporting Services includes two new services: the Mapped Extraction service and the Straight Through Extraction service. Each is used in business processes to extract data and store it in the fact repository.
- ◆ Events – Application events can be used to trigger updates to the fact repository. The Systems and Operations fact model included with Reporting Services makes extensive use of the system and operational events in Application to gather its data.

Reporting Services Security Considerations

Before you begin using Reporting Services, decide who will have access to the Fact Model Editor and other Eclipse plug-ins, as well as to the other parts of Application: schema, report, service, map, and business processing functionality. Decide where the fact repository will be located. For information about how to set up access to various features based on user accounts in Application, see *Role Based Security*.

Fact Models Included in Reporting Services

Reporting Services comes with two fully operational fact models and report sets for Application systems and operations data. You can check out the schemas for the fact models and open them in the Fact Model Editor in Eclipse. Do not change these fact models. If you want to make changes, it is strongly recommended that you copy the fact model and make changes to the copy. Changing the original system and operational fact models can cause unpredictable results.

The included fact models are:

- ◆ system.bifm – Tracks system log information
- ◆ WFFactModel.bifm – Tracks detailed information about business process events in Application.

Example Bank Fact Model

Although not installed with Reporting Services, the BankFactModel is available in the `[Install_dir]/bi` folder. You can check in the fact model, or open it in the Fact Model Editor, and use it as an example.

Key Terms Used With Reporting Services

Term	Definition
Cube, Multi-Dimensional Cube, Multidimensional database	A data structure that has at least three dimensions. It is the fundamental structure for information in an OLAP (On-Line Analytical Processing) system.
Dimension	A collection of data of the same type. For example, "Time" could be a dimension.
Fact	A single piece of raw data.
Fact Meta Data	Data that describes the fact. Examples are data types and acceptable values, mappings to the data source, descriptions of any transformation of the data, and any information needed to support the fact model.
Fact Name	The reference name for a specific data field. This referenced name must be used consistently to ensure accurate Reporting Services reporting. For example, using the same data name for two different types of data could cause an unintentional mix of data.
Fact Model	A named, versioned Application resource that contains a group of fact sets, meta data about the named fact, and measures, dimensions, and links for the data.
Fact Pattern	Defines a parent/child relationship between two fact sets.
Fact Set	A group of logically categorized facts.
Fact Repository	Specified Reporting Services data storage area.
Hierarchy	Standard tree view (parent with one or many children) organization for positions in a dimension.
JPivot	A custom JSP (Java Server Pages) tag library that allows users to perform common OLAP (Mondrian and XMLA) tasks.
JOLAP	(Java Online Analytical Processing) is a Java application-programming interface (API) for the Java 2 Platform, Enterprise Edition (J2EE) environment that supports the creation, storage, access, and management of data in an online analytical processing (OLAP) application.
Level	Within a dimension, facts may have various positions for detail. For example, a time dimension may have levels for minute (level 1), hour (level 2), day (level 3), etc.
Listener	In Reporting Services, a listener is used to wait (listen) for specified events to take place which have been defined as triggers for data extraction. Listeners are internal to Application – the only configuration required is to start the Event Listener.
MDX	Multi-dimensional expressions. MDX is the main query language implemented by Mondrian.

Term	Definition
Measure	A numeric fact or result of a calculation.
MOLAP	OLAP data is stored in a multi-dimensional database
Mondrian	Mondrian is the OLAP that is being delivered as part of Reporting Services solution. Mondrian is an OLAP engine written in Java. Executes queries in MDX, reads data from relational databases and presents the results in multi-dimensional format.
OLAP	Online Analytical Processing systems used to analyze large quantities of data in real time.
Retention	The amount of time defined to keep Reporting Services data in the fact repository.
ROLAP	OLAP data is stored in a relational database.
Star Schema	A data structure that is has a central fact set that has pointers from other facts.
XMLA	Open industry-standard web service interface designed specifically for online analytical processing (OLAP) and data-mining functions

Reporting Services Checklist

Use the following checklist to help you plan your Reporting Services implementation. It is recommended that you follow the sequence shown in the table for these tasks.

Process	Description
Phase 1 – Planning	<ul style="list-style-type: none"> ◆ What are the questions that you want answers to? ◆ What are the business processes that will be needed to get the data into the Reporting Services Fact Repository? ◆ What data translation must be done? ◆ What services are required? ◆ Where will the fact repository be kept?
Phase 2 – Installation	
Install Application	You must install Application before you can install MESA Developer Studio and Reporting Services.
Install MESA Developer Studio and Reporting Services	MESA Developer Studio is required for Reporting Services. If you have not already purchased and installed the MESA Developer Studio separately, install it with Reporting Services. You must install the Mesa Developer Studio component either before or at the same time you install Reporting Services. You cannot install it after you install Reporting Services.
Phase 3 – Defining and Extracting Data	
The processes in this phase are iterative, although the order or number of steps may change with each iteration. It is recommended that you wait to customize your reports until you are satisfied that your fact model is correct, and that the autogenerated reports display the information that you want to see.	
Plan and create a fact model	You create the fact model in the Reporting Services Fact Model Editor in Eclipse. How you structure the fact model determines how you will ultimately be able to view your data.
Define the data	Once you have created the fact model, use it to define all of the data and relationships necessary.
Create maps	If there is data that must be translated or transformed in some way, create maps in the Application Map Editor to perform the necessary functions. Check the maps into Application and test them. In the fact model, add one or more mapped extraction components for the maps (depending on whether the maps will be used in a single business).
Create or edit business processes	To populate the Reporting Services fact repository, you must create new business processes or edit existing business processes in Application to extract the data. Check in the business processes and test them.
Start listeners	To begin collecting data, you must start the listeners in Application. See Starting Listeners.

Process	Description
Check in the fact model to Application for the first time	Checking in your fact model to Application creates the basic set of reports, a schema for a Mondrian cube, and the fact model schema itself.
Check out fact models	To edit a fact model that you have already checked in to Application, you must check it out again. Open the file in the Fact Model Editor in Eclipse. To open the file in Eclipse, drag and drop the .bim file into the project created for this fact model and its reports.
Check in fact model again	Once you have made any changes to the fact model, you must check in the fact model to Application. When you check in the fact model, all associated files (reports, report formats, Mondrian schemas) are regenerated and reflect the updates to the fact model. Note: If you need to go back to an earlier version of one or more files, .bak files are created and stored on the file system.
View default reports	View the basic set of reports for your fact model. Ensure that they reflect the answers to your planning questions.
Phase 4 – Edit Existing Reports and Create New Reports	
Once you have completed all the iterations necessary of the preceding processes, you are ready to edit the existing reports or create new ones.	
Check out reports and report formats	If you want to edit the reports or the report formats, check them out of Application.
Open reports and report formats in Eclipse editors	To edit an existing report or report format, you drag and drop the report or format file that you have checked out of Application into the Eclipse editor.
Create and edit reports and report formats	You can create custom reports for your fact model using the BI Report Editor in Eclipse. You can also create custom report formats, using the Reporting Services Format Editor in Eclipse.
Check in reports, report formats	Once you have made any changes to a report or report format, you must check in the file to Application. This will create a new version of the report or format.

Reporting Services Database Considerations

The Reporting Services fact repository in Application can be any of the following databases:

- ◆ DB2 8.1
- ◆ MySQL 4.1
- ◆ Oracle 9I
- ◆ Oracle 10g
- ◆ MSSQL/SQL Server 2000/2005

When planning your Reporting Services installation, decide how you want to set up your fact models and databases. The following guidelines can help you find the most efficient and safest methodology for your data and environment:

Databases:

- ◆ Consider whether or not the database you use for Application is the database best suited to Reporting Services usage, as well. By default, Reporting Services will build the repository in the same database as Application. Some databases may have restrictions, such as keyword or column restrictions, so bear these in mind when selecting a database to use. For example, manual updating of the table structure may be required for the Date field for custom fact models installed using DB2, which does not accept NULL values.

Catalogs and Pools:

- ◆ If using multiple fact models that will have the same fact set names, you must use different database pools for each fact model.
- ◆ Even if the fact models will not use the same fact sets, other reasons, such as security, might make using separate database pools for each fact repository a better solution.

See *Specifying a Different Database or Pool for Reporting Services* for more information.

It is important to consider the database layout when designing your fact models; the parts of the fact model directly relate to the parts of the database, as follows:

- ◆ Database name = name of fact repository
- ◆ Table name = Fact set name
- ◆ Column name = Fact name
- ◆ Max columns per table = Max facts in a table
- ◆ Max table row length = The total table row length cannot exceed the length shown in the table below

The limitations for each database are shown in the following table:

	Database name max length (bytes)	Table name max length (bytes)	Column name max length (bytes)	Max columns per table	Max table row length (bytes)
DB2 8.1	8	128	128	1012	32677

	Database name max length (bytes)	Table name max length (bytes)	Column name max length (bytes)	Max columns per table	Max table row length (bytes)
MySQL 4.1	64	64	64	2599	2GB
Oracle 9i	8	30	30	1000	2000
Oracle 10g	8	30	30	1000	2000
MSSQL/SQLServer 2000/2005	128	128	128	1024	8060

Specifying a Different Database or Pool for Reporting Services

By default, Reporting Services uses the same database and pool for its fact repository as your Application system uses for its database. However, if you are using multiple fact models that will contain the same fact set names, you must set up different database pools for each fact model/fact repository.

To set up Reporting Services to use a different database pool:

1. Open the `customer_overrides.properties` file, which is located in the `[install_dir]/properties` folder on your Application system.
2. Add the following line, substituting the name of the pool to be used for the italicized example text:
`bi.poolName=mysamplePool`
3. Add a JDBC entry for the database pool you defined in step 2.
4. Save the file.

Note: See *Sample Customer Overrides Properties File* for examples of entries to the `customer_overrides.properties` file.

5. Open `startlisteners.properties.in`, which is located in the `[install_dir]/properties` folder on your Application system.
6. In the file, locate the `ifdef` statement for your environment. Add lines for each vendor jar file that you will use in the section for the `ifdef` statement. For example:

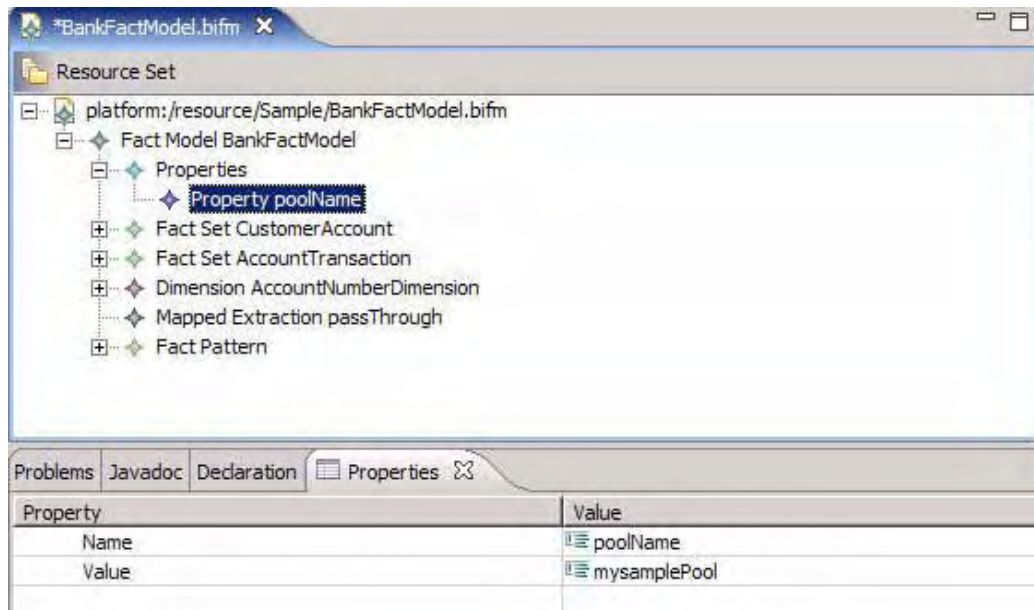
```
#:ifdef WINDOWS
VENDORJAR_DB=&JAR_JAVA_HOME;/jre/lib/ext/&JDBC_DRIVER;
#:else
VENDORJAR_DB=&JDBC_DRIVER;
setSystemProperty@java.io.tmpdir=&INSTALL_DIR;/tmp
#:endif
```

Note: You can choose any names for the entries, but each name must be unique. They do not have to share the same prefix (`VENDORJAR_DB`, for example), but using this type of convention may make it easier to locate the entries in the file.

- Save and make a backup copy of the file.

Caution: If you apply a new Application patch or update, this file will be overwritten. Keep a backup copy so that you do not have to recreate all of your changes after a product update. You can add your changes to the new version of the `startlisteners.properties.in` file after running an update.

- Install the jar files for each vendor using the `install3rdParty` script in the `install_dir/bin` folder on your Application system. See *Managing Services and Adapters* for more information about using the script.
- In your fact model, add a Properties element. Under the Properties, add a Property child element. In the Name field, enter `poolName`. In the value field, enter the name of the pool that you referenced in the properties files. For example:



- Validate the fact model (either validate just the new properties element, or validate the entire fact model from the root element), save it, and check in the fact model to Application.
- Run `setupfiles.sh` (UNIX) or `setupfiles.cmd` (Windows) to apply the properties changes.
- Stop and restart Application.
- Restart the event listeners.

Note: For more information about modifying properties and using the `customer_overrides.properties` file, see *See Overriding Property File Settings*.

How Reporting Services Determines Which Database Pool to Use

Reporting Services will use the default database pool used by Application, unless you have specified a different pool name in another location. Reporting Services checks for and uses database pools in this order of precedence:

- First, Reporting Services checks for a pool name specified in the fact model itself (as a Properties/Property element).

2. If no pool name is in the fact model, it checks for a pool name in the customer_overrides.properties file (as an override value for bi.properties).
3. If no pool name is specified in either of the first two locations, it checks for and uses the pool name specified in the bi.properties file.
4. Last, if no pool name is specified in any of the first three locations, it will use the pool name specified in the jdbc.properties file.

Sample Customer Overrides Properties File

The following example entries for a customer_overrides.properties file show two pools specified for use with fact repositories.

Note: Only one pool can be specified as the Reporting Services pool (bi.poolName). Only reports for the fact repository that uses the Reporting Services pool can be scheduled; any others must be run manually. See the *Release Notes* for more information about reporting issues when using multiple database pools and fact repositories.

In the example, italicized text indicates that this is an item which requires a value for your system.

```

bi.poolName=myPool

jdbcServiceCustomer.oraclePool.driver=oracle.jdbc.driver.OracleDriver
jdbcServiceCustomer.oraclePool.url=jdbc:oracle:thin:@myoracleserver:0000:mycatalog
jdbcServiceCustomer.oraclePool.user=userID
jdbcServiceCustomer.oraclePool.password=password
jdbcServiceCustomer.oraclePool.prop_TCP.NODELAY=YES
jdbcServiceCustomer.oraclePool.maxconn=30
jdbcServiceCustomer.oraclePool.catalog=mycatalog
jdbcServiceCustomer.oraclePool.schema=myschema
jdbcServiceCustomer.oraclePool.type=remote
jdbcServiceCustomer.oraclePool.dbvendor=oracle
jdbcServiceCustomer.oraclePool.buffersize=500
jdbcServiceCustomer.oraclePool.maxsize=28
jdbcServiceCustomer.oraclePool.initsize=1
jdbcServiceCustomer.oraclePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc
c.ConnectionFactory
jdbcServiceCustomer.oraclePool.behaviour=2
jdbcServiceCustomer.oraclePool.waittime=1000
jdbcServiceCustomer.myPool.driver=com.mysql.jdbc.Driver
jdbcServiceCustomer.myPool.url=jdbc:mysql://myserver:0000/anothercatalog?useUnicode
=true&characterEncoding=UTF-8
jdbcServiceCustomer.myPool.user=userID
jdbcServiceCustomer.myPool.password=password
jdbcServiceCustomer.myPool.maxconn=20
jdbcServiceCustomer.myPool.catalog=anothercatalog
jdbcServiceCustomer.myPool.dbvendor=mysql
jdbcServiceCustomer.myPool.buffersize=500
jdbcServiceCustomer.myPool.maxsize=28
jdbcServiceCustomer.myPool.initsize=1
jdbcServiceCustomer.myPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.Co
nnectionFactory
jdbcServiceCustomer.myPool.behaviour=2
jdbcServiceCustomer.myPool.waittime=1000

```

Modeling the Data for Reporting Services

In Reporting Services, defining your data is done using fact models and mapping. A *fact model* is a named, versioned Application resource that contains facts, metadata about the facts, and measures, dimensions, and links for the data. All of the information that you ultimately have access to in reports is derived from the fact model. In Application Reporting Services, you use the Fact Model Editor to create fact models.

To get data into the fact model, the Fact Model Editor pulls facts directly from a datastream using a mapped or straight through extraction.

Reporting Services Planning

Before you begin creating a fact model, you need to fully understand the domain that the fact model will represent. For example, some banking domains might be currency exchange, credit card, or consumer banking.

What Are the Questions to Answer?

Using Reporting Services begins by asking questions: What information do you want to see? How often does the data need to be updated - on demand or on a schedule, or both? How often will people access it? Who will access it? How should the data be broken down? By time increments (monthly, yearly)? What other measurements are needed (by salesperson, department, company, product line, etc.?)

What raw data is needed? Where does it come from? What format is it in? Which pieces of data will need to be combined, and how, to get to the final form of the information you want to see?

By answering these questions, you can come up with a blueprint for your Reporting Services installation.

Facts

The lowest level of information that you will need to define in Reporting Services is a *fact*. A fact is a single piece of raw data. For example, a fact for sales data might be a product ID or a price, or a salesperson's name. A fact for expenses might be a type of utility, such as electric or water. Or it could be an item that your company purchases, such as a box of paper. Or, it could be the salary per pay period for an employee.

Define facts for only the items you really need to answer your planning questions. The saying "More is better" does not apply to fact models. The goal is to have exactly the facts you need and no more. Extra information will only slow processing and report generation, and clog the fact repository with unnecessary data.

Grouping Facts and Defining Relationships

Determine how the raw data should be assembled to achieve the desired end result. Once you have decided on the facts, what calculations must be performed, and what relationships between facts must be defined, to answer your questions?

In Reporting Services, you can define:

- ◆ Fact sets – Groups of related facts
- ◆ Measures – Performs calculations such as sums and counts on facts
- ◆ Dimensions – Defines a hierarchical relationship among facts
- ◆ Fact Patterns – Identifies a primary fact set in a fact model and explicitly links it to one or more child fact sets

How the Fact Model Affects Viewing Data

It is essential to spend time planning your fact model. Usually, there are multiple ways to build a fact model. Your planning objective is to decide how to build the fact model so that it enables you to get the reports you need. There can be different ways to define facts and relationships in a fact model; however, when building the fact model, always consider how the arrangement of facts and the way they are grouped will affect the reports.

Default reports are generated for each fact model when it is checked in to Application. To provide the most flexibility, the detail reports contain all possible options. You can edit the reports in the Report Editor to limit or change which data is included.

Reports are created as follows:

- ◆ A default detail and a summary report are created for each fact set that is not joined to any other fact set in the fact model, if a measure was defined.
- ◆ If a fact set is used in a fact pattern, a detail report and a summary report are created for each grouping where all the fact sets in the fact pattern are present in the report. This may produce a detail report with a large number of columns that you can customize.
- ◆ If a fact set does not have a grouping, a detail and summary NoGroup reports are generated.
- ◆ Each level creates a group report that is grouped by the last column in the fact path. That is, if you entered BookInfo/ISBN as the fact path for a level, the group report would be grouped by ISBN.
- ◆ Measures are used to create summary and group footer sections of the reports.

Building a Fact Model

First, you must create an Eclipse project, then a fact model.

Note: For fact models, it is recommended that you use the Simple project type.

To create a project:

1. Select **File > New > Project**.
2. On the New Project wizard, select the type of project you want to create, for example, **Simple > Project**.
3. Enter a name for the project and click **Finish**.

To create a fact model:

1. In the Eclipse Package Explorer, select the project you just created.
2. Select **File > New > Fact Model**.
3. Enter a name for the fact model file. Use the default extension, .bifm. Click **Finish**.
4. Double-click the name of the fact model in the Package Explorer to open it in the Resource Set pane.
5. Select **Properties** from the tabs below the Resource Set pane.
6. Select the fact model line in the Resource Set pane and enter the name for the fact model in the Name property field in the Properties pane. This must be the same name that you entered in step 3 for the fact model filename (without the .bifm extension).

Defining a Fact Set

In the fact model, right-click the fact model line in the Resource Set pane.

1. Select **New Child > Fact Set**.
2. In the Properties pane, enter a name for the fact set.

Add child elements to the fact set as needed. To add a child element, right-click the fact set and select New Child and then the type of element to create (Fact, Measure, Retention).

Defining a Dimension

Specifying dimensions for your fact model enable you to categorize your data. For example, you could create the dimension “time” and beneath this category, have levels for day, week, month, and year. Those levels can then be aggregated to define a hierarchical structure for the data - combining months into the yearly data, for example.

In the fact model, right-click the fact model line in the Resource Set pane.

1. Select **New Child > Dimension**.
2. In the Properties pane, enter a name for the dimension.

Add only one hierarchy child element to the dimension. To add a hierarchy, right-click the dimension and select **New Child > Hierarchy**.

Each hierarchy can contain multiple level elements. To add a level element, right-click the hierarchy and select **New Child > Level**.

Defining a Mapped Extraction

Mapped extractions enable you to manipulate or translate data prior to using it in Reporting Services. Before you can define a mapped extraction, you must create the translation map to be used by the extract. Either before or after defining the mapped extraction, you must create a Mapped Extraction service configuration and a business process that will extract the data and put it in the Reporting Services repository.

In the fact model, right-click the fact model line in the Resource Set pane.

1. Select **New Child > Mapped Extraction**.
2. In the Properties pane, enter a name for the mapped extraction.
3. Click the ellipsis (...) to the right of the Map Name field, which displays a map entry page.
4. Enter the name of the map to be used for this extraction and click **Add**. Repeat to add other maps, if necessary. Once you have added all maps, you can rearrange them using the Up and Down buttons.

Note: All maps added to a single mapped extraction must be of the same type – either XML or DB (database).

Defining a Fact Pattern

The fact pattern is the primary fact set in a fact model, when using a star schema, or non-hierarchical fact model.

In the fact model, right-click the fact model line in the Resource Set pane.

1. Select **New Child > Fact Pattern**.
2. There are no fields for the fact pattern itself; however, you must add one root fact set element to the fact pattern. You can optionally add a retention element.

To add a root fact set, right-click the fact pattern and select **New Child > Root Fact Set**.

Each root fact set can contain one or more fact set ref elements. To add a fact set ref element, right-click the root fact set and select **New Child > Fact Set Ref**.

Defining Properties

Use properties to specify a different database to be used for the fact repository.

In the fact model, right-click the fact model line in the Resource Set pane.

1. Select **New Child > Properties**.
2. There are no fields for the Properties element itself; however, you must add at least one property element beneath it.

To add a property, right-click the fact pattern and select **New Child > Property**.

Validating the Fact Model

You can manually validate the fact model or any individual components in the Fact Model Editor. The Fact Model Editor validation checks for well-formedness and does some datatype validation.

When you check in a fact model, report definition, or report format to Application, it validates the structure and content of the of the component against the schema and does internal consistency checks.

How to Validate in the Fact Model Editor

When you make changes to the fact model, you should validate it at the root level.

- ◆ To validate a single element, select the element and click **Fact Model Editor > Validate** from the drop down menus. You can validate elements at any level in the fact model. For example, you can select and validate a fact set and its child elements, or a fact itself.
- ◆ To validate the entire fact model, select the root fact model element and click **Fact Model Editor > Validate** from the drop down menus.

If the selected element is valid, the message “Validation completed successfully” displays. If the selected element is not valid, an error message is displayed. Click Details to see more specific information about why validation failed.

About Versioning in Reporting Services

Reporting Services fact models, report definitions, and report formats are all saved in Application as *versioned resources*. A versioned resource has the following characteristics in Application:

- ◆ Multiple copies of the resource can be saved in the system with the same name.
- ◆ There is a default version of the resource. For Reporting Services, this is the last version checked in to Application. However, you can manually select a different version to be the default.
- ◆ The date and time for each version of the resource is recorded.

Reporting Services Resources

There are several resources associated with Reporting Services. The following are versioned resources and are stored in the database.

- ◆ Fact Model (.bifm)
- ◆ Reports (.rdef)
- ◆ Report format (.rfmt)

Note: A default report format is supplied.

The following are *transient* resources – they are regenerated each time a fact model, report definition, or report format is checked in again.

- ◆ Report.xml – contains information about all reports accessed through the Application Reporting wizard.
- ◆ Language properties files – used to define screens. Unless needed for localization, do not edit these files.
- ◆ Report properties file – This file is stored on the file system.
- ◆ Mondrian schema (.mond) – This schema can be exported and used with JPivot or other tools to generate a Mondrian cube.

Whenever you check in a fact model, there is a cascading effect that creates new versions of the rest of the resources.

How Editing the Fact Model Impacts the Fact Repository

When you check in a new version of a fact model, it causes changes to the structure of the Fact Repository.

- ◆ Properties changes – only make changes to properties when instructed by Customer Support.
- ◆ Dimension changes – Causes grouping changes
- ◆ Facts
 - ◆ Changing a fact name: In the fact repository, each fact is a column. If you change a fact name in the fact model, a new column with the new name is generated when you check in the revised fact model. The old column (under the old fact name) is preserved for reporting and tracking purposes.

Note: You must manually migrate any data from the old fact column to the new column. See your database vendor's query tools documentation for information on using queries to migrate data.

Caution: Reports based on the older version of the fact model *may* still run; however, because not all resources associated with a fact model are versioned, the report wizard will not display order lists correctly and can display fields that were added to the fact model after this report version was created. It is recommended that you do not run or rely on reports based on older versions of the fact model.

- ◆ Deleting a Fact: there is no change to the fact repository, because the old column for the fact is preserved, regardless. However, reporting resources are impacted by deleting a fact.

Note: To work around the impact, you can change the name of the fact set; however, this also has implications. See *Fact Set Changes*.

- ◆ Data Type: You can change a data type to a like data type - for example, one numeric type to another. To go to a different data type (for example, string to floating decimal point), you can set the autoddl property in bi.properties to drop all tables for the fact model upon check in. There is a property in the fact model as well, but it is not recommended to set this property in the fact model. If you import the fact model and its components to a new system (for example, test to production), the setting will be carried over with the fact model and could result in loss of data. Instead, use the bi.properties file, which will not be imported with the fact model, so setting will not carry over.
- ◆ Fact length - you can increase a fact length but cannot decrease it, to avoid loss of data through truncation.
- ◆ Fact Patterns and Links - the only impact to the fact repository is that it is reindexed. The old indexes are preserved as well as the new.
- ◆ Fact Set Changes
 - ◆ Changing a fact set name – the fact set name is the table name, so changing the name of a fact set will force creation of a new table. Any data that was in the original name table will not be transferred or migrated by Reporting Services or Application. You would have to manually migrate all data.
 - ◆ Be careful that if you are using multiple fact models that you do not use the same fact set names within each. If you need to do so, you can configure separate database pools for each fact model, which will mean that each has its own fact repository.
 - ◆ Deleting a fact set – The table remains in the fact repository for that fact set but no further data is added to it.

Impact to Reporting When Fact Model is Changed

Report resources are changed by edits to the fact model.

- ◆ Resources are generated on fact model, report definition, or report format check in.
- ◆ Deleting a fact - the Report Configuration wizard pages and reports will no longer contain those fields.
- ◆ Adding a fact - The new field is added to the Report Configuration wizard pages. The Report Configuration wizard pages will always reflect the most recent version of a report definition. So it is possible that older report definitions may not display correctly in the Report Configuration wizard.

One workaround is to give the fact set that contains the additional fact a new name. This generates a new report definition with its own Report Configuration wizard pages.

- ◆ Label changes - If you change a label in the fact model, the report wizard will display the new labels.

Note: There is a setting in the bi.properties file that sets Application to create a backup for the report.xml, report properties and language properties files. The backups are generated each time you check in one of the versioned resources. All of the backup files are preserved on the file system with the extensions bak.1, bak.2, etc. Use the customer.properties file to set the property value.

Importing and Exporting Reporting Services Resources

For Reporting Services, when importing or exporting resources, you must manually select the fact model and its associated report definitions and any custom report formats or report configurations. Report definitions, report formats, and report configurations are not automatically bundled with a selected fact model for import or export – you must manually select each file to be imported or exported. When importing a fact model and its associated report files, you must import them in the following order to avoid any possibility of having autogenerated resources overwrite your imported files:

1. Fact model
2. Report formats
3. Customized report definitions
4. Customized report configurations

For more information about Application resources and the import and export features, see *Managing Resources*.

Extracting Reporting Services Data

Data extraction is selectively pulling information (based on your fact model) and using it to populate the Reporting Services fact repository. Once you have set up your fact model and checked it in, you are ready to set up data extraction.

There are two types of data extractions used in Reporting Services:

- ◆ **Mapped Extraction:** use a Application map to specify data that will be pulled from your datastream, transformed or translated in some way, and placed into the Reporting Services fact repository.
If your data is already in the correct format for the fact repository (the schema format), you can use the “passThrough” option, which does no data translation. Using a mapped extraction element in the fact model, enter “passThrough” as the name of the mapped extraction and leave the Map Name field blank. Leave output type at default.
For mapped extractions, you will need to use the Mapped Extraction service in the business process you write for collecting data.
- ◆ **Straight Through Extraction:** Data is already in the correct format for the data repository and is extracted and placed directly in the fact repository with no transformation or translation. Data is collected using the Straight Through Extraction service in a business process.

A mapped extraction enables you to use a map to obtain Reporting Services data from a business process. Some considerations for mapped extractions are:

- ◆ A fact model can contain more than one mapped extraction element.
- ◆ A mapped extraction can refer to more than one map.
- ◆ A single mapped extraction can contain only maps of the same type, either XML or DB. To have a fact model use both XML and DB maps, create two separate mapped extraction elements, and include the XML maps in one and the DB maps in the other. It is currently recommended that you use XML mapped extractions, unless you have a great deal of expertise in creating and using databases in general, and creating and using database maps in the Application Map Editor.

Mapped Extraction Setup Checklist

Extracting data has many steps and separate processes. It is recommended that you follow the order of tasks shown in the following table when setting up a mapped extraction.

Task	Description
Create the fact model	<p>When creating your fact model, add the Mapped Extraction element. Decide on a map name now, and add it to the Mapped Extraction element.</p> <ol style="list-style-type: none"> 1 Enter the map name here, for reference later: Map name: _____ 2 Enter the Mapped Extraction element name here, for future reference (needed when creating the business process that will extract the data): Mapped Extraction name: _____
Check in the fact model	Checking in the fact model creates an xsd file, which is the schema you will use to create the output format of your transformation map.
Check out the xsd file	Check out the default version of the xsd (you may have gone through multiple iterations of checking in the fact model and resetting the default each time). Ensure that the file is can be accessed by the Application Map Editor.
Create a new map	<p>Using the Map Editor, create a map. Name the map using the name that you entered in the Fact Model (see the Create the Fact Model step).</p> <p>When creating the map, follow these steps:</p> <ul style="list-style-type: none"> ◆ Use the map type Sterling Integrator ◆ For Input format, select the format of your data (for example, delimited, variable, or XML) ◆ For Output format, when using a map for Reporting Services, you must select XML or SQL (database connectivity). Only use SQL if your mapped extraction is to a database. It is recommended that you extract data to XML instead of to a database. If you choose XML, click Customize. Choose Schema as the Customization File Type, and browse to the xsd file you checked out of Application. Select the file, and continue with creating the map.
Check in the map to Application	Once you have validated and compiled the map, check in both the source (.mxi) and compiled (.txo) versions to Application.
Test the map	<p>One way to test the map is to create a simple business process that contains just the Translation service. In the GPM, select the map to use for translation, and set the Output to Process Data parameter to Yes. This enables you to see the results of your translation right from the Business Process Execution Manager page.</p> <p>By doing this, you can isolate and fix any translation issues prior to using the map as part of Reporting Services.</p>

Task	Description
Create a business process for Reporting Services	<p>The business process that you create for data extraction must contain the Mapped Extraction service.</p> <p>Your final business process may be very complex, but for testing, create a business process that contains just the Mapped Extraction service. This enables you to verify that the data is being extracted to the fact repository correctly, and to view the reports. By limiting the number of steps in the business process, you can focus on the mapped extraction step.</p> <p>When you define the parameters for the Mapped Extraction service in the GPM:</p> <ul style="list-style-type: none"> ◆ Type the name of the fact model (without the .bifm extension) in the Fact Model field. ◆ Type the name of the mapped extraction element in the fact model (see the Create the Fact Model step) <p>Note: These fields are case-sensitive – ensure that the capitalization matches exactly the way the name appears in the Fact Model itself.</p>
Check in the business process	Once you have validated and saved the business process, check in the business process to Application.
Create sample input files	Create a sample input file in the correct format for the input side of your map, and ensure it is available to your Application system.
Execute the business process	Run the business process. On the Business Manager page, search for the business process using either the Search or List option. On the results page, click Execution Manager . On the Execution Manager page, click Execute. On the Execute page, type in or browse to the name of your sample input file, and click Go!
Check results by looking at reports	You can view your data by viewing the default summary and detail reports generated for each fact model. Go to Operations > Reports > and select Create New Report Configuration and click Go! Select the report you want from the list. The report names include the name of the associated fact model.
If necessary, make changes to the fact model.	If the data was not extracted to the fact repository correctly, review the bizintell.log to see if any errors were logged when the mapped extraction ran. Also, verify that the event listeners are running.

Creating, Editing, and Formatting Reporting Services Reports

Once you have checked in a fact model, you can view the autogenerated reports and note any changes that you want to make. See *Reporting* for information on viewing reports in Application.

You can check out report definitions in Application and open them in the Report Editor and edit them there. You can eliminate unwanted fields and groups from the report. You can change the order of filter fields. The order determines the default detail order list on the Report Configuration UI for a detail report.

Default Reports

Reporting Services provides a detail report that includes each fact and a group report for each dimension in the fact model.

To edit your default reports, check out the report definitions and open them in the Reporting Services Report editor.

1. Once you have checked out a report definition from Application, you must copy it into the Report Editor. After saving the report file to a location on your hard drive, open the Report Editor in Eclipse.
2. Create a new project, or ensure that the existing project you want to use for this report definition is listed in the Package Explorer in Eclipse.
3. Drag and drop the report from its location on your hard drive to the Package Explorer in Eclipse.

Note: The report definition filename and the Name attribute within the report definition must be the same. If you save the report definition under a different filename, you must also search the report definition and change all instances of the Name attribute; otherwise, you will receive errors when you check in the report definition to Application. The easier way to create new reports from a report definition is to use the Report wizard in Application. See *Reports* for information about using the Report wizard.

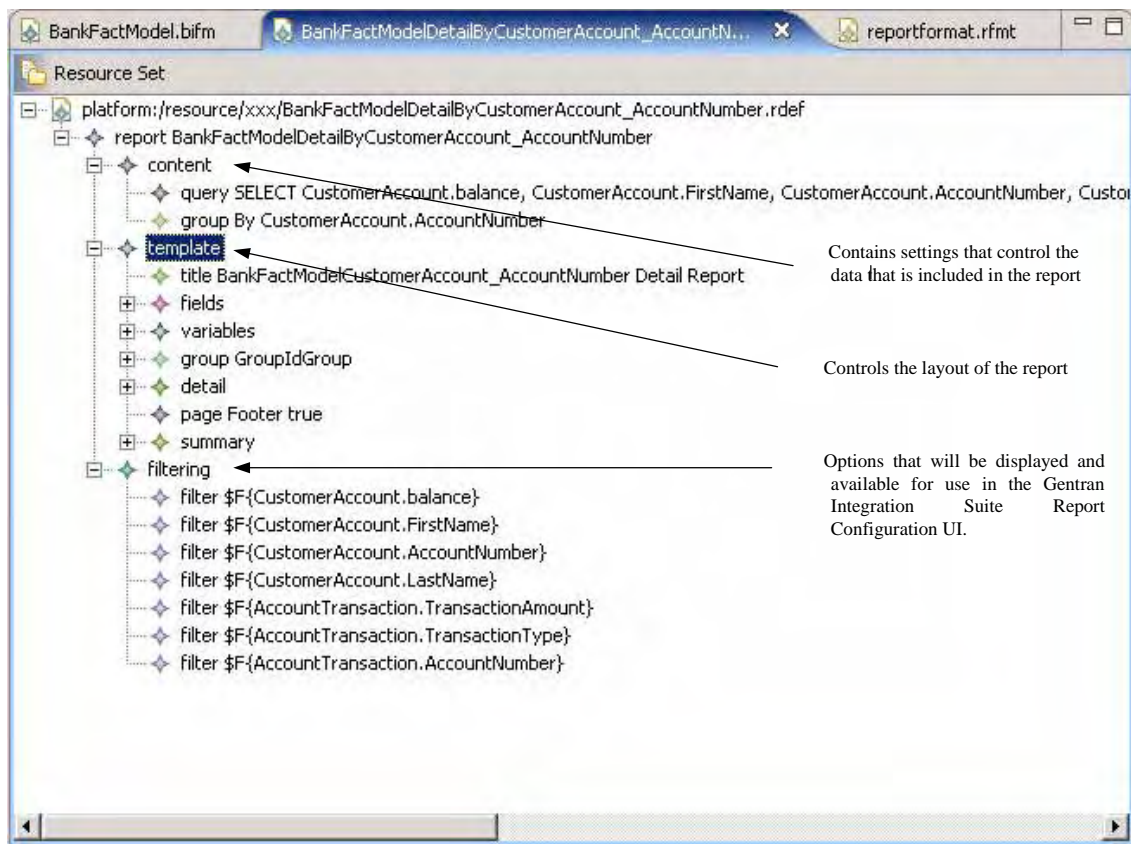
Default Report Content

The default reports contain the following values and sections:

- ◆ Query – The system includes one query for each fact when creating the default report template.
- ◆ Primary Key – Optional. Subset of fields in the query that uniquely identifies a record.
- ◆ Group by – One report definition is created for each level in the fact model.

Parts of a Report Definition

The following figure shows the three sections of a report definition:



Content

These are the query options, which include one for each fact. The types of queries are:

- ◆ Primary Key. Optional. Subset of fields in the query that uniquely identifies a record.
- ◆ Group by. One report section is created for each of the levels in the fact model.

Template

The following table describes the template options that can be changed:

Option	Description
Title	Text field for title. This includes two subfields: <ul style="list-style-type: none"> ◆ Title label – appears on the report ◆ Value – designates the variable or field
Field	One field is created for each fact in the fact model.

Option	Description
Variable	Created when there are measures in the fact model. If there is a Groupby in the content, then two variables are created for each measure in the fact model; one variable is for the group result and another variable for the entire report summary. When there are no groups, then, for each measure, only the entire report variable is created.
Group	Groups determine the “Report By” options available in the Application Reporting UI, and the sections in the report. Groups contain the following attributes: <ul style="list-style-type: none"> ◆ GroupID – Select true to display GroupID in the page footer. ◆ Within GroupID, there is a text field that contains a Title label and Value. The Value is a Java expression composed of fields/variables. Fields and variables would be wrapped by \$F() and \$V(), respectively.
Group Footer	This creates the group summary section of the report. It contains the following: <ul style="list-style-type: none"> ◆ Text field for group footer, which contains a Title label and Value. The value is a Java expression composed of fields/variables. Fields and variables would be wrapped by \$F() and \$V(), respectively. ◆ Line ◆ Rectangle
Page Footer	Takes value or true or false. If true, shows page numbers on report in the right footer section.
Detail	One created for each fact. Determines what will display in the detailed report that is automatically generated.
Summary	Shows an aggregated summary for all dimensions (groups). Options are the same as Group property options.
Configure UI report wizard filtering options	By default, the system creates a filter option for every fact. <ul style="list-style-type: none"> ◆ Control – type of UI control that should appear in the wizard for selecting filters. For example, for a text string, this would be a combo list box (select form drop down list), and for a numeric field it would be a text control (allows you to enter a string). ◆ Fields. One is created for each fact. ◆ Label. This is the text string that will appear as the label on this element (row header) on the report. <p>Note: The order of filter fields in the definition determines the detail order list on the UI for a detail report.</p>

Customizing the Appearance of Reporting Services Reports

You can create additional report formats by using the Reporting Services Format Editor in Eclipse. A default format is installed with Reporting Services. It is recommended that you do not change the original format, but you can create other “default” formats from it and change the formats and settings in those.

Note: To apply a different format to a report, you must edit the entry for the report in the bi.properties file.

Editing a Report Format

Examples of options you can specify are colors, fonts, spacing, or page layout for a specific report format.

Report format properties are based on the default report format. Any changes made override the defaults. Edits can be made to any of these properties without affecting the defaults for the others. To make changes that can be used on many reports, you can create a different default report format that can be referenced by more than one report.

Changes can be made to the following properties:

- ◆ Line
 - ◆ Align
 - ◆ Back color
 - ◆ Height
 - ◆ Width
 - ◆ X
 - ◆ Y
- ◆ Rectangle
 - ◆ Backcolor
 - ◆ Forecolor
 - ◆ Height
 - ◆ Position type
 - ◆ Width
 - ◆ X
 - ◆ Y
- ◆ Text Field Values, Label Title
 - ◆ Align
 - ◆ Back Color
 - ◆ Font Color
 - ◆ Font Size
 - ◆ Height
 - ◆ Value
 - ◆ Width
 - ◆ X
 - ◆ Y
- ◆ Title
 - ◆ Rectangle

- ◆ Value
- ◆ Report
 - ◆ Base type
 - ◆ Base type label. Used in UI as the base label for Report Configurations when a new one is created, for example.
 - ◆ Name
- ◆ Query String
 - ◆ Query statement string used for the report
- ◆ Template
 - ◆ Bottom margin
 - ◆ Date pattern
 - ◆ Generate fields
 - ◆ Generate group ID
 - ◆ Left margin
 - ◆ Orientation
 - ◆ Page height
 - ◆ Page width
 - ◆ Right margin
 - ◆ Top margin

Reporting Services Frequently Asked Questions

Can I Use Reporting Services to Generate Mondrian Cubes?

For each fact model that is checked into Application, a Mondrian cube schema is created. You can export and use this with tools such as JPivot.

About the Reporting Services Listeners

Reporting Services uses three listeners to extract data and populate the fact repository:

- ◆ Straight Through Extraction - Simple one-to-one extractions
- ◆ Mapped Extraction - Structured extractions based on maps
- ◆ Event Forwarding Listener - Converts existing Application events to straight through events

Setting Up Remote Listeners

To set up listeners on a system other than the Application server, you can set them up to listen remotely. The `remotelisteners.properties` file contains the procedures for setting up and invoking remote listeners for use with your Application system. The file is located in the `[install_dir]/bi/samples` folder of your Application system.

Starting and Stopping the Reporting Services Listeners

To begin populating the fact repository, the Reporting Services event listeners in Application must be started. Refer to the following sections for specific instructions for each platform.

UNIX Environment

To start the listeners, run the `startListeners.sh` command from the `bin` folder under your Application install folder.

There are two optional parameters:

- ◆ `debug` - turns on additional debugging messages.
- ◆ `<logfilename>` - sends all messages to the specified logfile name.

You can use one or both of the parameters, in any order. If the `debug` parameter is used, the system turns on those messages. If `debug` is not used, the system assumes it's a logfile name. The following is an example of the Start Listeners command for UNIX:

```
startListeners.sh debug <bi.log>
```

To stop the listeners, run the `stopListeners.sh` command from the `bin` folder under your Application install folder. The following is an example of the Stop Listeners command for UNIX:

```
stopListeners.sh
```

Windows Environment

For Application on a Windows system, the event listeners are a Windows service. By running `startWindowsServices.cmd`, you start all Application services as well as the listeners service.

The following are commands that can be used to individually control the event listeners service:

- ◆ `InstallEventListenersWindowsService.cmd`
- ◆ `startEventListenersWindowsService.cmd`

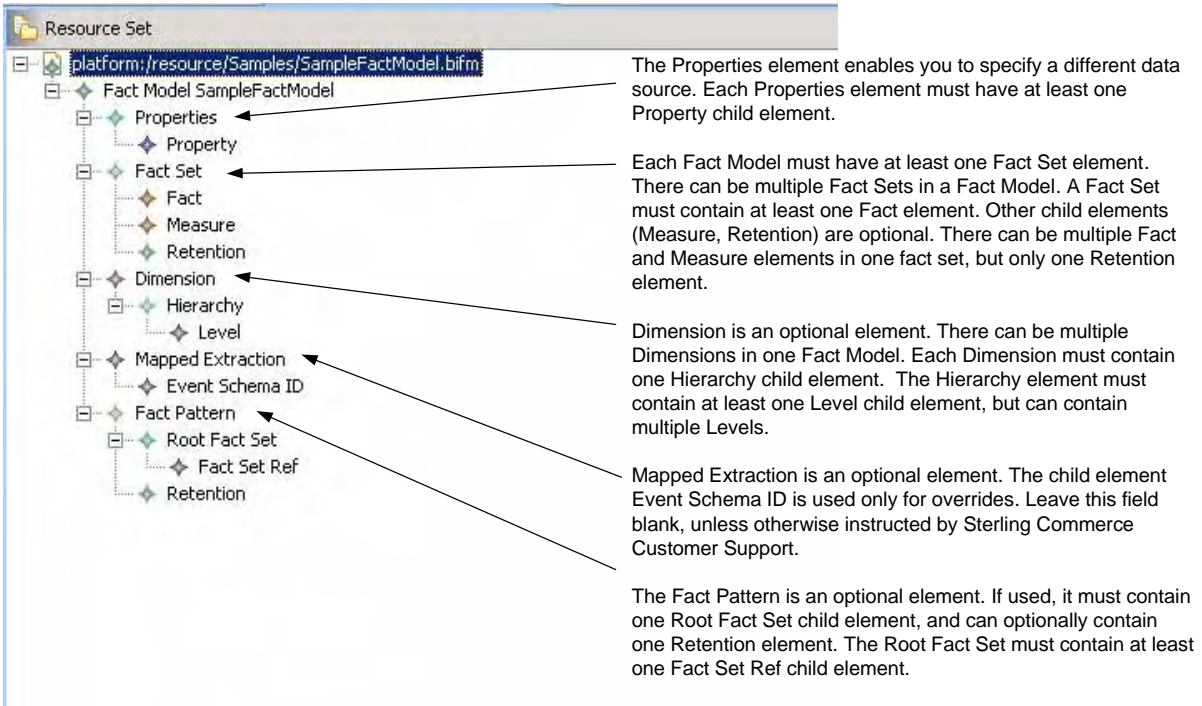
- ◆ stopEventListenersWindowsService.cmd
- ◆ uninstallEventListenersWindowsService.cmd

About the startlisteners.properties.in File

Reporting Services also uses the startlisteners.properties.in file, which is overwritten during patch installs, which can mean the loss of customized properties for starting event listeners. As a result, you may have to re-apply your custom properties and run the setupfiles script to generate the updated startlisteners.properties file.

Fact Model Components

The following graphic shows a fact model in the Reporting Services Fact Model Editor in Eclipse:



The following sections describe each element shown in the previous graphic:

Properties

Properties enable you to specify a different data source for your Reporting Services fact repository and can have the following child element:

Field	Description
Property	Enter the name of the data source to be used instead of the default. Required if Properties is used.

Properties Child Element: Property

The following table contains the fields that are included in a property element:

Field	Description
Name	Enter a name to be associated with this property in the fact model and schemas. For example, poolName.

Field	Description
Value	Enter the database to be used for the fact repository for this fact model.

Note: For more information about specifying a different database for Reporting Services, see *Reporting Services Database Considerations*.

Fact Set

A *fact set* is a group of logically categorized facts. A fact set can have the following child elements:

Field	Description
Fact	A single piece of raw data. At least one fact per fact set is required. A fact is the most basic, granular element of a fact model.
Measure	A numeric fact or result of a calculation.
Retention	Enables you to define the amount of time the data is kept in the Reporting Services repository.

Fact Set Child Element: Fact

The following table contains the fields that are included in a fact element:

Field	Description
Label	If you want to call this fact something other than the name in reports, enter a label. For example, the fact name is "BookTitle" and you want to save space on the reports. Enter "Title," as a label for this fact, and the label will be displayed as the column heading for this field on reports instead of "BookTitle." Optional.
Max Length	Default is 255. Valid values are 1-255. Used for string type only.
Name	This is the name that will be associated with the fact in the fact model and in the schemas created from it for reports, Mondrian cubes, and the fact model in Application. Required. Alphanumeric. Any constraints on naming for your database apply to naming fact sets and facts. Also, the names cannot contain spaces or special characters except underscore and hyphen (unless prohibited by your database). You cannot use keywords as fact names.

Field	Description
Type	Select a type from the list: <ul style="list-style-type: none"> ◆ String – Alphanumeric. Default type. ◆ Date – Format the date in the Report Editor ◆ Timestamp – Format the timestamp in the Report Editor ◆ Integer – Positive or negative number or zero ◆ Long – Long Integer field ◆ Float – Real number, single precision floating point ◆ Double – Real number, double precision floating point ◆ Boolean – Uses values of True and False.
Unique	Valid values are true and false. Defines whether there can be more than one occurrence of the fact name in the fact set. Default is false.

Fact Set Child Element: Measure

The following table contains the fields that are included in a measure element:

Field	Description
Aggregate	<p>You can have the Fact Model Editor perform the following operations on a fact within the same fact set:</p> <ul style="list-style-type: none"> ◆ sum – For numeric data types only. Calculates the total of the facts. ◆ count – For all data types. Counts total number of facts. ◆ min – For numeric data types only. Returns the minimum value included in the facts. ◆ max – For numeric data types only. Returns the minimum value included in the facts. ◆ avg – For numeric data types only. Calculates the average of the facts. <p>To use the Measure to aggregate data from one fact, select one of the aggregate types, then type the Fact name in the Fact Name field. Type a representative name for this measure in the Name field, such as “SumBooks.”</p> <p>Note: The fact used must be part of the same Fact Set as the measure. Sum, Count, Min, Max, Average.</p> <p>Optional.</p>
Fact Name	Enter the name of the fact to use.
Name	Enter a name to describe this measure, for example, “CountWidgets.”

Fact Set Child Element: Retention

The following table contains the fields that are included in a retention element:

Field	Description
Cascade	Deprecated. Leave at default.

Field	Description
Days	Enter the number of days to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Hours	Enter the number of hours to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Months	Enter the number of months to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Purge Time	Enter the time to start the purge in hours, minutes and seconds (24 hour clock).
Weeks	Enter the number of weeks to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Years	Enter the number of years to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.

Dimension

A dimension is a collection of data of the same type. Examples of dimensions are:

- ◆ Time (Year, Month, Day, or Hour, Minute, Second)
- ◆ Corporate structure (Division, Department, Functional Group; Sales Region, Sales Team, Salesperson)
- ◆ Location (Country, State, City, Street)

A dimension must have a hierarchy child element.

Field	Description
Hierarchy	Standard tree view (parent with one or many children) organization for positions in a dimension. Required if Dimension is used. Hierarchies can have level child elements.
Level	Describes the hierarchical structure of the dimension. Arrange in increasing levels of granularity. For example, a Time dimension and hierarchy might have the following levels: <ul style="list-style-type: none"> ◆ Yearly ◆ Monthly ◆ Weekly Required if Dimension and Hierarchy are used.

Dimension Child Element: Hierarchy

The following table contains the fields that are included in a hierarchy element:

Field	Description
HasAll	Leave at default. Used by Mondrian schema.

Field	Description
Name	This is the name that will be associated with the hierarchy in the fact model and in the schemas created from it for reports, Mondrian cubes, and the fact model in Application. Cannot contain spaces or special characters except hyphen and underscore. Required.

Hierarchy Child Element: Level

The following table contains the fields that are included in a level element:

Field	Description
Expression	Leave at default. Reserved for future use.
Fact Path	Enter the path to the fact defined by this level in the format: <i>fact set/fact</i>
Unique Member	Whether this is unique. Valid values are true and false. Default is true.

Mapped Extraction

A mapped extraction enables you to use a map to obtain Reporting Services data from a business process. You can specify one or more maps to use for each mapped extraction. If a mapped extraction uses multiple maps, the maps must all be the same type: either any-to-XML or any-to-JDBC. You cannot use both types in one mapped extraction. However, a fact model can have more than one mapped extraction, so different map types can be used as parts of different mapped extractions.

Field	Description
Map Name	Click the ellipsis (...). On the window that is displayed, enter the name of a map to be used for this mapped extraction in the Value field. Click Add to move the map to the list on the right. Add any additional maps, then use the Up and Down buttons to put the maps into the correct order. The map to be used first should be at the top of the list; the map to be used last at the end of the list. Note: If using the passThrough option, leave this field blank.
Name	Enter a unique name for this mapped extraction. You will enter this name as a parameter for the Mapped Extract service when creating the business process. If your data is already in the correct format for the fact repository, you can enter passThrough. Note: The .xsd file created for the fact model by Application represents the correct format.
Output Type	Select an option: <ul style="list-style-type: none"> ◆ If mapping to JDBC or using passThrough, select DB (default) ◆ If mapping to XML, select XML

Mapped Extraction Child Element: Event Schema ID

A Mapped Extraction can have the following child element:

Field	Description
Event Schema ID	Identifies the event that invokes a mapped extraction. This field is automatically assigned by Application when the fact model is checked in, and should be left blank in the fact model. Currently, this is for internal use only.

Fact Pattern

A fact pattern is a description of the logical construct for a fact set. A fact pattern is limited to a single parent fact that may have multiple children.

The fact pattern element itself has no fields.

The fact pattern is optional and can have the following child elements:

Field	Description
Retention	Time to store fact pattern data in the repository before purging.
Root Fact Set	The fact set that is the primary information that ties all other information in the fact model together.

Fact Pattern Child Element: Root Fact Set

The following table contains the fields that are included in a root fact set element:

Field	Description
Key Fact	Links the parent fact set to the child fact set.
Path	Enter the path to the root fact set. Required if fact pattern is used.

Root Fact Set Child Element: Fact Set Ref

The following table contains the field that is included in a fact set ref element:

Field	Description
Path	Enter the fact set associated with this fact set ref. Required.

Fact Pattern Child Element: Retention

The following table contains the fields that are included in a retention element for a fact pattern:

Field	Description
Cascade	Deprecated. Leave at default.

Field	Description
Days	Enter the number of days to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Hours	Enter the number of hours to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Months	Enter the number of months to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Purge Time	Enter the time to start the purge in hours, minutes and seconds (24 hour clock).
Weeks	Enter the number of weeks to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.
Years	Enter the number of years to retain the information in this fact set in the Reporting Services repository. Optional. Valid values are 0-2,147,483,648.

Fact Model Naming Considerations

Because the fact set name ultimately becomes the name of a table in the fact repository, and the default fact repository is the Application database, it is possible to create fact set names that are not supported in Application. For example:

- ◆ Giving a fact set a name that happens to be the same as a Application table adds extra columns to the table. This can destroy the Application system's ability to run.
- ◆ Naming a fact set the same as one of the columns in the database table causes data to start persisting in the table, causing issues in Application.

Do not use Application table names as fact set names. You can check the database repository to verify which names are used, however this is not a failsafe, because future Application patches may collide with your fact sets. Therefore, best practice is to have a separate Reporting Services repository outside of the Gentran Integration Suite database. A separate fact repository enables you to name fact sets as you please.

Checking In a Fact Model

Once you have completed your fact model in the Fact Model Editor, you must check in the fact model to Application. Checking in a fact model creates schemas for the default detail and summary reports, and Mondrian cube schema. These are the keys to viewing your data.

Note: The first time you check in a fact model, it also creates a fact repository for the fact model.

Checking in a Fact Model for the First Time

1. From the Application interface, select Deployment > Schemas.
2. Select Check in new XML Schema or DTD.
3. Enter the path and fact model filename, or browse to the file.
4. Enter check in comments, and click Next.
5. The default is the name of the fact model file.
6. Select a schema type (in this case, select Reporting Services Fact Model). Click Next, then Finish.

Note: The first time you check in a new fact model, the check in process may take a few minutes.

Editing a Fact Model

You edit fact models in the Fact Model Editor Eclipse plug-in. Before doing so, you must check out the fact model from Application.

1. From the Application interface, select Deployment > Schemas.
2. In the List pane, under By Schema Type, select Fact Model.
3. On the results page, click Source Manager next to the fact model to be checked out.
4. On the Source Manager page, click check out next to the desired version of the fact model.
5. Type or browse to the path on your system where the fact model should be saved.

Checking in an Edited Fact Model

1. From the Application interface, select Deployment > Schemas.
2. In the List pane, under By Schema Type, select Fact Model.
3. On the results page, click Source Manager next to the fact model to be checked in.
4. On the Source Manager page, click Go! next to Check in a new version of this schema.
5. Type or browse to the file on your system and click OK.
6. On the Set Default Version page, select the new version as the default.
7. On the Confirm page, click Finish.

8. Restart the BI listener JVM after you check in an updated version of a fact model. For information on restarting a listener, see *Starting and Stopping the Reporting Services Listeners* on page 35.

Changing the Default Version of a Resource

1. From the Application interface, select Deployment > Schemas.
2. In the List pane, under By Schema Type, select the type of resource: Fact Model, Report Definition or Report Format.
3. On the results page, click Version Manager next to the desired resource.
4. On the Version Manager page, in the Default column, select the desired version of the resource and click Save.

Viewing Your Reporting Services Data

In Reporting Services, there are three tasks that must be performed to view data:

- ◆ Creating the schemas for the fact model and its components (done by checking in the fact model to Application)
- ◆ Editing and creating reports (done after you view the default reports)
- ◆ Viewing reports or using the Mondrian cube to view data (done in JPivot outside of Application)

Default Reports

Once you have checked in the fact model, you have access to the default details reports that were automatically created for the fact model (and summary reports for measures, if used).

When the fact model is checked in, Application creates detail and summary reports for each set of linked fact sets within a fact model. It also creates detail and summary reports for each fact set that is not linked to any other fact set.

In the reports, a group is created for each level in the fact model.

Reporting Services Properties File

Application includes a properties file called `bi.properties` for Reporting Services. It is located with the rest of the Application properties files, in the `[install_dir]/properties` folder on your Application system.

The following table includes the Reporting Services properties that you may need to change:

Caution: Do not change other properties in this file unless told to do so by Sterling Commerce Customer Support.

Property	Description
<code>maxRowSize=<Value></code>	The maximum table row length for the fact repository database. Differs depending on the database. See <i>Reporting Services Database Considerations</i> . Can be changed to use a higher value if needed.
<code>poolName=mysqlPool</code>	Database pool to be used for fact repository. Can be changed. See <i>Reporting Services Database Considerations</i> .
<code>reporting.backupResources=true</code>	If set to true, backup copies of the report and language properties files are generated each time you check in one of the versioned resources (fact models, report definitions, report formats). The backup files are preserved on the file system with an extension of <code>bak.n</code> , where <code>n</code> is the number of the backup for that file. For example, the following backup copies might exist: <code>report.xml.bak.1</code> , <code>report.xml.bak.2</code> , <code>report.xml.bak.3</code> .
<code>reporting.maxdetailrowcount=5000</code>	If running a Reporting Services report is causing an “out of memory” error, change this parameter to use a lower value.

There are two `bi.properties` files:

- ◆ `bi.properties.in` – This is the template file and contains all possible parameters
- ◆ `bi.properties` – Once you run the `setupfiles` command, this file is populated with the information specific to your system. Do not make changes directly to this file – it is overwritten during updates and any time that the `setupfiles` command is run.

Instead of directly changing either `bi.properties` file, make changes by adding entries to the `customer_overrides.properties` file. For example:

```
bi.poolName=myPool
```

You will need to create the `customer_overrides.properties` file if this is the first property override done for this installation. The file must be located in the `[install_dir]/properties` folder and must be named `customer_overrides.properties`. You can create the file using a text editor. See *Overriding Property File Settings* for more information.

Caution: Reporting Services also uses the `startlisteners.properties.in` file, which is overwritten during patch installs, which can mean the loss of customized properties for starting event listeners. As a

result, you may have to re-apply your custom properties and run the setupfiles script to generate the updated startlisteners.properties file.

C

customizing reports 31

D

data visualization 46

default reports 46

dimension

 creating in fact model 20

 defined 40

 field descriptions 40

 hierarchy field descriptions 40

 level field descriptions 41

E

Eclipse

 accessing Reporting Services 7

 creating a project 20

F

fact

 field descriptions 38

fact model

 building 20

 check in through UI 44

 check-in 44

 components 37

 creating 20

 creating an Eclipse project 20

 naming considerations 43

fact pattern

 creating in fact model 21

 defined 42

 fact set ref field descriptions 42

 field descriptions 42

 retention field descriptions 42

 root fact set field descriptions 42

fact repository 7

fact set

 creating in fact model 20

 defined 38

 field descriptions 38

fact set ref

 field descriptions 42

fact, defined 18

H

hierarchy

 field descriptions 40

 level field descriptions 41

L

level

 field descriptions 41

M

mapped extraction

 creating in fact model 21

 defined 26, 41

 field descriptions 26, 41

measure

 field descriptions 39

Mondrian

 cube 46

N

naming fact models 43

P

project

 creating in Eclipse 20

properties

 creating in fact model 21

 defined 37

field descriptions 37

R

Report Editor

creating reports 29
editing reports 29

Report Format Editor

customizing reports 31

report formats

creating 31
editing 31

Reporting Services

components 7
configuring 5, 11
creating a fact model 20
creating report formats 31
creating reports 29
defined 7
dimension 20
editing reports 29
fact model components 37
fact pattern 21
fact repository 7
fact set 20
installing 5, 11
mapped extraction 21
properties 21
security considerations 8

reports

creating 29
customizing appearance 31
default 46
editing 29

retention

field descriptions 39, 42

role based security 8

root fact set

fact set ref field descriptions 42
field descriptions 42

S

security in Reporting Services 8

V

viewing data 46

viewing results 46