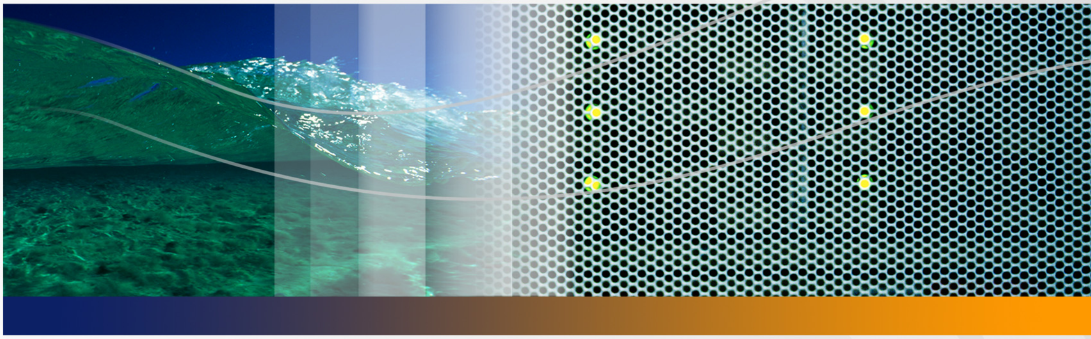
Product Documentation

Peregrine Systems, Inc.

Reporting Data Store 5.2 for ServiceCenter



Administration Guide

Part No. DBNI-52-EN05

The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document. The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental. If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com. If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com. This edition of the document applies to version 5.2 of the licensed program.

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com

Contents

PEREGRINE

About this Guide 7

Using this Guide 7

Related Documentation 8

Typographical conventions. 9

Special elements. 9

Need further assistance? 10

Customer Support 10

Documentation Web site 10

Education Services Web site 11

Chapter 1 Reporting Data Store and Schema 13

What is the Reporting Data Store (RDS) 13

RDS design overview. 14

RDS naming conventions. 16

RDS metadata overview 16

RDS schema 16

RDS ETL mapping information 17

RDS semantic layer — universe 19

RDS schema file description 20

Dimension table schema definition 21

Fact table schema definition 23

Associate table schema definition 25

Hierarchy table schema definition 26

Direct mapping table schema definition 27

Slowly changing dimension (SCD) keys schema definition 29

Aggregate keys schema definition 30

RDS dimension attributes 31

RDS InitRun command 32

Synchronizing ServiceCenter record deletions with RDS 32

Chapter 2 Using Connect-It to Handle RDS Data Synchronization 35

RDS data synchronization process 35

Starting and stopping the RDS scenarios 36

Configuring the connectors 39

Chapter 3 Customization Workflow 45

Workflow overview 45

Overview: Planning and information gathering 46

Overview: Stopping programs and processes 48

Overview: Backing up crucial files. 48

Overview: Editing the RDS schema 49

Overview: Editing the Connect-It scenario 50

Overview: Editing the universe 50

Overview: Editing security profiles 51

Overview: Restarting programs and processes 51

Chapter 4 RDS Customization 53

RDS schema customization. 53

Adding a new table 54

Adding a new field 57

Changing an existing field 59

ServiceCenter trigger customization 60

Predefined ServiceCenter triggers 60

Add and remove triggers for a user-defined ServiceCenter table . 62

Chapter 5 Connect-It Scenario Customization 65

Using CIT Scenario Builder tools to view an RDS scenario 66

Adding new mapping 67

Adding mapping for a new table 68

Adding a new field to an existing mapping 78

Test your customized RDS scenario 82

Update existing mapping 82

Scheduling automatic data synchronization. 87

Chapter 6 Business Objects Universe Customization 91

Editing the universe 91

Adding a new table 91

Adding a new field 95

Modifying an existing field 100

Checking database integrity 101

Exporting the universe 102

Editing security profiles 103

Appendix A RDS System Tables. 109

Tracking RDS schema metadata 109

RDS_UNIQUECOLUMN table 109

RDS_SCDCOLUMN table 110

RDS environment tables 111

RDS_DBINFO table. 111

RDS Security System Table - RDS_USER 111

RDS System tables for tracking ETL processing 112

RDS_CIT_LOG 112

RDS_SEC_SYNC table 113

RDS_SEQUENCE table 113

RDS_ETLSYNC_LOG table 114

Other useful tables. 114

RDS time dimension table 114

Appendix B Copyright Notices 117

Notices 117

Index 127

About this Guide

PEREGRINE

This document is for advanced users who are going to customize the Reporting Data Store (RDS). If you intend to customize the `RDS_etl.xml` file, contact Peregrine Educational Services about BI Portal training and engage Peregrine Professional Services for additional assistance. This Reporting Data Store Administration Guide provides information about customizing the RDS.

Using this Guide

This guide includes the following chapters.

This chapter	Description
Chapter 1, Reporting Data Store and Schema	Provides an overview of the Reporting Data Store (RDS).
Chapter 2, Using Connect-It to Handle RDS Data Synchronization	Explains how to utilize the Connect-It tools to perform RDS data synchronization and how to tailor the scenario for your deployment.
Chapter 3, Customization Workflow	Discusses the recommended workflow for creating useful and effective RDS customizations.
Chapter 4, RDS Customization	Discusses RDS schema customization and ServiceCenter triggers customization.
Chapter 5, Connect-It Scenario Customization	Discusses how to customize the RDS Connect-It scenario, `rds_sc.scn`, for ServiceCenter.
Chapter 6, Business Objects Universe Customization	Explains how to use the Business Objects Designer tool to customize a universe file.
Appendix A, RDS System Tables	Provides descriptions and samples of the system tables founds in the RDS.
Appendix B, Copyright Notices	Contains additional copyright information.

Prior to using this guide, you should read the following sections:

- Related Documentation
- Typographical conventions
- Need further assistance?

Related Documentation

In addition to this guide, the following documentation is available for the Reporting Data Store product and for WebIntelligence. Unless otherwise noted, the documentation is available at *http://support.peregrine.com*.

Manual	Description
BI Portal User Guide	Provides base reports and describes how to create and work with both base and custom reports.
BI Portal Installation Guide	Describes how to install and configure the application and Web servers for Reporting Data Store.
BI Portal Administration Guide	Provides an overview of the OAA platform. Describes how you customize the BI Portal and the OAA Administration module. Provides information about security in BI Portal, RDS Universe administration, data connection and transfer, configuring components after a custom installation, and performance tuning.
RDS for AssetCenter Administration Guide	Provides information about customizing the RDS for AssetCenter.
BI Portal Release Notes	Includes last-minute enhancements, known issues, and closed issues.
WebIntelligence User's Guide	Describes how to use WebIntelligence for building and running queries, reporting, and analysis. This is available using the Help button of the WebIntelligence Java Report Panel, which is accessed when creating or editing reports.

Typographical conventions

This guide uses typeface conventions to indicate special terms and actions.

Convention	Meaning
Bold	Information that you must type exactly as shown appears in bold. The names of buttons, menus, and menu options also appear in **bold**.
Italics	Variables and values that you must provide are in *italics*. New terms and book titles also are in *italics*.

`Monospace`	Code or script examples, output, and system messages are in a `monospace` font.
	`var msgTicket = new Message( "Problem" );`
	...
	`msgTicket.set( "_event", "epmc" );`
	An ellipsis (…) indicates that portions of a script have been omitted because they are not needed for the current topic. Samples of code are not entire files, but they are representative of the information discussed in a particular section.
	Filenames, such as `login.asp`, appear in a `monospace` font.

Special elements

This guide uses special elements to help you locate information. These special elements and their uses are in the following table.

Element	Usage
Important:	Information that is required to complete a task
Note:	Information that is of general interest
Tip:	Information that can make a task easier or faster
Warning:	Information that is needed when there is a risk of losing data

Need further assistance?

For further information and assistance with this release, you can download documentation or schedule training.

Customer Support

For further information and assistance, contact Peregrine Systems' Customer Support at the Peregrine CenterPoint Web site.

To contact customer support:

1 In a browser, navigate to *http://support.peregrine.com*

2 Log in with your user name and password.

3 Follow the directions on the site to find your answer. The first place to search is the KnowledgeBase, which contains informational articles about all categories of Peregrine products.

4 If the KnowledgeBase does not contain an article that addresses your concerns, you can search for information by product; search discussion forums; and search for product downloads.

Documentation Web site

For a complete listing of current Reporting Data Store documentation, see the Documentation pages on the Peregrine Customer Support Web.

To view the document listing:

1 In a browser, navigate to *http://support.peregrine.com*.

2 Log in with your login user name and password.

3 Click either Documentation or Release Notes at the top of the page.

4 Click the Reporting Data Store link.

5 Click a product version link to display a list of documents that are available for that version of Reporting Data Store.

6 Documents may be available in multiple languages. Click the Download button to download the PDF file in the language you prefer.

You can view PDF files using Acrobat Reader, which is available on the Customer Support Web site and through Adobe at *http://www.adobe.com*.

Important: Release Notes for this product are continually updated after each release of the product. Ensure that you have the most current version of the Release Notes.

Education Services Web site

Peregrine Systems offers classroom training anywhere in the world, as well as “at-your-desk” training using the Internet. For a complete listing of Peregrine’s training courses, refer to the following web site:

http://www.peregrine.com/education

You can also call Peregrine Education Services at +1 858.794.5009.

1

CHAPTER

Reporting Data Store and Schema

This chapter provides an overview of the Reporting Data Store (RDS) and serves as an introduction to some of the RDS design concepts you need to be familiar with if you customize the out-of-box RDS used with BI Portal. This chapter discusses the following topics:

- What is the Reporting Data Store (RDS) on page 13
- RDS design overview on page 14
- RDS metadata overview on page 16
- RDS schema file description on page 20
- Synchronizing ServiceCenter record deletions with RDS on page 32

Note: This section provides an introduction to the Reporting Data Store (RDS). If you customize the `RDS_etl.xml` file, contact Peregrine Educational Services about BI Portal training and consult Peregrine Professional Services for additional assistance.

What is the Reporting Data Store (RDS)

The RDS is a repository of integrated information that is available for queries and analysis and includes Extract, Transform and Load (ETL) processes and metadata.

The RDS provides the following advantages:

- Relieves reporting pressure on the database.
- Enables administrators to restructure data to speed up data analysis and reporting capabilities.
- Reduces the difficulty users have with generating new reports.
- Enables administrators to create *clean data* that does not require wholesale changes to the ServiceCenter (SC) or AssetCenter (AC) transactional system or business processes.
- Enables hierarchy analysis.
- Allows historical analysis by providing a data source supporting a longer span of time.

RDS design overview

The RDS uses Dimensional Modeling to present data in a standard framework that is intuitive and allows for high performance access. It is extendable and provides easy management of aggregates.

The RDS defines tables based on the following categories:

- Dimension tables
- Fact tables
- Associate or join tables and bridge tables
- Direct mapping tables from ServiceCenter transaction tables
- Aggregates of fact tables
- Hierarchy tables
- RDS system tables

Dimension tables

Most dimension table fields are directly mapped from ServiceCenter tables through Connect-It. Dimension tables add additional RDS ETL (Extract, Transform and Loading) attributes. ServiceCenter tables should be considered as possible dimension tables if they include historical, reserved data or time associated data or the table is used as search criteria for multiple ServiceCenter reporting modules.

Each dimension table has the system key, Z_RDS(tablename)ID. The surrogate keys allow the data warehouse to assign a new key version for Slowly Changing Dimensions and allow the data warehouse to encode uncertain, not known, not recorded, and null record types. Every join between dimension and fact tables should be based on surrogate keys.

Fact tables

Fact tables are the RDS tables created for measurements for the associated dimension tables used by the reporting modules.

Associate or join tables and bridge tables

Associate or bridge tables created in the RDS are used for sub-dimension tables or solving many-to-many relationships.

Direct mapping tables from ServiceCenter transaction tables

BI Portal uses ServiceCenter transaction tables for detail reporting and as search criteria for one ServiceCenterapplication. These tables are directly mapped from ServiceCenter tables into the RDS.

Aggregates of fact tables

Measures grouped by common sort fields are pre-aggregated to allow faster summarization of data.

Hierarchy tables

Hierarchy tables contain fields to create parent-child relationships between records. For example, manager is the parentfieldname in the *Contact* table. The maximum hierarchy level is 10.

RDS system tables

In the RDS, there are system tables defined to track RDS schema metadata, RDS user security, the ETL process status and time dimension. All the system table names begin with RDS_; the only exception is BIRECORDDELETE.

RDS naming conventions

The table naming conventions are:

- Dimension table names end with `_D` and have a maximum length of 18 characters. The primary key for dimension table is `Z_RDSXXX_DID`. Sample name is `XXX_D`.
- The dimension table names use the first 15 characters from original source table names without the `S`.
- Fact table names end with `_F`.
- Do not use count character for measurement names for fact tables.
- Aggregate table names end with `_AGG`.
- Associate or bridge table names use the first eight characters from the two associated table names with an underscore (_) as the separator.
- All the direct mapping table names use the same name as the source name.
- Convert dots (.) in ServiceCenter field names to underscores for RDS field names. For example, convert `parent.name` to `parent_name`.
- Direct Mapping Indexes length can be no longer than 18 characters.
- The RDS system table names start with `RDS_`.

RDS metadata overview

The RDS metadata consists of the RDS schema, ETL process and RDS semantic layer, and the RDS universe.

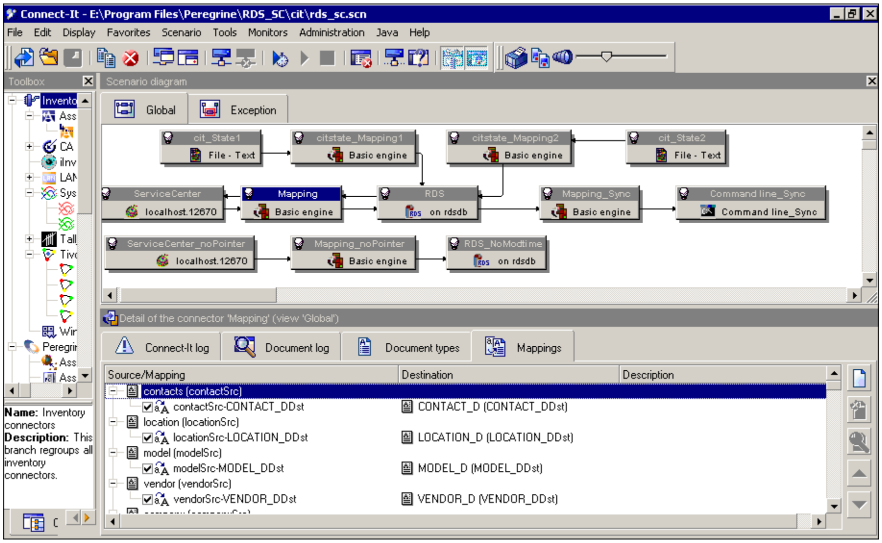
RDS schema

The `RDS_etl.xml` provides the RDS schema data. The `rds_etl.xsd` defines the XML schema syntax. All these files are stored in `/conf` folder of the RDS installation directory.

RDS ETL mapping information

The Connect-It RDS scenario (`rds_sc.scn`) describes the metadata for the extraction, transformation and mapping. You can launch Connect-It Scenario Builder to open `rds_sc.scn` from the `/cit` folder of the RDS installation directory. You can use the Edit function to view any Source/Mapping and review the ETL mapping information.

The following diagram shows the mapping for the *contacts* table.

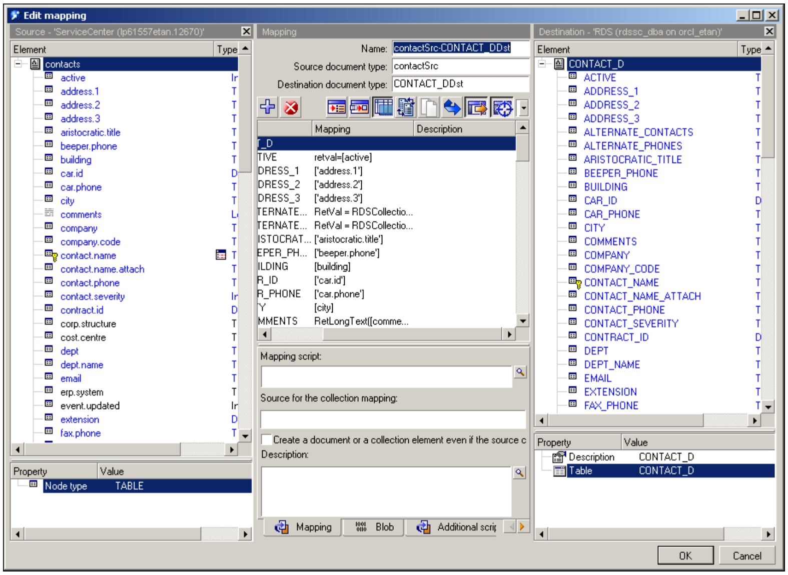


Name: Inventory connectors
Description: This branch regroups all inventory connectors.

Source/Mapping	Destination	Description
contacts (contactSrc)		
☑ contactSrc-CONTACT_DDst	CONTACT_D (CONTACT_DDst)	
location (locationSrc)		
☑ locationSrc-LOCATION_DDst	LOCATION_D (LOCATION_DDst)	
model (modelSrc)		
☑ modelSrc-MODEL_DDst	MODEL_D (MODEL_DDst)	
vendor (vendorSrc)		
☑ vendorSrc-VENDOR_DDst	VENDOR_D (VENDOR_DDst)	

After you click **Edit a mapping**, the Edit Mapping window opens.

The ServiceCenter data source appears in the left pane and the right pane shows the RDS destination table. The middle pane defines the mapping script for each attribute. The Connect-It User Guide provides more information about how to use this tool.



Source - 'ServiceCenter (lp61557etan.12670)'

Element	Type
contacts	
active	Ir
address.1	T
address.2	T
address.3	T
aristocratic.title	T
beeper.phone	T
building	T
car.id	D
car.phone	T
city	T
comments	L
company	T
company.code	T
contact.name	T
contact.name.attach	T
contact.phone	T
contact.severity	Ir
contract.id	D
corp.structure	T
cost.centre	T
dept	T
dept.name	T
email	T
erp.system	T
event.updated	Ir
extension	D
fax.phone	T

Property	Value
Node type	TABLE

Mapping

Name: contactSrc-CONTACT_DDst

Source document type: contactSrc

Destination document type: CONTACT_DDst

	Mapping	Description
T_D		
TIVE	retval=[active]	
DRESS_1	['address.1']	
DRESS_2	['address.2']	
DRESS_3	['address.3']	
TERNATE...	RetVal = RDSCollectio...	
TERNATE...	RetVal = RDSCollectio...	
ISTOCRAT...	['aristocratic.title']	
EPER_PH...	['beeper.phone']	
ILDING	[building]	
R_ID	['car.id']	
R_PHONE	['car.phone']	
Y	[city]	
MMENTS	RetLongText([comme...	

Mapping script:

Source for the collection mapping:

☐ Create a document or a collection element even if the source c

Description:

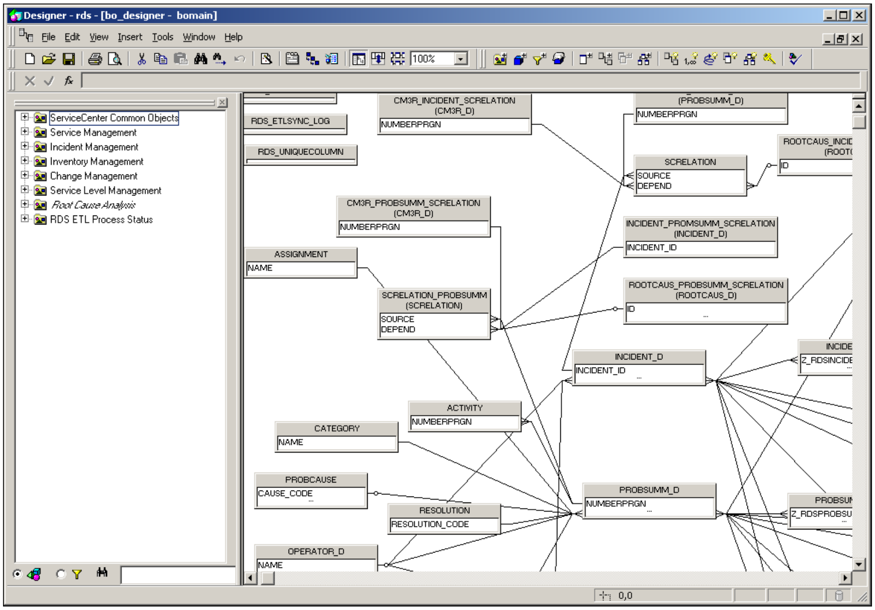
Destination - 'RDS (rdssc_dba on orcl_etan)'

Element	Type
CONTACT_D	
ACTIVE	T
ADDRESS_1	T
ADDRESS_2	T
ADDRESS_3	T
ALTERNATE_CONTACTS	T
ALTERNATE_PHONES	T
ARISTOCRATIC_TITLE	T
BEEPER_PHONE	T
BUILDING	T
CAR_ID	D
CAR_PHONE	T
CITY	T
COMMENTS	T
COMPANY	T
COMPANY_CODE	T
CONTACT_NAME	T
CONTACT_NAME_ATTACH	T
CONTACT_PHONE	T
CONTACT_SEVERITY	T
CONTRACT_ID	D
DEPT	T
DEPT_NAME	T
EMAIL	T
EXTENSION	T
FAX_PHONE	T

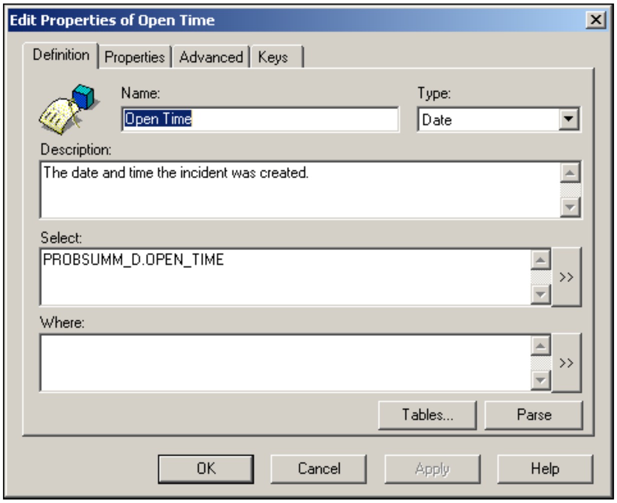
Property	Value
Description	CONTACT_D
Table	CONTACT_D

RDS semantic layer — universe

Use the Business Objects (BO) designer tool to develop the RDS universe. The diagram below shows the RDS universe displayed in the designer tool.



You can use Business Objects designer tool to view the object definitions, which you can use to determine how to map the fields of RDS tables. To open an object's properties, select the section. The following figure shows an example of how the tools display the properties.



Name: Open Time

Type: Date

Description: The date and time the incident was created.

Select: PROBSUMM_D.OPEN_TIME

Where:

You can use the Business Objects designer tool to produce a PDF file of the universe metadata. The `SupportFiles\ServiceCenter\universe` folder on the RDS for ServiceCenter installation CD contains the `rds.unv` file. When you open this file using the Business Objects designer tool, you can save it as a PDF document.

RDS schema file description

The RDS database is created based on the schema defined in the `RDS_etl.xml` file. This schema is built based on the XML schema definition defined in `rds_etl.xsd`. These two files normally reside in the directory, `/conf`, under the RDS installation root directory. The detail of the `rds_etl.xsd` is included in the appendix.

The following definitions organize the RDS schema in sequential order:

- A list of dimension table definitions
- A list of associate table definitions
- A list of hierarchy table definitions
- A list of direct mapping table definitions

The RDS schema uses the following XML syntax.

```
<rds>
   <dimensions/>
   <associates/>
   <hierarchies/>
   <directmappings/>
</rds>
```

The following sections list the XML schema definition and provide examples each for the table types.

Dimension table schema definition

Dimension tables are grouped under a root element, `dimensions`. You can have one or more dimension table definitions under this root element. Each dimension table definition is tagged under an element name, `dimension`. In general, a dimension table definition contains the following information.

A dimension table name

Required elements

- A data source table name
- A list of field definitions
- A list of unique key definitions

Optional elements

- A list of fact table definitions
- A list of SCD key definitions
- A list of aggregate key definitions

For an illustration of a detailed XML schema for this table type, examine the definition section for a complex type of DimensionType in `rds_etl.xsd`.

The following example shows the RDS schema definition for the device table in ServiceCenter.

```
<!--  Device inventory dimension ETL definition  -->
    <dimension name="DEVICE" >
      <dimensionTableName>DEVICE_D</dimensionTableName>
      <dataSourceTableName>device</dataSourceTableName>
      <dimensionTableFields>
        <dimensionTableField name="logical_name" type="char" size="60"/>
        <dimensionTableField name="vendor" type="char" size="60"/>

        . . .
        <dimensionTableField name="comments" type="long"/>
        <!-- !!! SC 5.1 renamed attributes -->
        <dimensionTableField name="cost_center" type="char" size="60"/>
        <!-- !!! SC 5.1 new attributes -->
        <dimensionTableField name="assignment" type="char" size="60"/>

        <dimensionTableField name="port_desc" type="long"/>
        . . .
      </dimensionTableFields>
      <uniqueKeys name="device_unique">
        <uniqueKey fieldName="logical_name"
          srcFieldName="''logical.name''"
         srctype="char" size="60" defaultValue="no match"/>
      </uniqueKeys>
      <facts>
        <fact name="DEVICE_F">
          <factKeys>
             <factKey name="Z_RDSDEVICE_DID"/>
             <factKey name="Z_RDSCONTACT_DID" fieldName="CONTACT_NAME"
         tableName="CONTACT_D" matchFieldName="CONTACT_NAME"/>
             <factKey name="Z_RDSLOCATION_DID" fieldName="LOCATION"
         tableName="LOCATION_D" matchFieldName="LOCATION"/>
             <factKey name="Z_RDSMODEL_DID" fieldName="part_no"
         tableName="MODEL_D" matchFieldName="part_no"/>
             <factKey name="Z_RDSVENDOR_DID" fieldName="VENDOR"
         tableName="VENDOR_D" matchFieldName="VENDOR"/>
             <factKey name="z_rdscompany_did" fieldName="COMPANY"
         tableName="COMPANY_D" matchFieldName="COMPANY"/>
             <factKey name="z_rdsdept_did" fieldName="dept"
               tableName="DEPT_D"
         matchFieldName="DEPT"/>
             <factKey name="install_dateID" fieldName="install_date"
         tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
             <factKey name="last_scanID" fieldName="last_scan"
         tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
          </factKeys>
          <factMeasures>
```

```
<!-- !!! Don't use count in measurement names if it is possible, because
         BO E6 bugs !!! -->
             <factMeasure name="pcount" srcType="src"/>
             <factMeasure name="total_downtime" srcType="src"/>
             <factMeasure name="breaks" srcType="src"/>
             <factMeasure name="DEVICE_number" srcType="rdsfactless"/>
             <factMeasure name="DEVICE_SCAN" srcType="rds"
         sqlConditions="LAST_SCAN is not null" resetCount=""/>
          </factMeasures>
        </fact>
      </facts>
      <scdKeys>
        <scdKey rdsFieldName="network_name"
         srcFieldName="''network.name''"
         srctype="char"/>
        <scdKey rdsFieldName="domain" srcFieldName="domain"
          srctype="char"/>
        <scdKey rdsFieldName="contact_name"
          srcFieldName="''contact.name''"
         srctype="char"/>
      </scdKeys>
      <aggregateKeys>
        <aggregateKey name="type" type="char" size="60"/>
        <aggregateKey name="subtype" type="char" size="60"/>
        <aggregateKey name="Istatus" type="char" size="60"/>
        <aggregateKey name="vendor" type="char" size="60"/>
        <aggregateKey name="install_date" type="TIME"/>
      </aggregateKeys>
    </dimension>
```

Fact table schema definition

Fact tables are defined under a root element, `facts`. Under this root element, one or more fact tables can be defined. Each fact table is tagged by an element name, `fact`.

The following shows the XML schema definition for the `FactType` table type.

```
<xsd:complexType name="FactType">
     <xsd:sequence>
     <xsd:element name="factKeys">
          <xsd:complexType>
              <xsd:sequence>
      <xsd:element name="factKey" type="FactKeyType" minOccurs="1"
                               maxOccurs="unbounded"/>
              </xsd:sequence>
          </xsd:complexType>
     </xsd:element>
     <xsd:element name="factMeasures">
       <xsd:complexType>
          <xsd:sequence>
              <xsd:element name="factMeasure" type="FactMeasureType"
                minOccurs="1"
                      maxOccurs="unbounded"/>
               </xsd:sequence>
      </xsd:complexType>
     </xsd:element>
        </xsd:sequence>
<!-- the name has to be same as dimension name before '_' -->
   <xsd:attribute name="name" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="FactKeyType">
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="fieldName" type="xsd:string"/>
   <xsd:attribute name="tableName" type="xsd:string"/>
   <xsd:attribute name="matchFieldName" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="FactMeasureType">
      <!-- Don't use count in measurement names, because BO E6 bugs  -->
      <!-- only ONE rdsfactless measurement, ex open ticket count -->
      <!-- only ONE rds measurement with resetCount for rdsfactless not
            blank, ex close
      ticket count -->
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="srcType" type="xsd:string" use="required"/>
   <xsd:attribute name="sqlConditions" type="xsd:string"/>
   <xsd:attribute name="resetCount" type="xsd:string"/>
</xsd:complexType>
```

The following is an example of a fact table definition in dimension table DEVICE.

```
<facts>
        <fact name="DEVICE_F">
          <factKeys>
             <factKey name="Z_RDSDEVICE_DID"/>
             <factKey name="Z_RDSCONTACT_DID" fieldName="CONTACT_NAME"
         tableName="CONTACT_D" matchFieldName="CONTACT_NAME"/>
             <factKey name="Z_RDSLOCATION_DID" fieldName="LOCATION"
         tableName="LOCATION_D" matchFieldName="LOCATION"/>
             <factKey name="Z_RDSMODEL_DID" fieldName="part_no"
         tableName="MODEL_D" matchFieldName="part_no"/>
             <factKey name="Z_RDSVENDOR_DID" fieldName="VENDOR"
         tableName="VENDOR_D" matchFieldName="VENDOR"/>
             <factKey name="z_rdscompany_did" fieldName="COMPANY"
         tableName="COMPANY_D" matchFieldName="COMPANY"/>
             <factKey name="z_rdsdept_did" fieldName="dept"
               tableName="DEPT_D"
         matchFieldName="DEPT"/>
             <factKey name="install_dateID" fieldName="install_date"
         tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
             <factKey name="last_scanID" fieldName="last_scan"
         tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
          </factKeys>
          <factMeasures>
             <!-- !!! Don't use count in measurement names if it is
                       possible, because BO E6 bugs !!! -->
             <factMeasure name="pcount" srcType="src"/>
             <factMeasure name="total_downtime" srcType="src"/>
             <factMeasure name="breaks" srcType="src"/>
             <factMeasure name="DEVICE_number" srcType="rdsfactless"/>
             <factMeasure name="DEVICE_SCAN" srcType="rds"
         sqlConditions="LAST_SCAN is not null" resetCount=""/>
          </factMeasures>
        </fact>
</facts>
```

Associate table schema definition

Associate tables are defined under a root element, associates. Each associate table is tagged by an element, associate. Under this root element there can be one or more associate table definition. An associate table contains the following attributes.

Required attributes	Name of the table Table name and field name for table 1 of the association relationship Table name and field name for table 2 of the association relationship
Optional attributes	Name of the long field name RDS version string

The following shows the XML schema definition for an associate table.

```
   <xsd:complexType name="AssociateType">
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="rdsVersion" type="xsd:string" use="optional"/>
      <xsd:attribute name="tableName" type="xsd:string" use="required"/>
      <xsd:attribute name="fieldName" type="xsd:string" use="required"/>
      <xsd:attribute name="longFieldName" type="xsd:string"/>
      <xsd:attribute name="tableName2" type="xsd:string" use="required"/>
      <xsd:attribute name="fieldName2" type="xsd:string" use="required"/>
   </xsd:complexType>
```

The following is an example of an associate table definition in the RDS schema for the ServiceCenter *device* table.

```
<associates>
     <associate name="INCIDENT_MODEL_ASS" rdsVersion="5.0"
      tableName="INCIDENT"
      fieldName="MODEL" tableName2="MODEL" fieldName2="MODEL"/>
     <associate name="PROBSUMM_MODEL_ASS" tableName="PROBSUMM"
      fieldName="MODEL"
      tableName2="MODEL" fieldName2="MODEL"/>
     <associate name="CM3R_DEVICE_ASS" tableName="CM3R"
      fieldName="logical_name"
      longFieldName="assets" tableName2="DEVICE"
      fieldName2="LOGICAL_NAME"/>
</associates>
```

Hierarchy table schema definition

Hierarchy tables are defined under a root element, `hierarchies`. Each hierarchy table definition is tagged by an element name, `hierarchy`. This element can contain one or more hierarchy tables. A hierarchy table has the following attributes.

Required attributes	Name of the table Type string Level string Unique field name
Optional attributes	RDS version string Parent table name and parent field name Full path field name

The following code shows the XML schema definition for a hierarchy table.

```
<xsd:complexType name="HierarchyType">
      <xsd:attribute name="tablename" type="xsd:string" use="required"/>
      <xsd:attribute name="rdsVersion" type="xsd:string" use="optional"/>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
      <xsd:attribute name="level" type="xsd:string" use="required"/>
      <xsd:attribute name="uniquefieldName" type="xsd:string"
       use="required"/>
      <xsd:attribute name="parenttableName" type="xsd:string"/>
      <xsd:attribute name="parentfieldname" type="xsd:string"/>
      <xsd:attribute name="fullpathfieldname" type="xsd:string"/>
</xsd:complexType>
```

The following code is an example of the `hierarchy` table type.

```
<hierarchies>
      <hierarchy tablename="location" rdsVersion="5.0"
                           type="fullpath" level="5"
                            uniquefieldName="location"
                           fullpathfieldname="location_full_name"/>
</hierarchies>
```

Direct mapping table schema definition

Direct mapping tables are defined under a root element, `directMappings`. Under this root element there can be one or multiple direct mapping table definitions. Each direct mapping table is tagged by an element name, `directMapping`.

The following is an XML schema definition for the `directMapping` table type.

```
<xsd:element name="directMapping">
                <!-- If dataSourceTableName is defined, SC delete
                      eventout put can be used to delete sync -->
                <!-- parentDimensionTableName can be multiple src table
                      names separated by | -->
                <!-- If parentDimensionTableName is not null, DEL type
                      index has to be defined -->
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="directMappingFields"/>
            <xsd:element ref="directMappingIndexes"/>
         </xsd:sequence>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="rdsVersion" type="xsd:string"/>
         <xsd:attribute name="dataSourceTableName" type="xsd:string"/>
         <xsd:attribute name="parentDimensionTableName"
type="xsd:string"/>
      </xsd:complexType>
</xsd:element>
```

This XML schema references two additional definitions: `directMappingFields` and `directMappingIndexes`. For additional details for these two definitions, examine the appropriate sections of `RDS_etl.xsd`.

The following is an example of a direct mapping table for the ServiceCenter *assignment* table.

```
<directMapping name="assignment" dataSourceTableName="assignment">
      <directMappingFields>
        <directMappingField name="name" type="char" size="60"/>
        <directMappingField name="assignment2" type="char" size="60"/>
        <directMappingField name="assignment3" type="char" size="60"/>
        <directMappingField name="reassignment" type="char" size="60"/>
        <directMappingField name="calendar_name" type="char" size="60"/>
        <directMappingField name="duty_hours" type="char" size="30"/>
        <directMappingField name="type" type="char" size="30"/>
        <directMappingField name="reassign" type="char" size="1"/>
        <directMappingField name="route_to" type="float"/>
        <directMappingField name="route_if" type="char" size="60"/>
        <directMappingField name="wdManagerName" type="char" size="60"/>
        <directMappingField name="sysmodcount" type="float"/>
        <directMappingField name="sysmoduser" type="char" size="30"/>
        <directMappingField name="sysmodtime" type="date"/>
        <directMappingField name="company" type="char" size="60"/>
        <directMappingField name="operators" type="long"/>
      </directMappingFields>
      <directMappingIndexes>
        <directMappingIndex name="UNIQASSIGNMENT_IDX">
          <DirectMappingIndexKey fieldName="name" srcFieldName="name"
         srctype="char" size="60" seqIndex="1"/>
        </directMappingIndex>
      </directMappingIndexes>
    </directMapping>
```

Slowly changing dimension (SCD) keys schema definition

The SCD keys are grouped under a root element, `scdKeys`. Each SCD key definition is tagged by `scdKey` and has the following attributes:

- rds field name
- source field name
- source type

The **Slowly Changing Dimension** data problem is a common database problem, particularly for data warehousing. Essentially, this situation applies to cases where the attribute for a record varies over time.

You can solve this type of problem in one of three ways, categorized in the following table.

Problem	Solution
Type 1	The new record replaces the original record. No trace of the old record exists.
Type 2	A new record is added to the table to represent the new information. And the original record is marked as non-active.
Type 3	The original record is modified to reflect the change.

The RDS only uses Type 2—create a new dimension record. These are used to partition the historical data and maintain historical accuracy of the RDS.

The following shows the XML schema definition for the SCD key.

```
<xsd:complexType name="scdKeyType">
    <xsd:attribute name="rdsFieldName" type="xsd:string" use="required"/>
    <!-- !!!  src field name needs '' for the sc field name with dot. ex
               "''alert.status''" -->
        <xsd:attribute name="srcFieldName" type="xsd:string"
         use="required"/>
        <xsd:attribute name="srctype" type="xsd:string" use="required"/>
</xsd:complexType>
```

The following XML is an example of `scdKeys` defined in the schema for the ServiceCenter *device* table in the RDS database.

```
<scdKeys>
        <scdKey rdsFieldName="network_name"
srcFieldName="''network.name''"
      srctype="char"/>
        <scdKey rdsFieldName="domain" srcFieldName="domain"
         srctype="char"/>
        <scdKey rdsFieldName="contact_name"
         srcFieldName="''contact.name''"
      srctype="char"/>
</scdKeys>
```

Aggregate keys schema definition

The aggregate keys are grouped under a root element, `aggregateKeys`. Each aggregate key definition is tagged by an element name, `aggregateKey`.

The following shows the XML schema definition for an `aggregateKey`.

```
<xsd:complexType name="aggregateKeyType">
                <!-- only ONE and LAST aggregate field is used as the
type of TIME -->
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
      <xsd:attribute name="size" type="xsd:string"/>
</xsd:complexType>
```

The following is an example of aggregate keys defined in the schema for the ServiceCenter *device* table.

```
<aggregateKeys>
        <aggregateKey name="type" type="char" size="60"/>
        <aggregateKey name="subtype" type="char" size="60"/>
        <aggregateKey name="Istatus" type="char" size="60"/>
        <aggregateKey name="vendor" type="char" size="60"/>
        <aggregateKey name="install_date" type="TIME"/>
</aggregateKeys>
```

RDS dimension attributes

All the dimension keys have a value of 2 if it is not the match type, and 1 if its value is blank.

For updates, the dimensions attributes provide the following information, which is used to track changes to RDS table reporting:

- When new records are added into rds dimension tables (`XXX_D`), `Z_RDSCREATEDDATE` is set to data update sync time.
- When the records are updated into rds dimension tables (`XXX_D`), `Z_RDSLASTMODDATE` is set to data update sync time.
- When the records are deleted from ServiceCenter, attributes are updated as:
  - `Z_RDSACTIVESTATUSIND = ‘N’`.
  - `Z_RDSDELETEDDATE` is set to the timestamp value at the time the record was deleted in ServiceCenter.
  - `Z_RDSTRANSENDDATE` is set to data update sync time.
  - `Z_RDSLASTMODDATE` is set to data update sync time.

- When the records are updated as SCD, the original dimension table record attributes are updated as:
  - Z_RDSTRANSLASTIND= 'N'.
  - Z_RDSLASTMODDATE is set to the data update sync time.
  - Z_RDSTRANSENDDATE is set to the data update sync time.
  - And the new record is added as ADD case.
- Z_RDSTRANSLASTIND = 'N' or Z_RDSACTIVESTATUSIND = 'N' means the dimension record is not active any more.
- Z_RDSOBSOLETEIND = 'Y' means the fact record is either a historic or a deleted record.

RDS InitRun command

You use the RDSInitRun command to have the changes you made in the RDS_etl.xml file take effect.

Usage	RDSInitRun RDS-Schema-XML-File	Modification-Indicator
Sample	RDSInitRun (the default xml file is set as rds_etl.xml)	
Sample	RDSInitRun rds_etl.xml	
Sample	RDSInitRun testdirect.xml	m
Sample	RDSInitRun testdim.xml	m
Sample	RDSInitRun rds_etl.xml	time
Sample	RDSInitRun rds_etl.xml	m_upgrade

In the sample commands, m means modified.

Synchronizing ServiceCenter record deletions with RDS

To synchronize records deleted from the ServiceCenter database with those in the RDS, earlier releases of BI Portal performed whole table scans for marking records for deletion in RDS. BI Portal 5.1 and later uses a new mechanism to improve the general performance for marking records for deletion in RDS.

BI Portal uses the following mechanisms to synchronize record deletions in the ServiceCenter database with the RDS:

- When records of a set of pre-defined ServiceCenter tables are being deleted, the associated table-level post-delete trigger is invoked. The trigger application saves the deletion events in ServiceCenter. One deletion event is saved for each deleted record. In each of these events, the system saves the table name, primary key(s) of the specified deleted record, and the timestamp of the deletion.
- When the next RDS data synchronization cycle starts, the RDS ETL process retrieves these deletion events from ServiceCenter. The data recorded in each event is processed in the following manner:
  - Find the RDS table corresponding to the specified ServiceCenter table.
  - Select the deleted record based on the primary key(s).
  - Mark the status of the record as Inactive, update the deletion timestamp of the record, and update the record accordingly.
- After the RDS ETL process completes, all the records that were deleted from ServiceCenter are properly synchronized in the RDS. The event records in ServiceCenter for the deletions are removed.

CHAPTER 2

Using Connect-It to Handle RDS Data Synchronization

This chapter explains how to utilize the Connect-It (CIT) tools to perform Reporting Data Store (RDS) data synchronization and how to tailor the scenario for your deployment. This chapter discusses the following topics:

- RDS data synchronization process
- Starting and stopping the RDS scenarios on page 36
- Configuring the connectors on page 39

RDS data synchronization process

This section explains the manual process for data synchronization.

Once the RDS database is initialized, the tables are defined in the database, but there is no data in the tables. The RDS database needs to be populated with the data from the ServiceCenter database. Connect-It scenarios perform the initial database population as well as the periodic data synchronization at a pre-defined time interval.

Connect-It performs the data population or periodic synchronization based on a pre-defined scenario. There are two RDS scenario files: `rds_sc.scn` and `rds_user.scn`. These two scenario files reside in the `/cit` folder of the RDS installation directory.

The following steps are the general process flow for the manual RDS data synchronization.

Warning: When using RDS multiple data source deployment, run only one RDS user scenario. See the Administration Guide for more information about multiple data sources.

Step 1 Start the Connect-It console application.

Step 2 Make sure services for both scenarios, `rds_sc.scn` and `rds_user.scn`, exist.

Step 3 Skip this step if the scenarios are properly configured; otherwise, configure both scenarios. This includes configuring the database connection information as well as adjusting the time interval for period data synchronization.

Step 4 Once the scenarios are properly configured, you can start the services in the Connect-It console application.

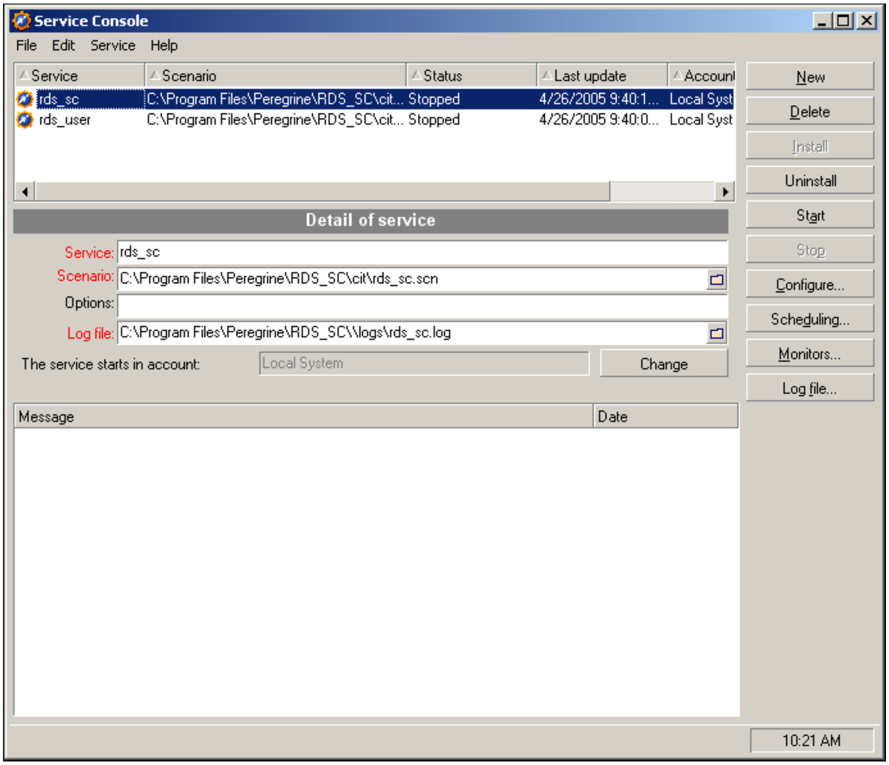
- For initial population, delete any pre-existing `rds_sc.ini` and `rds_user.ini` files before starting the services. These files reside in the `/cit` folder in the RDS installation root directory.
- Start the `rds_sc` service in the console to perform the initial data population. It then performs data synchronization in a pre-defined time interval. See Scheduling automatic data synchronization in Chapter 5 of this guide for more information. The `rds_sc.log` file contains the result of this activity. The log file resides in the `RDS/logs` directory.
- Start the `rds_user` service to populate the `RDS_USER` table in the RDS database with the operator information from ServiceCenter. It then is scheduled to perform user data synchronization at a predefined time interval.

Step 5 The services continue to run until you stop them using the Connect-It console application window.

Starting and stopping the RDS scenarios

Use the Connect-It console application to start and stop the RDS scenarios. The following figure shows two services, `rds_sc` and `rds_user`.

Warning: When using RDS multiple data source deployment, run only one RDS user scenario. See the Administration Guide for more information about multiple data sources.



Service	Scenario	Status	Last update	Account
rds_sc	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped	4/26/2005 9:40:1...	Local Syst
rds_user	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped	4/26/2005 9:40:0...	Local Syst

Service: rds_sc

Scenario: C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn

Options:

Log file: C:\Program Files\Peregrine\RDS_SC\\logs\rds_sc.log

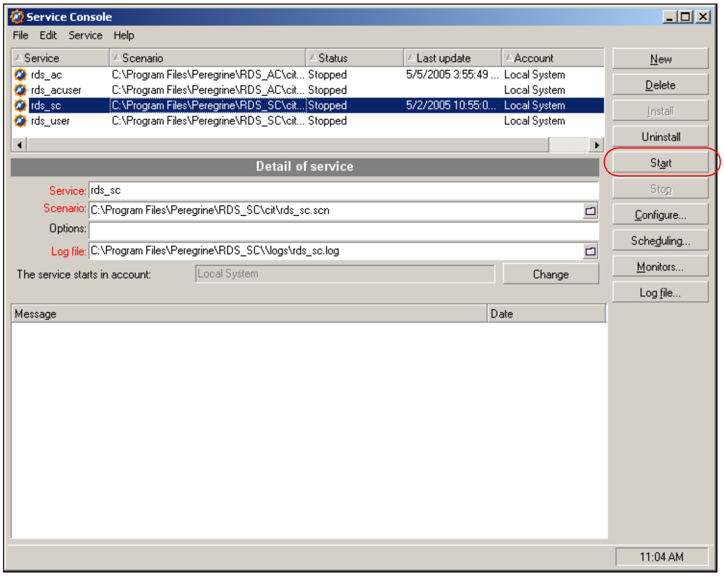
The service starts in account: Local System

Message	Date

To start the service:

1 Select the service name from the service panel.

2 Click **Start**.



Service	Scenario	Status	Last update	Account
rds_ac	C:\Program Files\Peregrine\RDS_AC\cit...	Stopped	5/5/2005 3:55:49 ...	Local System
rds_acuser	C:\Program Files\Peregrine\RDS_AC\cit...	Stopped		Local System
rds_sc	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped	5/2/2005 10:55:0...	Local System
rds_user	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped		Local System

Service: rds_sc
Scenario: C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn
Options:
Log file: C:\Program Files\Peregrine\RDS_SC\\logs\rds_sc.log
The service starts in account: Local System

Message	Date

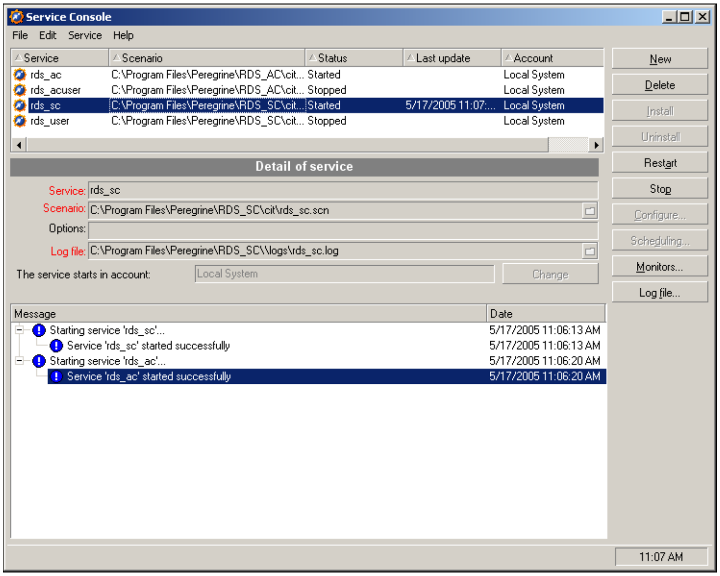
The data synchronization activities are recorded in the log file, `rds_sc.log`. Information is continuously appended to this file while the service is active. Peregrine recommends that you check the log file to monitor the status of your RDS data synchronization.

Tip: To verify that the data synchronization has no any errors, search for the string `-53` in the log file. If not found, the data synchronization ran without a problem.

To stop the service:

1 Select the service name entry from the service list panel.

2 Click **Stop** to stop the service.



Service	Scenario	Status	Last update	Account
rds_ac	C:\Program Files\Peregrine\RDS_AC\cit...	Started		Local System
rds_acuser	C:\Program Files\Peregrine\RDS_AC\cit...	Stopped		Local System
rds_sc	C:\Program Files\Peregrine\RDS_SC\cit...	Started	5/17/2005 11:07:...	Local System
rds_user	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped		Local System

Service: rds_sc
Scenario: C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn
Options:
Log file: C:\Program Files\Peregrine\RDS_SC\\logs\rds_sc.log
The service starts in account: Local System

```
Message                                      Date
⊟ Starting service 'rds_sc'...               5/17/2005 11:06:13 AM
    Service 'rds_sc' started successfully    5/17/2005 11:06:13 AM
⊟ Starting service 'rds_ac'...               5/17/2005 11:06:20 AM
    Service 'rds_ac' started successfully    5/17/2005 11:06:20 AM
```

Configuring the connectors

Use the Connect-It console application to configure the connectors. There are two out-of-box Connect-It (CIT) scenarios that come with the installation of RDS.

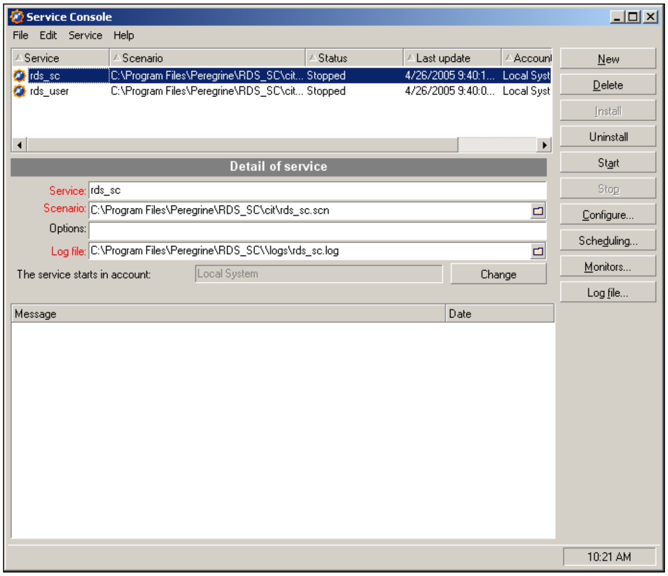
Scenario	Description
rds_sc	Synchronizes data between ServiceCenter and the RDS database.
rds_user	Synchronizes the operator data between ServiceCenter and the RDS_USER table in the RDS database.

During the installation of RDS, these two scenarios are automatically configured based on the data entered during the RDS installation. If the RDS installer cannot automatically configure these scenarios, the system displays an error message to the user. If this occurs, you must resolve the problem and then manually configure the scenarios using the Connect-It Service Console.

Warning: When using RDS multiple data source deployment, run only one RDS user scenario. See the Administration Guide for more information about multiple data sources.

To manually configure the scenarios:

1 Open the Connect-It Service Console.



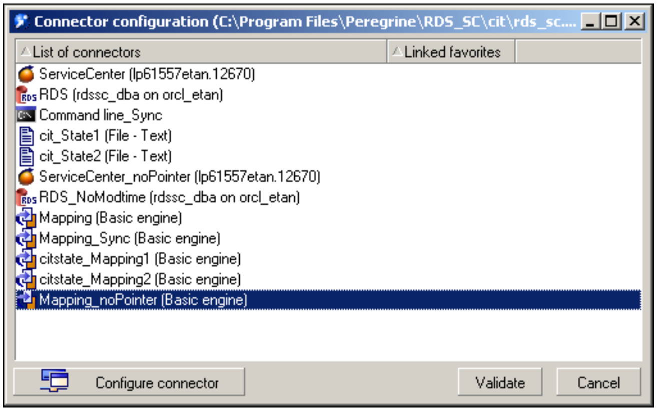
Service	Scenario	Status	Last update	Account
rds_sc	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped	4/26/2005 9:40:1...	Local Syst
rds_user	C:\Program Files\Peregrine\RDS_SC\cit...	Stopped	4/26/2005 9:40:0...	Local Syst

Service: rds_sc
Scenario: C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn
Options:
Log file: C:\Program Files\Peregrine\RDS_SC\\logs\rds_sc.log
The service starts in account: Local System

Message	Date

2 Highlight the `rds_sc` scenario.

3 Click **Configure** to open the Connector Configuration window and display a list of connectors.



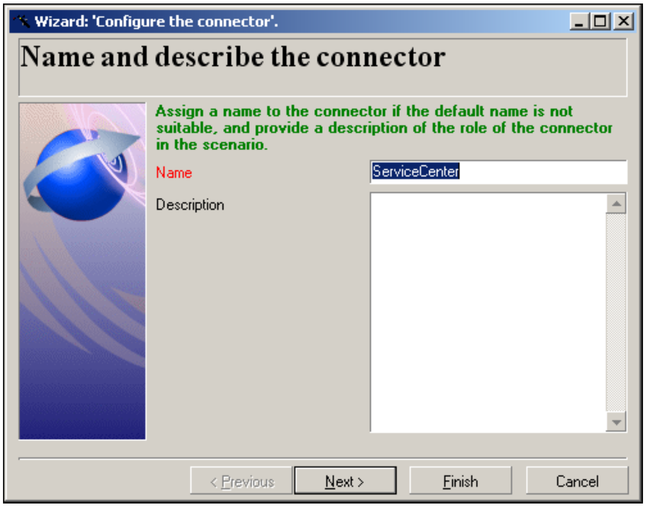
Connector configuration (C:\Program Files\Peregrine\RDS_SC\cit\rds_sc....

List of connectors	Linked favorites
ServiceCenter (lp61557etan.12670)	
RDS (rdssc_dba on orcl_etan)	
Command line_Sync	
cit_State1 (File - Text)	
cit_State2 (File - Text)	
ServiceCenter_noPointer (lp61557etan.12670)	
RDS_NoModtime (rdssc_dba on orcl_etan)	
Mapping (Basic engine)	
Mapping_Sync (Basic engine)	
citstate_Mapping1 (Basic engine)	
citstate_Mapping2 (Basic engine)	
Mapping_noPointer (Basic engine)	

4 Configure the following connectors one at a time.

ServiceCenter
RDS
ServiceCenter_noPointer
RDS_NoModtime

5 Select the connector and click **Configure connector**.



Wizard: 'Configure the connector'.

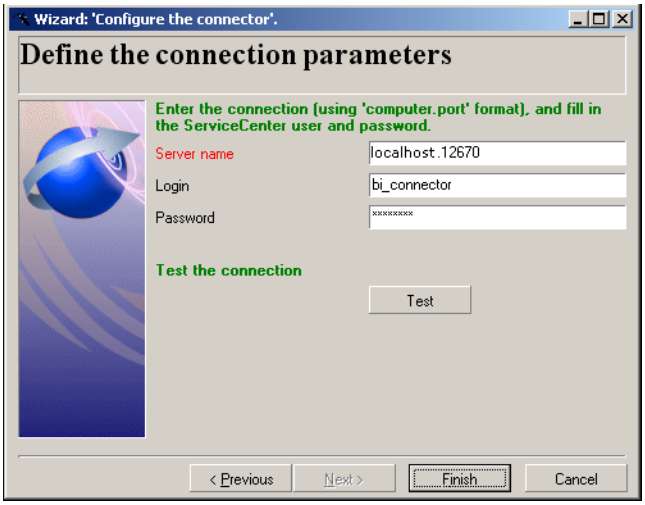
Name and describe the connector

Assign a name to the connector if the default name is not suitable, and provide a description of the role of the connector in the scenario.

Name: ServiceCenter

Description:

6 Using the **Configure the connector** wizard, click Next and enter the following information.



Define the connection parameters

Enter the connection (using 'computer.port' format), and fill in the ServiceCenter user and password.

Server name: localhost.12670

Login: bi_connector

Password: ********

Test the connection

a Type the ServiceCenter server host name and port number.

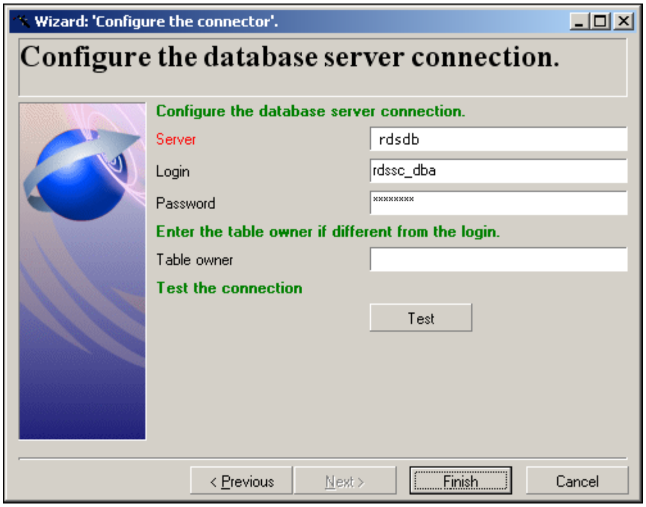
b Type the ServiceCenter default login, **bi_connector**.

c Type the ServiceCenter default login password, **passw0rd**.

d Click **Test** to verify that the connection is okay.

e If the connection is okay, click **Finish**.

7 Complete the information for the RDS-related connectors.



Wizard: 'Configure the connector'.

Configure the database server connection.

Configure the database server connection.

Server: rdsdb

Login: rdssc_dba

Password: ********

Enter the table owner if different from the login.

Table owner:

Test the connection

a Select the RDS connector and click **Configure connector**.

b Type the database alias name in Server field.

c Type the login name and password for the RDS database.

d Click **Test** to verify that the connection is okay.

e If the connection is okay, click **Finish**.

8 After you successfully configure each connector for the `rds_sc` scenario, click **Validate** to close the Connector configuration window.

9 Select the `rds_user` scenario, then click **Configure**.

10 Repeat Step 4 on page 41 through Step 8 in this section for the `rds_user` scenario.

3 CHAPTER Customization Workflow

Customizing the Reporting Data Store (RDS) is a complex, multi-step process. It is important that you properly plan these changes and execute them in the correct order. This chapter discusses the recommended workflow for creating useful and effective customizations of the RDS; however, this chapter only discusses RDS customization for the simplest case. Consult with Peregrine Professional Services if you are planning any complex customization.

Note: Always do any customization in a test environment before implementing the changes in your production environment. Do **not** customize the **rds_user** scenario.

Workflow overview

Step 1 **Planning and information gathering**—There are a number of key decisions to make during the customization process, and these considerations are best done before you even start your modifications. You need to consider the impact of your changes, and how new, changed, or deleted fields and tables will relate to other database objects. Always consider the big picture.

Step 2 **Stopping programs and processes**—You need to limit user access to some programs during the customization process. Plan for a time when the system gets little traffic and warn users of impending downtime.

Step 3 **Backing up crucial files**— Always create copies of any key files you will be editing in case you need to back out your changes.

Step 4 **Editing the RDS schema**—Before you can bring data in from ServiceCenter, that data needs a place to reside. The RDS needs to have additional tables and fields created to accept new ServiceCenter files, including all appropriate links, indices, and aliases.

Step 5 **Editing the Connect-It scenario**—Once the RDS is prepared, you need to map your modified ServiceCenter fields to the newly edited RDS fields.

Step 6 **Editing the universe**—Once the data is imported into the RDS, you need to expose that data to your users. You must add the new objects to this layer and rename or reorder them in a manner that is meaningful to the end user.

Step 7 **Editing security profiles**—If necessary, you may need to edit user access to the data you changed.

Step 8 **Restarting programs and processes**— Finally, you can bring the changes online and begin testing.

Overview: Planning and information gathering

The planning phase of your customization process is perhaps the most crucial step in your project. It is important to map out your changes before you begin.

Who should be involved?

Customization impacts multiple systems in your business intelligence network. Consult your ServiceCenter administrator, the DBA for your RDS database, and Report Designers. Notify all IT personnel responsible for bringing your servers up and down of your timetables. If this is your first time customizing the system, or if this is a larger than normal modification, Peregrine recommends that you contact Peregrine Professional Services for a consultation.

Considerations for adding data structures

In addition to simply knowing what ServiceCenter tables and fields you want to add to your RDS, you need to consider the following options:

- How does this new data link to existing tables? Will you need to define a join in the RDBMS? If so, what type? Will you need to create aliases for any tables so the joins will work correctly?
- How should the new data be indexed in the RDBMS?
- Will you be mapping your tables as a DirectMapping table, or will you be creating Dimension, Fact, and Aggregation tables? The latter are more versatile, but also require more effort to create.
- Will you need to create a Hierarchical table to properly display some of the data?
- Will you be tracking historical data in the new structures? If so, what fields will trigger the creation of a historical record?
- Will new Filter objects need to be created in the Universe?
- Will your field and data level security need to be modified to include the new data? What types of restrictions should be put in place?
- Do Delete Triggers need to be added in ServiceCenter?

Considerations for modifying and deleting existing data structures

Important: Deletions are strongly discouraged in the RDS. If a field has been deleted in ServiceCenter, or you no longer wish to expose a field to the end user, Peregrine recommends that you simply remove the mapping from the Connect-It scenario or hide the related objects in the Universe.

In addition to the previous questions, before making any modifications and deletions, you should consider the following:

- Are the data objects being changed linked to other data objects? Is the field being changed used as a link between two tables, as part of an index or unique key combination, or as a member or a parent-child relationship.
- Will deleting or hiding this object cause other data to be orphaned or rendered obsolete?
- If this is not a DirectMapping table, what is the impact this change will have on related Fact and Aggregation tables?

- Are any of the fields being changed flagged for Historical tracking? Will this change require a modification of that flag?
- Will your field and data level security need to be modified to remove non-existing fields or update filter queries?
- Will any of your existing reports have to be altered to reflect the database changes?
- Will Delete Triggers in ServiceCenter need to be removed or modified?

Since all customization is, by definition, unique, this is by no means an exhaustive list of considerations. It is merely a primer for your planning sessions.

Overview: Stopping programs and processes

At a time when your system gets little use, stop the following services:

- Connect-It Service Console (stop the rds_sc scenario and the `rds_user` scenario).
- Business Objects (BO) server
- Application server

This allows you to proceed with changes without clashing with existing processes.

Overview: Backing up crucial files

Before making row-level changes to your business intelligence system, it is HIGHLY recommended that the following files be backed up. In the event you are unable to complete the changes successfully, these files allow you to restore much of your original configuration quickly and accurately.

Make backup copies of the following files.

Note: The names listed in the table reflect the out-of-box file names. Custom files may have different names.

File	Description
`rds_etl.xml`	The RDS schema file
`rds_sc.scn`	The Connect-It scenario file
`rds.unv`	The Business Objects Universe file
`rdms_keywords.xml`	Your database reserved words list
`rds_user.scn`	Your Connect-It scenario file

Overview: Editing the RDS schema

To edit the RDS schema file, you need to start with the XML file that defines the schema: `rds_etl.xml`. This is in the `\conf` folder of the RDS installation root directory.

Navigate to that file and open it in a text editor. Take some time to study the file and its structures. There are many examples of different types of mappings. Edit this file to:

- Add new tables and fields.
- Modify or delete existing data.
- Set unique keys.
- Define Dimension, Fact, or Direct Mapping tables.
- Create Aggregated fields.
- Define the fields that trigger Historical records.

After you make the necessary changes, run the RDS initialization tool to verify the validity of your modifications. If new tables are added and record deletions for the new tables need to be synchronized during the RDS data synchronization cycle, you need to log in to ServiceCenter as an administrator and edit your Delete Triggers at this time. See Chapter 4, RDS Customization for more information.

Overview: Editing the Connect-It scenario

Once the RDS is set to receive the new data, you need to edit the mapping that pulls data from ServiceCenter to the RDS.

Open the `rds_sc.scn` file in the Connect-It Scenario Builder. This is in the `\cit` folder of the RDS installation root directory.

Refresh your view of the database and add, modify, or remove data structures to reflect your changes in ServiceCenter. Then validate and test the scenario.

See Chapter 5, Connect-It Scenario Customization for more information.

Overview: Editing the universe

Once the actual data from ServiceCenter is in the RDS, you need to modify the metadata layer to expose this data to your end user.

Log in to your Business Objects Designer tool as a user with Designer rights. Navigate to the `rds.unv` file, located in the directory under the Business Objects installation root directory:

```
\nodes\<server>\<cluster>\universes\<domain>\
```

Edit this file to:

- Modify the objects to reflect your database changes.
- Define links and aliases for tables.
- Rename and re-order fields into meaningful objects.
- Supply descriptions for universe objects.

Create your changes, then do an integrity check to verify the validity of your links. Once successful, export the universe so that it can be viewed online. See Chapter 6, Business Objects Universe Customization for more information.

Overview: Editing security profiles

This is an optional step, only for those who have set up row level security.

Log in to Business Objects Supervisor as a user with Supervisor rights. Edit the rights of appropriate users to the newly modified data structures. See Chapter 6, Business Objects Universe Customization for more information.

Overview: Restarting programs and processes

You are ready to start the system and begin testing. As a final step, restart the following processes in the specified order:

- Business Objects server (WebIntelligence cluster service)
- Connect-It Service Console (start the **rds_sc** scenario)
- Application server

Make sure to test the following:

- The data appears according to your modifications.
- Aggregations roll up correctly.
- Multiple-table queries create proper joins.
- Security settings filter data correctly.
- All standard reports still run correctly.

4 CHAPTER RDS Customization

This chapter discusses the following topics:

- RDS schema customization on page 53
- ServiceCenter trigger customization on page 60

RDS schema customization

You can modify the `RDS_etl.xml` file, an out-of-box XML file. You need an XML editor tool such as XML Spy.

Important: When customizing any XML file, use a tool that validates any changes you make to the file against the schema file.

The Reporting Data Store (RDS) Schema requires customization if you tailored the ServiceCenter (SC) database on which RDS is deployed. This section covers three different customization examples:

- New tables are added to the ServiceCenter database
- New fields are added to existing ServiceCenter tables
- Existing field attributes are changed

This section explains how to customize the RDS schema to incorporate the changes in the RDS database. The following steps describe the general process flow for RDS schema customization.

Step 1 Back up your `RDS_etl.xml` schema file.

Step 2 Use an XML editor or a text editor to make the necessary changes to the `RDS_etl.xml`schema file. Verify that the changes are valid XML and save the changes.

Step 3 Run the RDS initialization tool to create the RDS database.

Step 4 Verify that the RDS database contains the new table or field.

When you customize the RDS schema, do the following tasks to ensure that the changes you make produce the desired results before you begin any customization.

- Determine what fields and tables you need for the reports you plan to create.
- Follow naming conventions. Peregrine recommends using a prefix (for example, the initials of your company) as part of the name of any new tables or fields you create.

The customization activities described in this section assume that they are for the initial setup, which means the RDS database does not exist yet. Contact Peregrine Customer Support for further assistance if you need to customize an existing RDS database.

Warning: Running the `rds_init.bat` file against an existing RDS database will result in data loss. The RDS initialization process drops all existing tables and recreates the database based on the currently defined schema.

Adding a new table

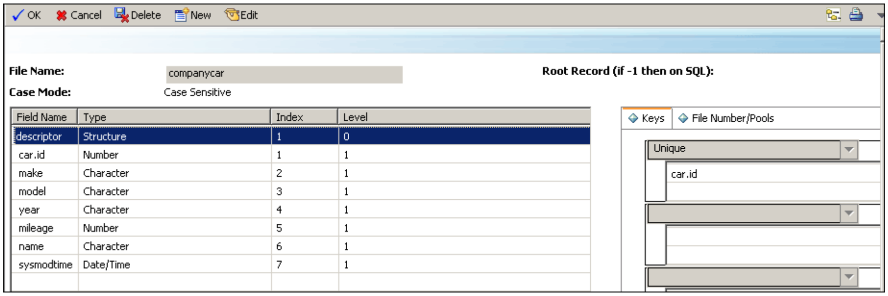
When you add a new table in ServiceCenter and you want to create reports from this new table, you must use the following procedures to add this new table to the RDS database.

Note: For ease of description, this section uses an example to describe the exact steps you need to do when adding a new table. These steps assume that the new table, *companycar*, already is in the ServiceCenter database. This table, *companycar,* has seven fields with `car.id` as the primary index.

Peregrine recommends that you always add a system-required field, *sysmodtime* to your new table. This smooth the RDS data synchronization process through Connect-It.

The following figure shows the table as it is defined in ServiceCenter.

Note: The figures in this section are for illustration only. The actual screens that display on your system depend on the client version of ServiceCenter that you are using.



File Name: companycar

Case Mode: Case Sensitive

Root Record (if -1 then on SQL):

Field Name	Type	Index	Level
descriptor	Structure	1	0
car.id	Number	1	1
make	Character	2	1
model	Character	3	1
year	Character	4	1
mileage	Number	5	1
name	Character	6	1
sysmodtime	Date/Time	7	1

Unique

car.id

After you add a new table to ServiceCenter, you need to add data to the table.

Important: Make sure that *sysmodtime* has a valid date/time data.

To add the companycar table to the RDS database:

1 Edit the RDS schema file to incorporate the new table definition.

a Identify the table type that you are adding to RDS.

In this example, `companycar` is a directMapping table, so you need to add an instance of a directMapping table into the schema file.

```
 <directMapping name="companycar" rdsVersion="5.2"
dataSourceTableName="companycar">
      <directMappingFields>
        <directMappingField name="name" type="char" size="60"/>
        <directMappingField name="car_id" type="float"/>
        <directMappingField name="make" type="char" size="60"/>
        <directMappingField name="model" type="char" size="60"/>
        <directMappingField name="year" type="char" size="8"/>
        <directMappingField name="mileage" type="long"/>
        <directMappingField name="sysmodtime" type="date"/>
      </directMappingFields>
      <directMappingIndexes>
        <directMappingIndex name="UNIQCOMPANY_CAR_IDX">
          <DirectMappingIndexKey fieldName="car_id" srcFieldName="car_id"
            srctype="float"/>
        </directMappingIndex>
      </directMappingIndexes>
   </directMapping>
```

b Using a text editor, open the RDS schema file, `RDS_etl.xml` and add the previous changes to the appropriate section of the schema file.

c Save the schema file.

2 Run the RDS initialization batch tool to create the empty table in the database.

a From a DOS command window, change the directory to the RDS deployment root directory. The default path is: `C:\Program Files\Peregrine\RDS_SC`.

b Type the following commands:

```
C: >cd common\bin
C: >rds_init
```

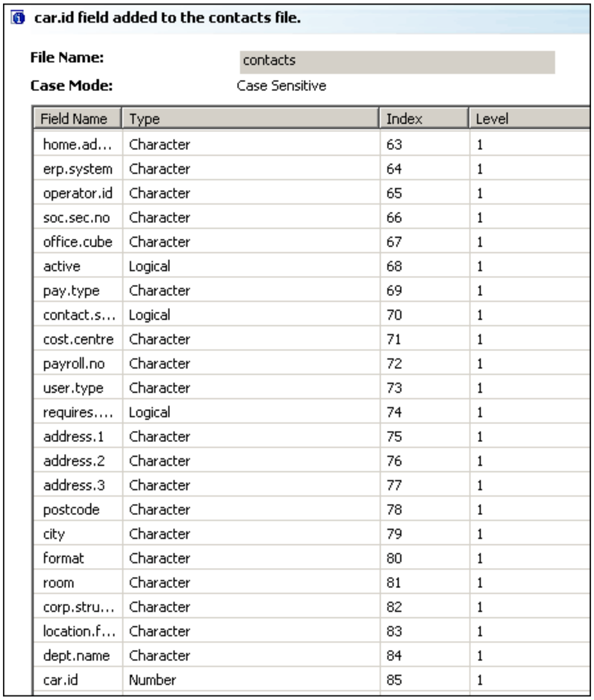
When the RDS initialization batch tool finishes initializing the RDS database, the new table is created.

3 Use the DBMS vendor’s tool to verify that the new table is created.

For example, use the Oracle DBA tool.

Adding a new field

This example begins with having a new field, `car.id`, added to an existing ServiceCenter table, *contacts*.



car.id field added to the contacts file.

File Name: contacts

Case Mode: Case Sensitive

Field Name	Type	Index	Level
home.ad...	Character	63	1
erp.system	Character	64	1
operator.id	Character	65	1
soc.sec.no	Character	66	1
office.cube	Character	67	1
active	Logical	68	1
pay.type	Character	69	1
contact.s...	Logical	70	1
cost.centre	Character	71	1
payroll.no	Character	72	1
user.type	Character	73	1
requires....	Logical	74	1
address.1	Character	75	1
address.2	Character	76	1
address.3	Character	77	1
postcode	Character	78	1
city	Character	79	1
format	Character	80	1
room	Character	81	1
corp.stru...	Character	82	1
location.f...	Character	83	1
dept.name	Character	84	1
car.id	Number	85	1

Follow the naming convention listed in Chapter 1, RDS naming conventions for naming the new RDS field name. In this example, the new field name in the ServiceCenter *contacts* table is `car.id`. Its counterpart RDS field name is `CAR_ID`.

To add a new field:

1 Add the field definition for `car.id` as `CAR_ID` in the `RDS_etl.xml` schema file.

Note new field name is added

```
 <dimension name="CONTACT">
      <dimensionTableName>
        CONTACT_D
      </dimensionTableName>
      <dataSourceTableName>
        contacts
      </dataSourceTableName>
      <dimensionTableFields>
      <dimensionTableField name="contact_name" type="char" size="140"/>
        ...(a lot of dimensionTableField definitions are omitted)
      <dimensionTableField name="comments" type="long"/>
      <dimensionTableField name="car_id" type="float"/>
      </dimensionTableFields>
      <uniqueKeys name="contact_unique">
        ...
     </uniqueKeys>
      <facts/>
      <scdKeys/>
      <aggregateKeys/>
</dimension>
```

2 Run the RDS initialization batch tool to add `car_id` to the existing `CONTACT_D` table in the database.

a From a DOS command window, change the directory to the RDS deployment root directory. The default path is: `C:\Program Files\Peregrine\RDS_SC`.

b Use the following commands:

```
C: >cd common\bin
C: >rds_init
```

When the tool finishes initializing RDS database, the new field is added.

3 Use the DBMS vendor’s tool to verify the existence of the new field in the `CONTACT_D` table.

For example, use the Oracle DBA tool to ensure that the field exists.

Changing an existing field

When you modify the attributes of an existing field of an existing ServiceCenter table, you may need to make similar modifications to RDS schema so the definitions are synchronized. When you rename a data field, it is optional to modify the RDS schema. When you change the data type of a field or change the size of a data field, Peregrine strongly recommends that you make the necessary changes in the RDS schema.

The following example demonstrates how to modify the name of an existing data field. In this example, the ServiceCenter table is *contacts*. The `cost.center` field is renamed to `mycost.center`. This example shows how to make the field change in the RDS schema.

To change an existing field:

1 Modify the RDS schema file, by renaming `cost_center` to `my_cost_center` as shown.

Note modified field name

```
<dimension name="CONTACT">
      <dimensionTableName>
        CONTACT_D
      </dimensionTableName>
      <dataSourceTableName>
        contacts
      </dataSourceTableName>
      <dimensionTableFields>
        <dimensionTableField name="contact_name" type="char" size="140"/>
        ...
        <dimensionTableField name="my_cost_center" type="char" size="30"/>
        ...
       <dimensionTableField name="comments" type="long"/>
        <dimensionTableField name="car_id" type="float"/>
      </dimensionTableFields>
      <uniqueKeys name="contact_unique">
        ...
     </uniqueKeys>
      <facts/>
      <scdKeys/>
      <aggregateKeys/>
</dimension>
```

2 Run the RDS initialization batch tool to the field renamed from cost_center to my_cost_center in the existing table, CONTACT_D, in the database.

From a DOS command window, change the directory to the RDS deployment root directory, then use the following commands:

```
C: >cd common\bin
C:>rds_init
```

When the tool finishes initializing RDS database, the name of the existing field is changed.

3 To verify the changes to the field, use the DBMS vendor's tool.

For example, use the Oracle DBA tool to ensure that the name is updated properly.

ServiceCenter trigger customization

Since the RDS database shadows the ServiceCenter database, a record deletion that occurs in ServiceCenter also needs to be properly synchronized in the RDS. To accomplish this, the RDS tracks deletion events of the various ServiceCenter tables by having ServiceCenter log record deletion events in ServiceCenter.

At the RDS data synchronization time, the RDS tool processes these deletion events. For each of the deletion events in ServiceCenter, the corresponding RDS record is processed and marked accordingly with an inactive status.

Predefined ServiceCenter triggers

To track deletion events in ServiceCenter, BI Portal uses predefined post-delete triggers for each of the following tables in ServiceCenter.

ServiceCenter Table Name	RDS Table Name	Table Type
activity	ACTIVITY	DIRECT_MAPPING
assignment	ASSIGNMENT	DIRECT_MAPPING
category	CATEGORY	DIRECT_MAPPING
clocks	CLOCK	DIMENSION
cm3r	CM3R	DIMENSION

ServiceCenter Table Name	RDS Table Name	Table Type
cm3rcatphase	CM3RCATPHASE	DIRECT_MAPPING
cm3rgroups	CM3RGROUPS	DIRECT_MAPPING
cm3sla	CM3SLA	DIRECT_MAPPING
cm3t	CM3T	DIMENSION
cm3tcatphase	CM3TCATPHASE	DIRECT_MAPPING
cmlabor	CMLABOR	DIMENSION
cmparts	CMPART	DIMENSION
company	COMPANY	DIMENSION
computer	COMPUTER	DIRECT_MAPPING
contacts	CONTACT	DIMENSION
country	COUNTRY	DIRECT_MAPPING
dept	DEPT	DIMENSION
device	DEVICE	DIMENSION
deviceparent	DEVICEPARENT	DIRECT_MAPPING
displaydevice	DISPLAYDEVICE	DIRECT_MAPPING
expline	EXPLINE	DIMENSION
furnishings	FURNISHINGS	DIRECT_MAPPING
handhelds	HANDHELDS	DIRECT_MAPPING
incidents	INCIDENT	DIMENSION
location	LOCATION	DIMENSION
mainframe	MAINFRAME	DIRECT_MAPPING
model	MODEL	DIMENSION
networkcomponents	NETWORKCOMPONENTS	DIRECT_MAPPING
officeelectronics	OFFICEELECTRONICS	DIRECT_MAPPING
operator	OPERATOR	DIMENSION
outage	OUTAGE	DIMENSION
outagedetail	OUTAGEDETAIL	DIRECT_MAPPING
pcsoftware	PCSOFTWARE	DIRECT_MAPPING
probcause	PROBCAUSE	DIRECT_MAPPING
probsummary	PROBSUMM	DIMENSION
resolution	RESOLUTION	DIRECT_MAPPING
rootcause	ROOTCAUS	DIMENSION
screlation	SCRELATION	DIRECT_MAPPING
servicecontract	SERVICEC	DIMENSION
serviceent	SERVICEENT	DIRECT_MAPPING
servicereviews	SERVICEREVIEWS	DIRECT_MAPPING
sla	SLA	DIMENSION

ServiceCenter Table Name	RDS Table Name	Table Type
slamonthly	SLAMONTHLY	DIRECT_MAPPING
slamonthlyag	SLAMONTHLYAG	DIRECT_MAPPING
slaresponse	SLARESPONSE	DIRECT_MAPPING
software	SOFTWARE	DIRECT_MAPPING
softwarelicense	SOFTWARELICENSE	DIRECT_MAPPING
storage	STORAGE	DIRECT_MAPPING
telecom	TELECOM	DIRECT_MAPPING
vendor	VENDOR	DIMENSION

Add and remove triggers for a user-defined ServiceCenter table

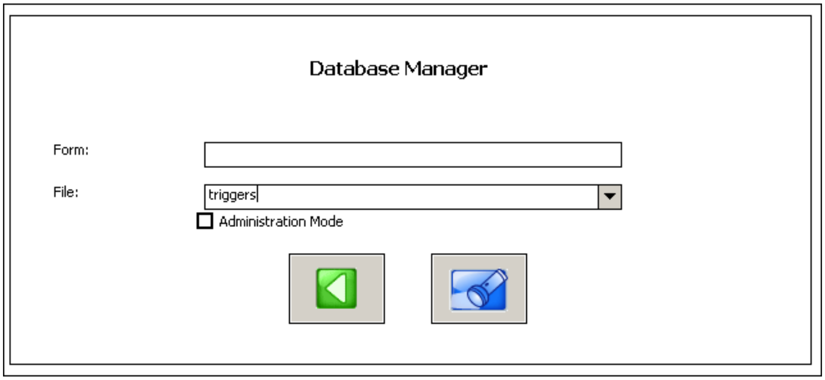
If you add a new ServiceCenter table and this new table requires the RDS record deletion synchronization support, you need to add manually a post-delete trigger for this table in ServiceCenter.

Note: The screen shots in this section are for illustration purposes only. The version of ServiceCenter you are using may have a different user interface. In the following example, the table name is `companycar`.

To define a post-delete trigger for a new ServiceCenter table:

1 Log on to ServiceCenter as `bi_admin` or some other ServiceCenter user that has administrative rights.

2 From the Main Menu, select **Toolkit** > **Database Manager** to open the Database Manager page.



Database Manager

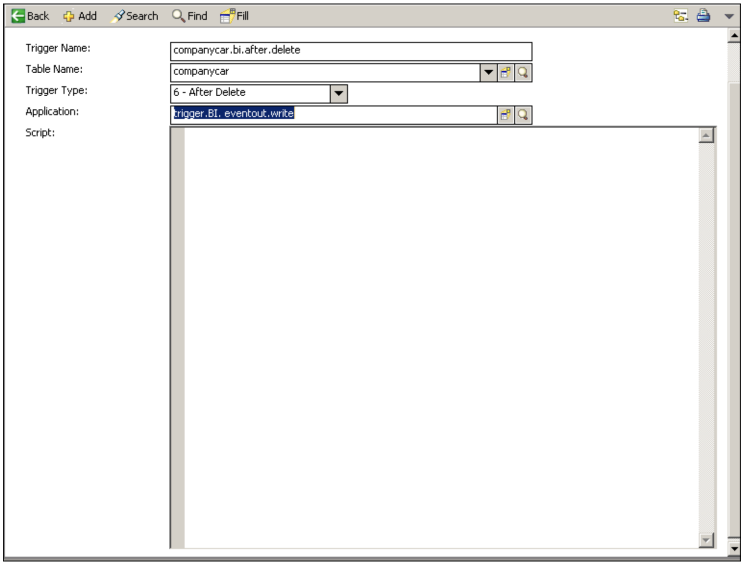
Form:

File: triggers

☐ Administration Mode

3 Type or select the table name, **triggers**, in the File text box.

4 Click **Search** to open the Triggers Search page.



Trigger Name: companycar.bi.after.delete

Table Name: companycar

Trigger Type: 6 - After Delete

Application: trigger.BI. eventout.write

Script:

5 Type the following information.

Trigger Name	`companycar.bi.after.delete`
Table Name	`companycar`
Trigger Type	`After Delete`
Application	`trigger.BI. eventout.write`

6 Click **Add** to add this trigger definition to ServiceCenter.

You can also remove a pre-defined trigger from ServiceCenter if you find that you no longer need it.

To remove a trigger for a new ServiceCenter table:

1 Log on to ServiceCenter as `bi_admin` or some other ServiceCenter user that has administrative rights.

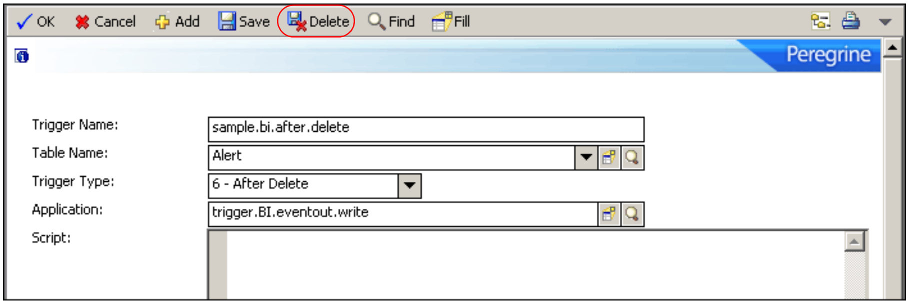
2 From the Main Menu, select **Toolkit** > **Database Manager** to open the Database Manager page.

3 Type or select the table name, **triggers**, in the File text box.

4 Click **Search** to open the Triggers Search page.

5 Type at least one search parameter; for example, the name of the trigger you want to delete.

The Triggers Search page displays the trigger record you requested or a list of records.



Trigger Name: sample.bi.after.delete
Table Name: Alert
Trigger Type: 6 - After Delete
Application: trigger.BI.eventout.write
Script:

6 If there is a list of records, select the record you want to delete and click **Delete**; otherwise, just click **Delete**.

The system asks you to confirm the delete request.

5 CHAPTER Connect-It Scenario Customization

This chapter discusses the steps to follow when you are doing BI Portal customization that requires you to modify some of the more complex components of BI Portal. In addition, this chapter discusses how to customize the Reporting Data Store (RDS) Connect-It (CIT) scenario, `rds_sc.scn`, for ServiceCenter. The RDS uses this scenario to perform data synchronization. This chapter also covers how to add a new data mapping, how to update existing mapping, and how to change the scenario scheduler. The topics in this chapter include:

- Using CIT Scenario Builder tools to view an RDS scenario on page 66
- Adding new mapping on page 67
- Update existing mapping on page 82
- Scheduling automatic data synchronization on page 87

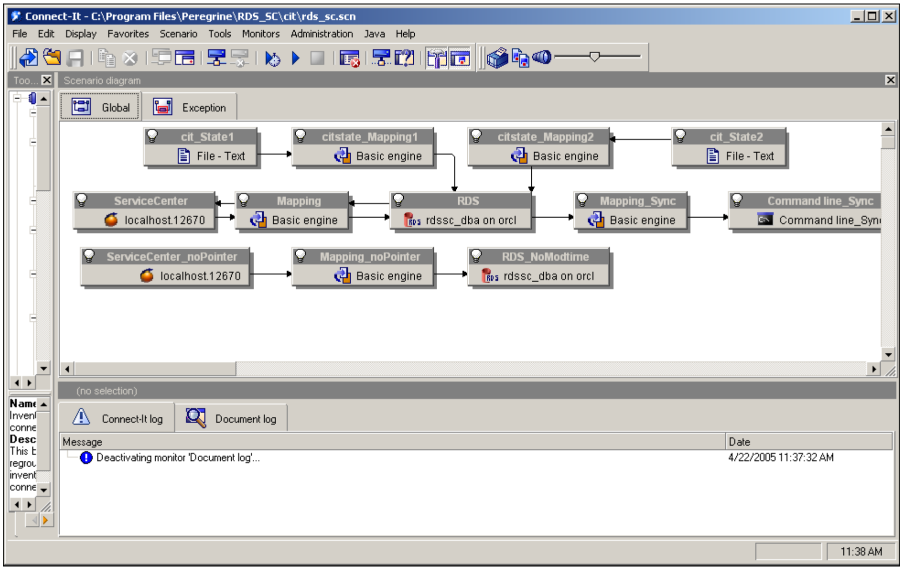
Using CIT Scenario Builder tools to view an RDS scenario

The RDS Connect-It scenario contains a collection of connectors. To view the detailed mapping information of the connectors, use the Connect-It Scenario Builder tool.

Note: Do not change the connector names defined in RDS Connect-It scenarios. There is logic dependency on these names.

To view the RDS scenario:

1 From the File menu of the Connect-It Scenario Builder tool, double-click `rds_sc.scn` to open the Scenario diagram window.

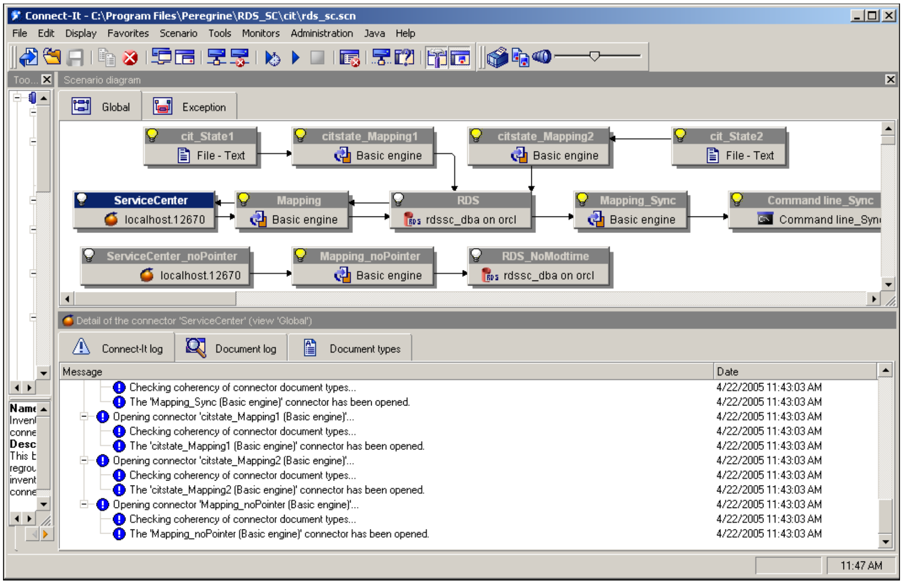


Deactivating monitor 'Document log'... 4/22/2005 11:37:32 AM

2 Highlight the ServiceCenter connector, then right-click and select **Open connector** in the pop-up menu.

This instructs Connect-It to open the data source, ServiceCenter, and make all of the document types available for selecting and mapping to the destination database, the RDS.

Connect-It displays a list of status messages in the message window highlighting information as it opens available document types defined in ServiceCenter (the source) against those defined in the scenario. If it finds any inconsistency between the scenario and ServiceCenter, the message is flagged with a red stop sign.



```
Message                                                          Date
    Checking coherency of connector document types...            4/22/2005 11:43:03 AM
    The 'Mapping_Sync (Basic engine)' connector has been opened.  4/22/2005 11:43:03 AM
Opening connector 'citstate_Mapping1 (Basic engine)'...          4/22/2005 11:43:03 AM
    Checking coherency of connector document types...            4/22/2005 11:43:03 AM
    The 'citstate_Mapping1 (Basic engine)' connector has been opened.  4/22/2005 11:43:03 AM
Opening connector 'citstate_Mapping2 (Basic engine)'...          4/22/2005 11:43:03 AM
    Checking coherency of connector document types...            4/22/2005 11:43:03 AM
    The 'citstate_Mapping2 (Basic engine)' connector has been opened.  4/22/2005 11:43:03 AM
Opening connector 'Mapping_noPointer (Basic engine)'...          4/22/2005 11:43:03 AM
    Checking coherency of connector document types...            4/22/2005 11:43:03 AM
    The 'Mapping_noPointer (Basic engine)' connector has been opened.  4/22/2005 11:43:03 AM
```

You can use the Connect-It Scenario Builder tool to make the following modifications to the existing scenario:

- Add or remove document types.
- Change existing mapping information.
- Update scheduler settings.

Adding new mapping

You must add a new data mapping to the existing scenario (`rds_sc.scn`) if you customized your ServiceCenter database by adding new tables or new data fields to existing tables. This section uses the following example to demonstrate how to add a new table to the existing scenario and how to add a new field to an existing table in the scenario.

Adding mapping for a new table

As the example in Chapter 4, RDS Customization shows, when you add a new table, companycar, in ServiceCenter, you need additional customization to make the data available for reports. The companycar table is created in the RDS database at RDS initialization time. The table has seven fields, with car.id the unique key to the table. This section presents the steps to customize the RDS scenario file, rds_sc.scn, to include the mapping information for the new table so that at RDS data synchronization time, the data in the ServiceCenter companycar table is populated to the RDS companycar table.

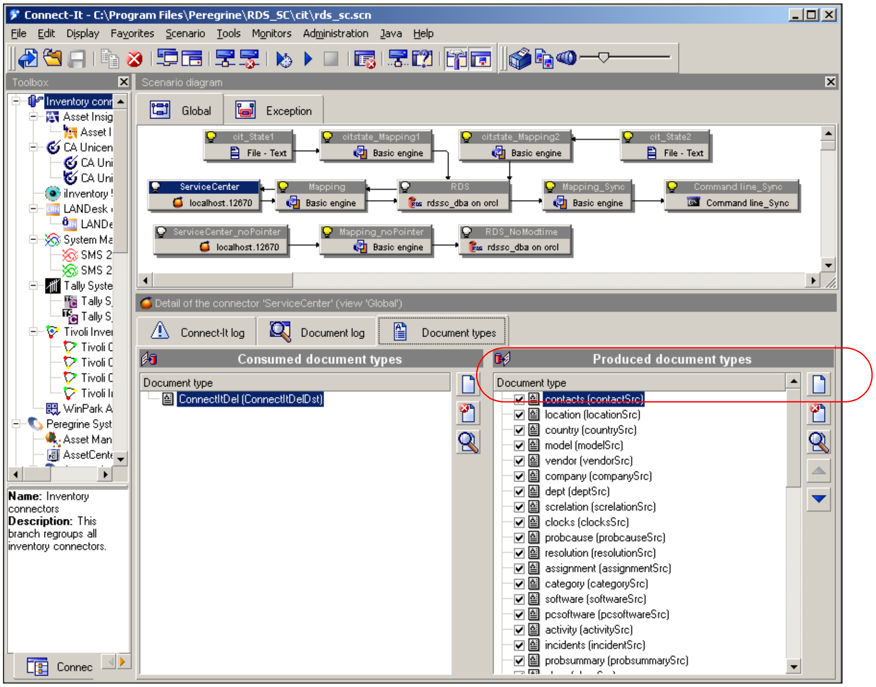
To add mapping for a new table:

1. From your Connect-It Scenario Builder tool, open rds_sc.scn.

2. Find and select the connector node with the **ServiceCenter** label, then right-click and select **Open connector** in the pop-up menu.

3. Select the Document types tab.

4 Click **Create** under the Produced document types pane.



Name: Inventory connectors
Description: This branch regroups all inventory connectors.

Detail of the connector 'ServiceCenter' (view 'Global')

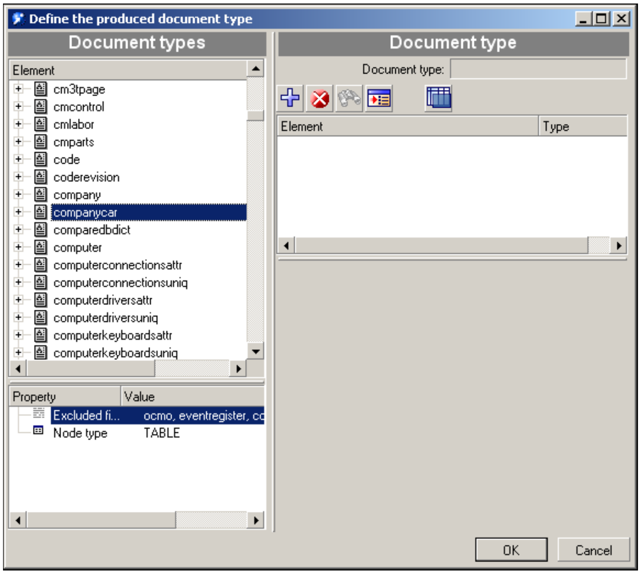
Consumed document types

Document type
- ConnectItDel (ConnectItDelDst)

Produced document types

Document type
- ☑ contacts (contactSrc)
- ☑ location (locationSrc)
- ☑ country (countrySrc)
- ☑ model (modelSrc)
- ☑ vendor (vendorSrc)
- ☑ company (companySrc)
- ☑ dept (deptSrc)
- ☑ screlation (screlationSrc)
- ☑ clocks (clocksSrc)
- ☑ probcause (probcauseSrc)
- ☑ resolution (resolutionSrc)
- ☑ assignment (assignmentSrc)
- ☑ category (categorySrc)
- ☑ software (softwareSrc)
- ☑ pcsoftware (pcsoftwareSrc)
- ☑ activity (activitySrc)
- ☑ incidents (incidentSrc)
- ☑ probsummary (probsummarySrc)

5 In the **Define the produced document type** window, select the new table, `companycar`, from the Document types list box.

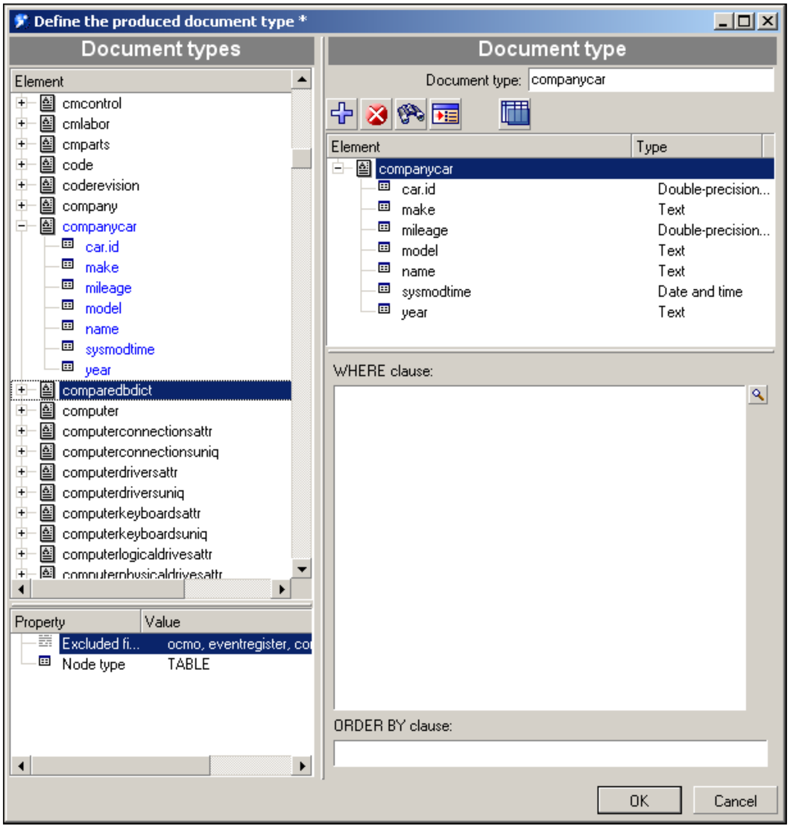


Element

- cm3tpage
- cmcontrol
- cmlabor
- cmparts
- code
- coderevision
- company
- companycar
- comparedbdict
- computer
- computerconnectionsattr
- computerconnectionsuniq
- computerdriversattr
- computerdriversuniq
- computerkeyboardsattr
- computerkeyboardsuniq

Property	Value
Excluded fi...	ocmo, eventregister, cc
Node type	TABLE

6 Click **Add** to add a new table mapping to the Document types frame.



Define the produced document type *

Document types

Element
- cmcontrol
- cmlabor
- cmparts
- code
- coderevision
- company
- companycar
  - car.id
  - make
  - mileage
  - model
  - name
  - sysmodtime
  - year
- comparedbdict
- computer
- computerconnectionsattr
- computerconnectionsuniq
- computerdriversattr
- computerdriversuniq
- computerkeyboardsattr
- computerkeyboardsuniq
- computerlogicaldrivesattr
- computerphysicaldrivesattr

Property	Value
Excluded fi...	ocmo, eventregister, co
Node type	TABLE

Document type

Document type: companycar

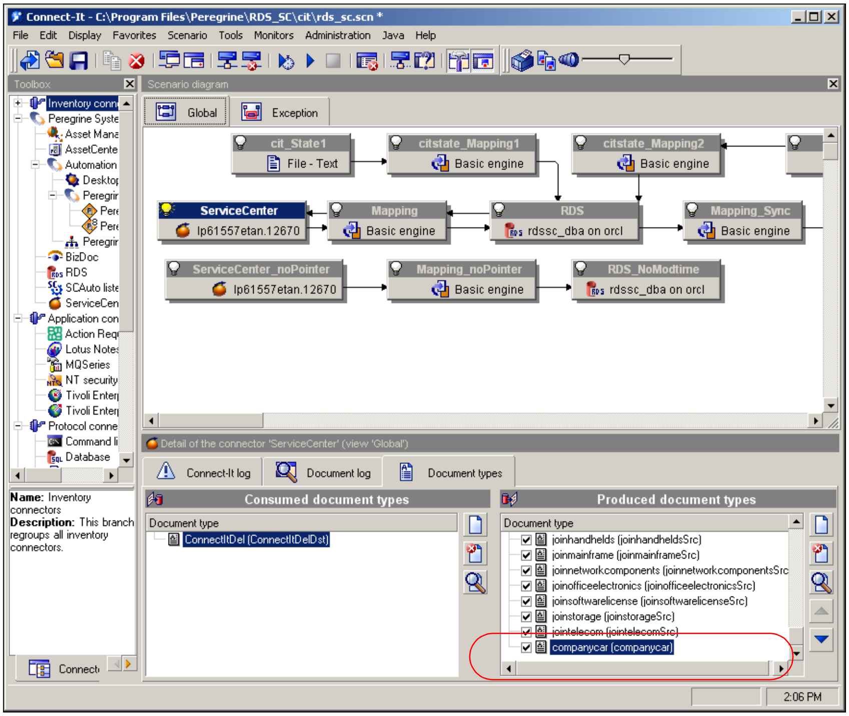
Element	Type
companycar	
car.id	Double-precision...
make	Text
mileage	Double-precision...
model	Text
name	Text
sysmodtime	Date and time
year	Text

WHERE clause:

ORDER BY clause:

7 Click **OK** to add the new table as a new document type and dismiss this window.

The `companycar` table appears as a new document type in the Produced document types pane.



Name: Inventory connectors
Description: This branch regroups all inventory connectors.

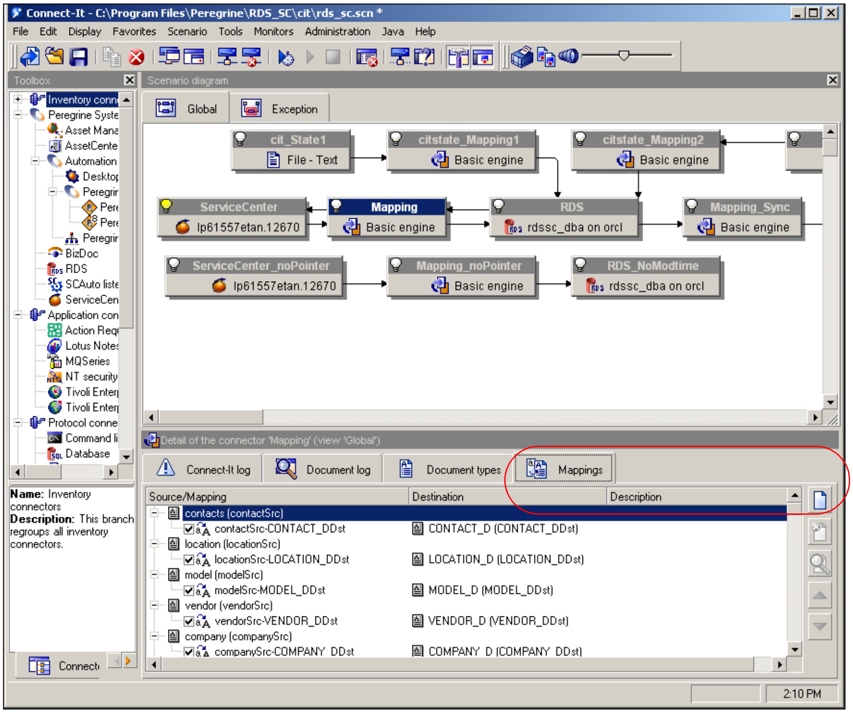
Consumed document types:

- ConnectItDel (ConnectItDelDst)

Produced document types:

- ☑ joinhandhelds (joinhandheldsSrc)
- ☑ joinmainframe (joinmainframeSrc)
- ☑ joinnetworkcomponents (joinnetworkcomponentsSrc
- ☑ joinofficeelectronics (joinofficeelectronicsSrc)
- ☑ joinsoftwarelicense (joinsoftwarelicenseSrc)
- ☑ joinstorage (joinstorageSrc)
- ☑ jointelecom (jointelecomSrc)
- ☑ companycar (companycar)

8 Select the **Mapping** connector node, click the **Mapping** tab in the lower pane, then click **Create**. 🗋



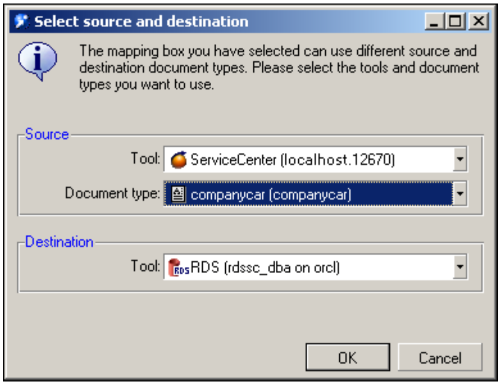
Connect-It - C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn *

Detail of the connector 'Mapping' (view 'Global')

Source/Mapping	Destination	Description
contacts (contactSrc)		
☑ contactSrc-CONTACT_DDst	CONTACT_D (CONTACT_DDst)	
location (locationSrc)		
☑ locationSrc-LOCATION_DDst	LOCATION_D (LOCATION_DDst)	
model (modelSrc)		
☑ modelSrc-MODEL_DDst	MODEL_D (MODEL_DDst)	
vendor (vendorSrc)		
☑ vendorSrc-VENDOR_DDst	VENDOR_D (VENDOR_DDst)	
company (companySrc)		
☑ companySrc-COMPANY_DDst	COMPANY_D (COMPANY_DDst)	

Name: Inventory connectors
Description: This branch regroups all inventory connectors.

9 Select `companycar` from the Document type drop-down list.



Select source and destination

The mapping box you have selected can use different source and destination document types. Please select the tools and document types you want to use.

Source

Tool: ServiceCenter (localhost.12670)

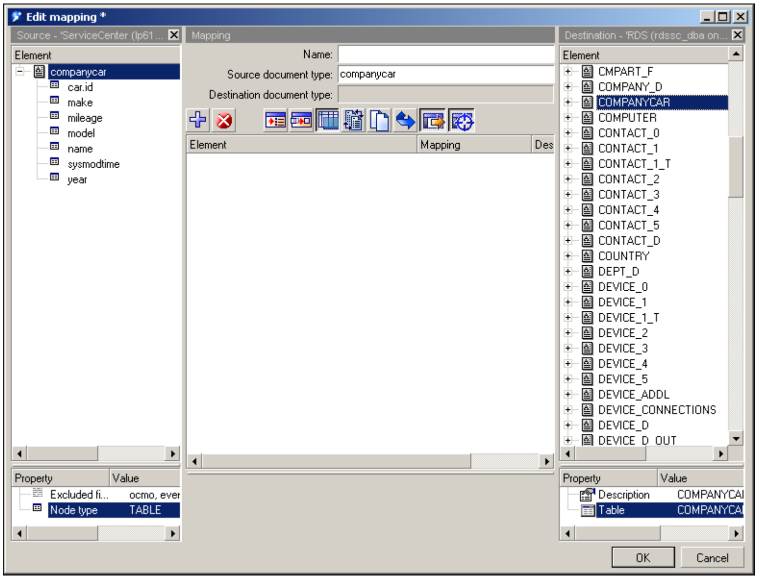
Document type: companycar (companycar)

Destination

Tool: RDS (rdssc_dba on orcl)

10 Click **OK** to dismiss the dialog box.

After Connect-It finishes processing the new mapping definition, a new mapping window opens for `companycar`.



Edit mapping *
Source - 'ServiceCenter (lp61...
Mapping
Destination - 'RDS (rdssc_dba on...
Element
Mapping
Des
OK
Cancel

Name:
Source document type: companycar
Destination document type:

Source elements: companycar, car.id, make, mileage, model, name, sysmodtime, year

Property	Value
Excluded fi...	ocmo, ever
Node type	TABLE

Destination elements: CMPART_F, COMPANY_D, COMPANYCAR, COMPUTER, CONTACT_0, CONTACT_1, CONTACT_1_T, CONTACT_2, CONTACT_3, CONTACT_4, CONTACT_5, CONTACT_D, COUNTRY, DEPT_D, DEVICE_0, DEVICE_1, DEVICE_1_T, DEVICE_2, DEVICE_3, DEVICE_4, DEVICE_5, DEVICE_ADDL, DEVICE_CONNECTIONS, DEVICE_D, DEVICE_D_OUT

Property	Value
Description	COMPANYCAI
Table	COMPANYCA

11 To edit the mapping for `companycar`, type **companycar** in the Name field.

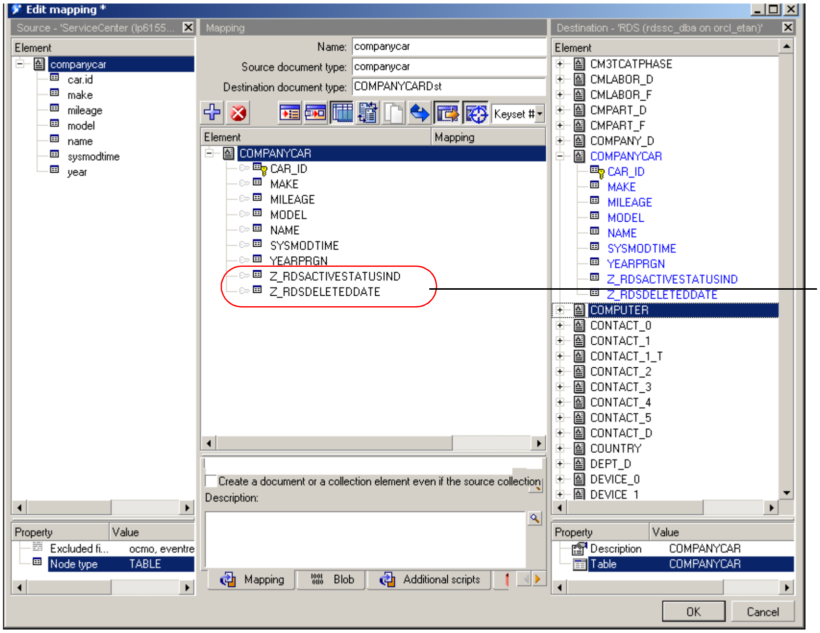
12 Select `COMPANYCAR` from the destination list box.

13 Click **Add** to add the mapping from source table to destination table. ✚

14 To add the field mapping information from a source table to its destination table, use the mouse to select a field from the source table and then drag the

selected field to the corresponding field in the mapping element list of the destination table.

For example, `car.id` of `companycar` maps to `CAR_ID` of `COMPANYCAR`, so selecting `car.id` and dragging the selection to `CAR_ID` causes the mapping `['car.id']` to be created visually.



Name: companycar

Source document type: companycar

Destination document type: COMPANYCARDst

☐ Create a document or a collection element even if the source collection

Description:

Property	Value
Excluded fi...	ocmo, eventre
Node type	TABLE

Property	Value
Description	COMPANYCAR
Table	COMPANYCAR

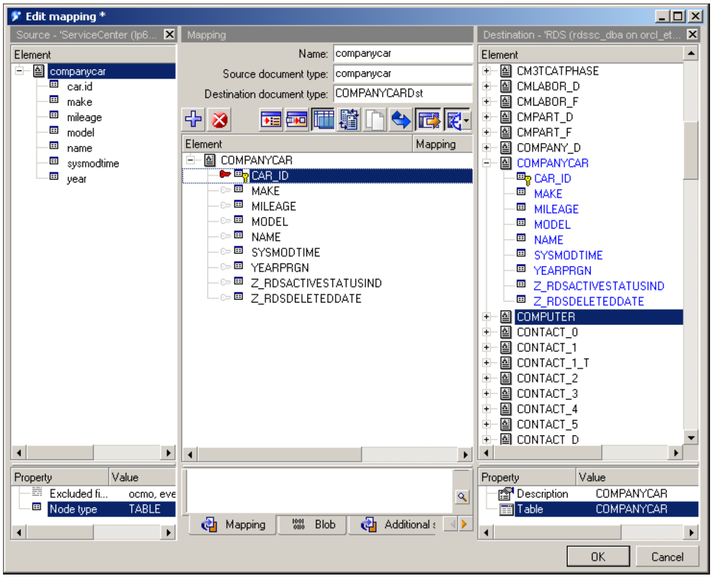
These are internal RDS fields. No action or mapping is required.

When you finish adding all field mappings, you need to define the primary key field.

To define the primary key field of the new table for Connect-It:

1 In the mapping panel (middle pane), click the tiny transparent key icon in front of the CAR_ID field.

In this example, the primary key for companycar is CAR_ID. It turns red.



Name: companycar

Source document type: companycar

Destination document type: COMPANYCARDst

Property	Value
Excluded fi...	ocmo, eve
Node type	TABLE

Property	Value
Description	COMPANYCAR
Table	COMPANYCAR

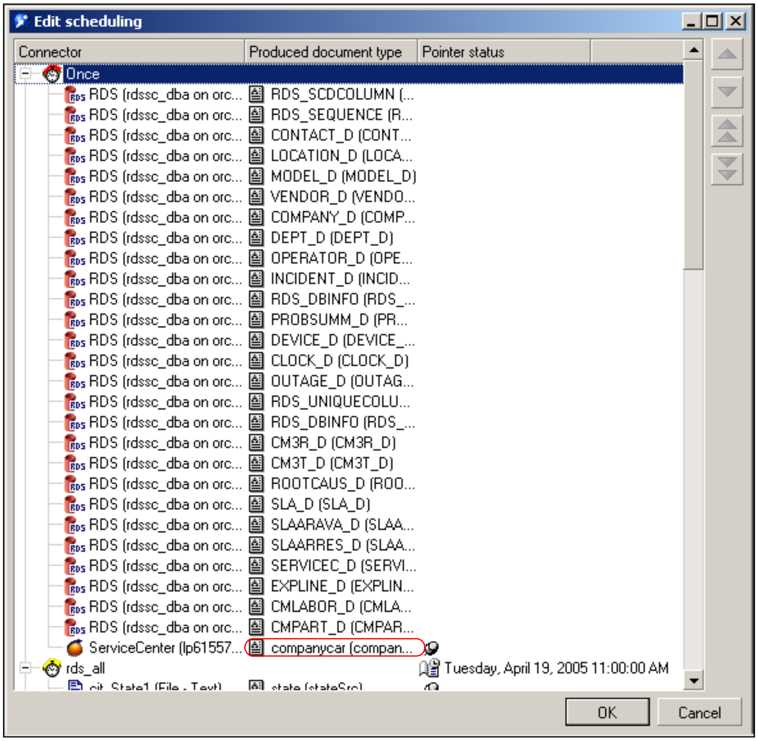
You defined the primary key.

2 Click **OK** to dismiss Edit mapping.

You successfully added a new table mapping to the RDS scenario.

To add the new table to the scheduling process:

1 From the Connect-It Scenario Builder, click **Scenario** > **Scheduling** to open Edit Scheduling.

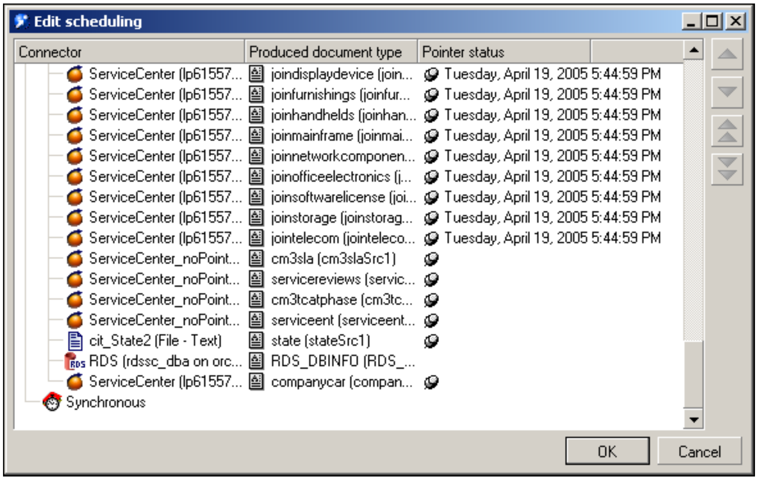


Connector	Produced document type	Pointer status
Once		
RDS (rdssc_dba on orc...	RDS_SCDCOLUMN (...	
RDS (rdssc_dba on orc...	RDS_SEQUENCE (R...	
RDS (rdssc_dba on orc...	CONTACT_D (CONT...	
RDS (rdssc_dba on orc...	LOCATION_D (LOCA...	
RDS (rdssc_dba on orc...	MODEL_D (MODEL_D)	
RDS (rdssc_dba on orc...	VENDOR_D (VENDO...	
RDS (rdssc_dba on orc...	COMPANY_D (COMP...	
RDS (rdssc_dba on orc...	DEPT_D (DEPT_D)	
RDS (rdssc_dba on orc...	OPERATOR_D (OPE...	
RDS (rdssc_dba on orc...	INCIDENT_D (INCID...	
RDS (rdssc_dba on orc...	RDS_DBINFO (RDS_...	
RDS (rdssc_dba on orc...	PROBSUMM_D (PR...	
RDS (rdssc_dba on orc...	DEVICE_D (DEVICE_...	
RDS (rdssc_dba on orc...	CLOCK_D (CLOCK_D)	
RDS (rdssc_dba on orc...	OUTAGE_D (OUTAG...	
RDS (rdssc_dba on orc...	RDS_UNIQUECOLU...	
RDS (rdssc_dba on orc...	RDS_DBINFO (RDS_...	
RDS (rdssc_dba on orc...	CM3R_D (CM3R_D)	
RDS (rdssc_dba on orc...	CM3T_D (CM3T_D)	
RDS (rdssc_dba on orc...	ROOTCAUS_D (ROO...	
RDS (rdssc_dba on orc...	SLA_D (SLA_D)	
RDS (rdssc_dba on orc...	SLAARAVA_D (SLAA...	
RDS (rdssc_dba on orc...	SLAARRES_D (SLAA...	
RDS (rdssc_dba on orc...	SERVICEC_D (SERVI...	
RDS (rdssc_dba on orc...	EXPLINE_D (EXPLIN...	
RDS (rdssc_dba on orc...	CMLABOR_D (CMLA...	
RDS (rdssc_dba on orc...	CMPART_D (CMPAR...	
ServiceCenter (lp61557...	companycar (compan...	
rds_all		Tuesday, April 19, 2005 11:00:00 AM
cit_State1 (File - Text)	state (stateSrc)	

2 Scroll down the list of table names for connector `Once`.

3 Select `companycar` in the Produced document type column.

4 Drag-and-drop the `companycar` entry from connector `Once` to connector `rds_all`.



Connector	Produced document type	Pointer status
ServiceCenter (lp61557...	joindisplaydevice (join...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinfurnishings (joinfur...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinhandhelds (joinhan...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinmainframe (joinmai...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinnetworkcomponen...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinofficeelectronics (j...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinsoftwarelicense (joi...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	joinstorage (joinstorag...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter (lp61557...	jointelecom (jointeleco...	Tuesday, April 19, 2005 5:44:59 PM
ServiceCenter_noPoint...	cm3sla (cm3slaSrc1)	
ServiceCenter_noPoint...	servicereviews (servic...	
ServiceCenter_noPoint...	cm3tcatphase (cm3tc...	
ServiceCenter_noPoint...	serviceent (serviceent...	
cit_State2 (File - Text)	state (stateSrc1)	
RDS (rdssc_dba on orc...	RDS_DBINFO (RDS_...	
ServiceCenter (lp61557...	companycar (compan...	
Synchronous		

This completes the process of adding a mapping for a new table.

5 Save the updated scenario and use the Connect-It console to run the updated scenario.

You are ready to use the scenario to perform data synchronization.

Adding a new field to an existing mapping

In Chapter 4, RDS Customization, you added a new field, `car.id`, to the `contacts` table in ServiceCenter and added the `CAR_ID` field to the `contact_d` table in the RDS. For the Connect-It scenario to populate this new field in the RDS with data, you must modify the RDS scenario to include mapping information for the new field. The following example shows how to add the new field, `car.id`, to an existing table mapping in `rds_sc.scn`.

To add a new field to rds_sc.scn:

1 Open `rds_sc.scn` in your Connect-It Scenario Builder tool.

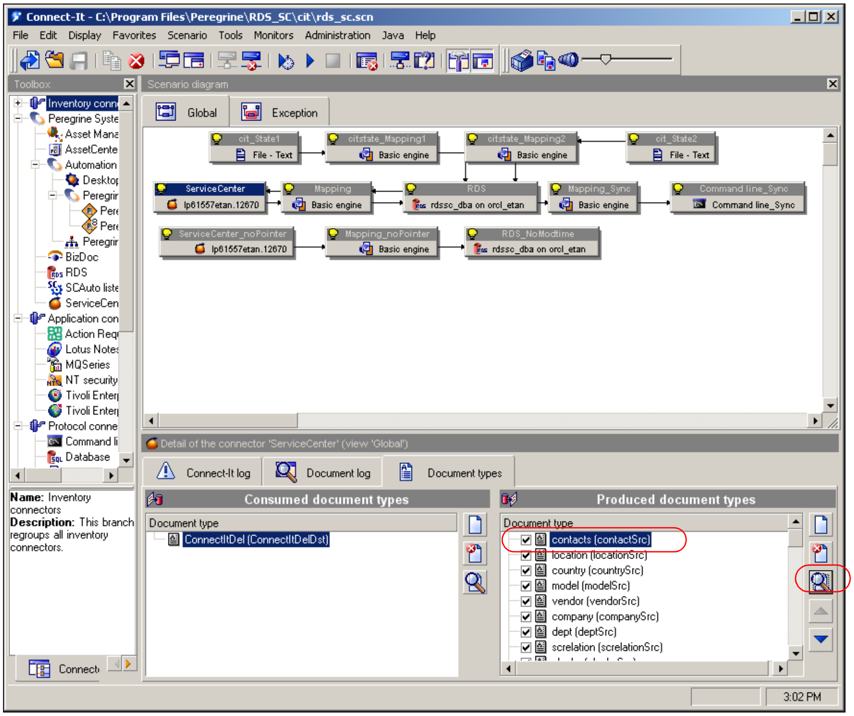
2 Find and select the **ServiceCenter**connector node, then right-click to display **Open connector** from the pop-up menu.

3 Add the new `car.id` field to the **contactSrc** document type.

a Click the **Document types** tab.

b Select the `contacts (contactSrc)` element in the Produced document type pane.

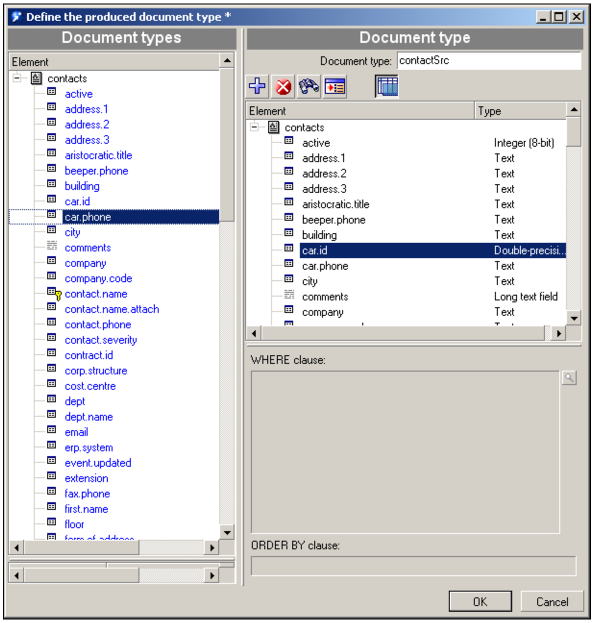
c Click **Edit** to edit the detail fields of the contacts table.



4 Edit the document type, `contactSrc`, and add the new field `car.id`.

a Select the `car.id` field in the Element list box for the `contacts` table in the left pane of the window and click **Add** to add the new field to **contactSrc.**

b Click **OK** to dismiss the edit document type window.



Document type: contactSrc

Element
- contacts
  - active
  - address.1
  - address.2
  - address.3
  - aristocratic.title
  - beeper.phone
  - building
  - car.id
  - car.phone
  - city
  - comments
  - company
  - company.code
  - contact.name
  - contact.name.attach
  - contact.phone
  - contact.severity
  - contract.id
  - corp.structure
  - cost.centre
  - dept
  - dept.name
  - email
  - erp.system
  - event.updated
  - extension
  - fax.phone
  - first.name
  - floor
  - form.of.address

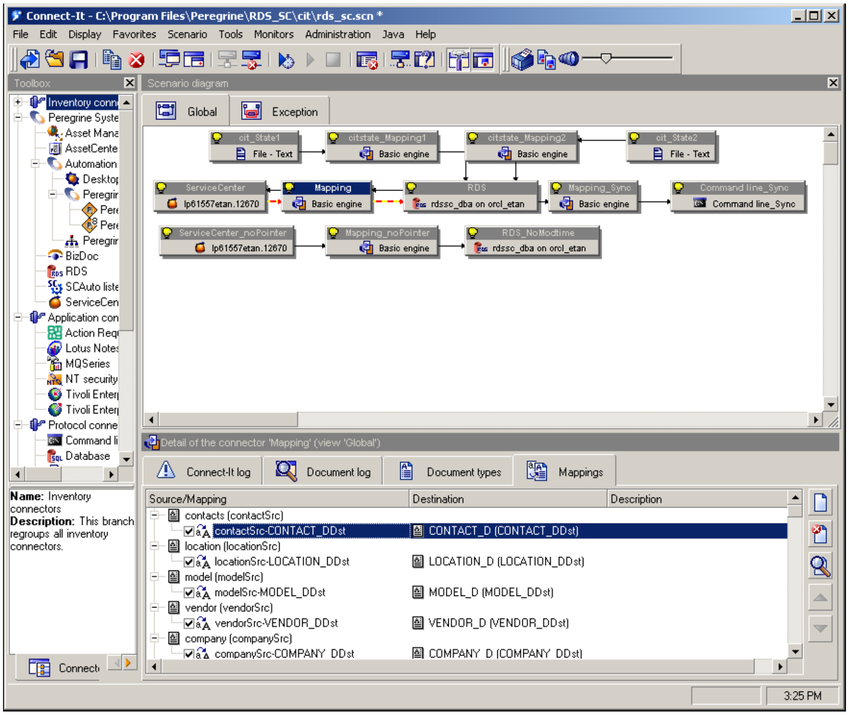
Element	Type
contacts	
active	Integer (8-bit)
address.1	Text
address.2	Text
address.3	Text
aristocratic.title	Text
beeper.phone	Text
building	Text
car.id	Double-precisi...
car.phone	Text
city	Text
comments	Long text field
company	Text

5 Add a new field mapping for `car.id` by selecting the **Mapping** connector.

a Click the **Mappings** tab.

b Select the mapping relationship of `contactSrc` to `CONTACT_D`.

c Click **Edit** to open the Edit mapping window for this mapping.



Connect-It - C:\Program Files\Peregrine\RDS_SC\cit\rds_sc.scn *

Name: Inventory connectors
Description: This branch regroups all inventory connectors.

Detail of the connector 'Mapping' (view 'Global')

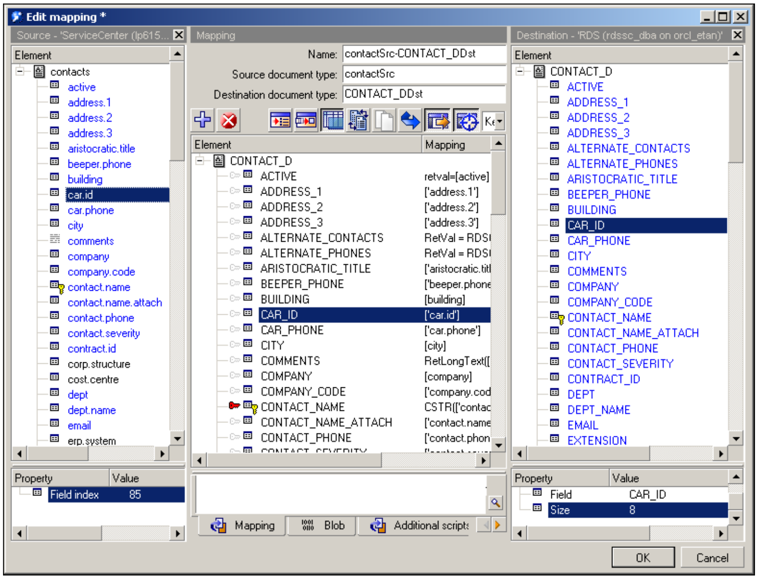
Source/Mapping	Destination	Description
contacts (contactSrc)		
☑ contactSrc-CONTACT_DDst	CONTACT_D (CONTACT_DDst)	
location (locationSrc)		
☑ locationSrc-LOCATION_DDst	LOCATION_D (LOCATION_DDst)	
model (modelSrc)		
☑ modelSrc-MODEL_DDst	MODEL_D (MODEL_DDst)	
vendor (vendorSrc)		
☑ vendorSrc-VENDOR_DDst	VENDOR_D (VENDOR_DDst)	
company (companySrc)		
☑ companySrc-COMPANY_DDst	COMPANY_D (COMPANY_DDst)	

d Select the CAR_ID field from the Destination element list box.

e Click **Add** to add to the mapping element list.

f Select the `car.id` field from source element list box, then drag-and-drop it on the `CAR_ID` element in the Mapping element list box.

g Click **OK** to finish the mapping definition.



Edit mapping *

Name: contactSrc-CONTACT_DDst

Source document type: contactSrc

Destination document type: CONTACT_DDst

Element	Mapping
CONTACT_D	
ACTIVE	retval=[active]
ADDRESS_1	['address.1']
ADDRESS_2	['address.2']
ADDRESS_3	['address.3']
ALTERNATE_CONTACTS	RetVal = RDS!
ALTERNATE_PHONES	RetVal = RDS!
ARISTOCRATIC_TITLE	['aristocratic.titl
BEEPER_PHONE	['beeper.phone
BUILDING	[building]
CAR_ID	['car.id']
CAR_PHONE	['car.phone']
CITY	[city]
COMMENTS	RetLongText([
COMPANY	[company]
COMPANY_CODE	['company.cod
CONTACT_NAME	CSTR(['contac
CONTACT_NAME_ATTACH	['contact.name
CONTACT_PHONE	['contact.phon

Source – Property / Value:

Property	Value
Field index	85

Destination – Property / Value:

Property	Value
Field	CAR_ID
Size	8

Test your customized RDS scenario

Once you customize the RDS scenario, run the scenario to perform the data synchronization process. See Chapter 3, Customization Workflow for more information.

Update existing mapping

If attributes for existing fields in the ServiceCenter database and the RDS database have been modified, then the mapping information for these changed fields needs to be modified in the RDS `rds_sc.scn` scenario file.

The following example demonstrates how you can incorporate changes for a field name; for example, `cost.center`.

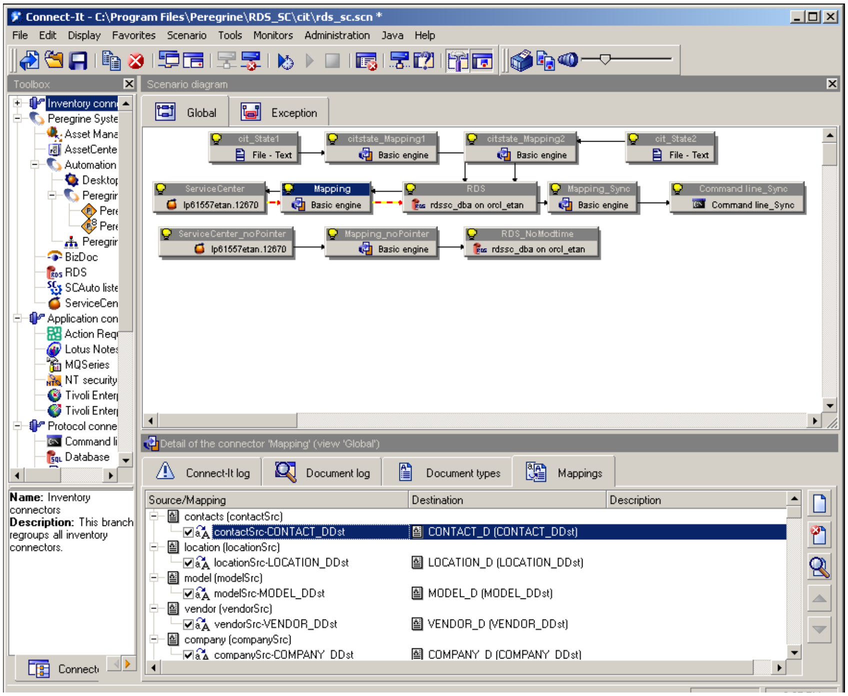
To update mapping information for a field:

1. Open `rds_sc.scn` in your Connect-It Scenario Builder tool.

2. Find and select the **ServiceCenter** connector node, then right-click to display **Open connector** from the pop-up menu.

3. Click the Document types tab.

4. Select the `contacts (contactSrc)` element in the Produced Document Type pane, then select **Edit Mapping** to edit the existing mapping.



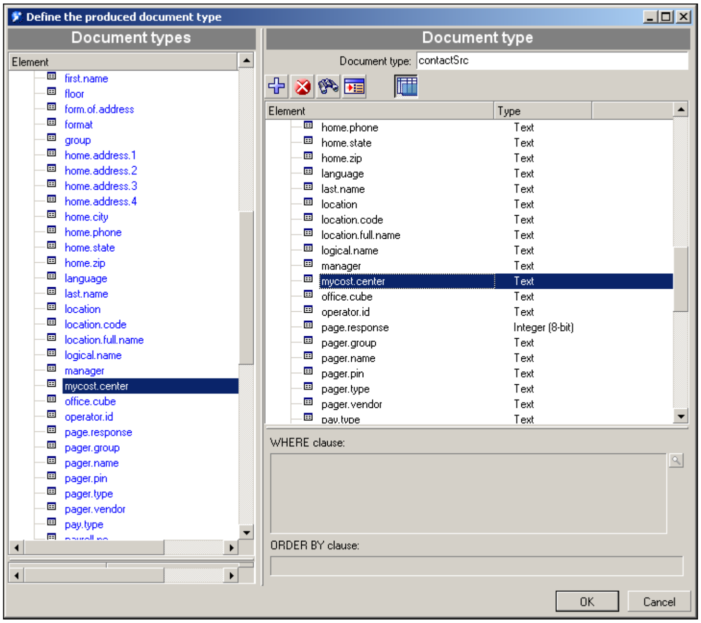
Name: Inventory connectors
Description: This branch regroups all inventory connectors.

Source/Mapping	Destination	Description
contacts (contactSrc)		
☑ contactSrc-CONTACT_DDst	CONTACT_D (CONTACT_DDst)	
location (locationSrc)		
☑ locationSrc-LOCATION_DDst	LOCATION_D (LOCATION_DDst)	
model (modelSrc)		
☑ modelSrc-MODEL_DDst	MODEL_D (MODEL_DDst)	
vendor (vendorSrc)		
☑ vendorSrc-VENDOR_DDst	VENDOR_D (VENDOR_DDst)	
company (companySrc)		
☑ companySrc-COMPANY_DDst	COMPANY_D (COMPANY_DDst)	

5. Select `mycost.center` from the `contacts` Document types list box on the left-hand side of the window.

6. Click **Add** to update the mapping for the renamed field to `contactSrc`.

7 Click **OK** to dismiss the window.



Define the produced document type

Document types

Element
- first.name
- floor
- form.of.address
- format
- group
- home.address.1
- home.address.2
- home.address.3
- home.address.4
- home.city
- home.phone
- home.state
- home.zip
- language
- last.name
- location
- location.code
- location.full.name
- logical.name
- manager
- mycost.center
- office.cube
- operator.id
- page.response
- pager.group
- pager.name
- pager.pin
- pager.type
- pager.vendor
- pay.type

Document type

Document type: contactSrc

Element	Type
home.phone	Text
home.state	Text
home.zip	Text
language	Text
last.name	Text
location	Text
location.code	Text
location.full.name	Text
logical.name	Text
manager	Text
mycost.center	Text
office.cube	Text
operator.id	Text
page.response	Integer (8-bit)
pager.group	Text
pager.name	Text
pager.pin	Text
pager.type	Text
pager.vendor	Text
pay.type	Text

WHERE clause:

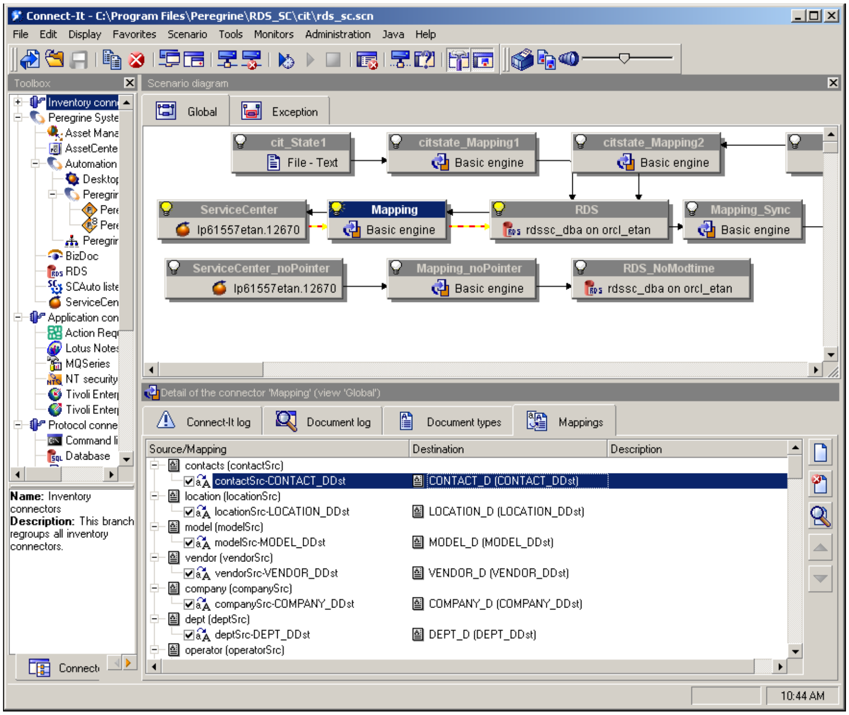
ORDER BY clause:

8 Update the mapping information.

a Select the **Mapping** connector node, then right-click to display **Open connector** from the pop-up menu.

b Select `contactSrc-CONTACT_DDst` from the Mappings tab.

c Click **Edit** to open the detail edit mapping window.

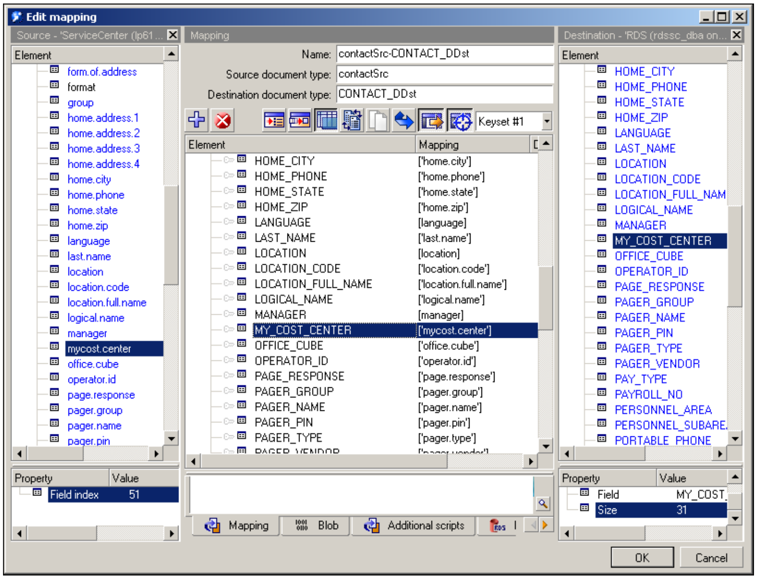


Source/Mapping	Destination	Description
contacts (contactSrc)		
☑ contactSrc-CONTACT_DDst	CONTACT_D (CONTACT_DDst)	
location (locationSrc)		
☑ locationSrc-LOCATION_DDst	LOCATION_D (LOCATION_DDst)	
model (modelSrc)		
☑ modelSrc-MODEL_DDst	MODEL_D (MODEL_DDst)	
vendor (vendorSrc)		
☑ vendorSrc-VENDOR_DDst	VENDOR_D (VENDOR_DDst)	
company (companySrc)		
☑ companySrc-COMPANY_DDst	COMPANY_D (COMPANY_DDst)	
dept (deptSrc)		
☑ deptSrc-DEPT_DDst	DEPT_D (DEPT_DDst)	
operator (operatorSrc)		

9 Bind the `MY_COST_CENTER` field in `CONTACT_D` in the RDS and `mycost.center` in the `contacts` table in ServiceCenter.

a Select `MY_COST_CENTER` from the `CONTACT_D` Destination RDS Element list box in the right pane.

b Click **Add** to add it to the Mapping element list box.



Edit mapping

Source - 'ServiceCenter (lp61 ...

Element

- form.of.address
- format
- group
- home.address.1
- home.address.2
- home.address.3
- home.address.4
- home.city
- home.phone
- home.state
- home.zip
- language
- last.name
- location
- location.code
- location.full.name
- logical.name
- manager
- mycost.center
- office.cube
- operator.id
- page.response
- pager.group
- pager.name
- pager.pin

Property	Value
Field index	51

Mapping

Name: contactSrc-CONTACT_DDst

Source document type: contactSrc

Destination document type: CONTACT_DDst

Keyset #1

Element	Mapping
HOME_CITY	['home.city']
HOME_PHONE	['home.phone']
HOME_STATE	['home.state']
HOME_ZIP	['home.zip']
LANGUAGE	[language]
LAST_NAME	['last.name']
LOCATION	[location]
LOCATION_CODE	['location.code']
LOCATION_FULL_NAME	['location.full.name']
LOGICAL_NAME	['logical.name']
MANAGER	[manager]
MY_COST_CENTER	['mycost.center']
OFFICE_CUBE	['office.cube']
OPERATOR_ID	['operator.id']
PAGE_RESPONSE	['page.response']
PAGER_GROUP	['pager.group']
PAGER_NAME	['pager.name']
PAGER_PIN	['pager.pin']
PAGER_TYPE	['pager.type']
PAGER_VENDOR	['pager.vendor']

Destination - 'RDS (rdssc_dba on...

Element

- HOME_CITY
- HOME_PHONE
- HOME_STATE
- HOME_ZIP
- LANGUAGE
- LAST_NAME
- LOCATION
- LOCATION_CODE
- LOCATION_FULL_NAM
- LOGICAL_NAME
- MANAGER
- MY_COST_CENTER
- OFFICE_CUBE
- OPERATOR_ID
- PAGE_RESPONSE
- PAGER_GROUP
- PAGER_NAME
- PAGER_PIN
- PAGER_TYPE
- PAGER_VENDOR
- PAY_TYPE
- PAYROLL_NO
- PERSONNEL_AREA
- PERSONNEL_SUBARE
- PORTABLE_PHONE

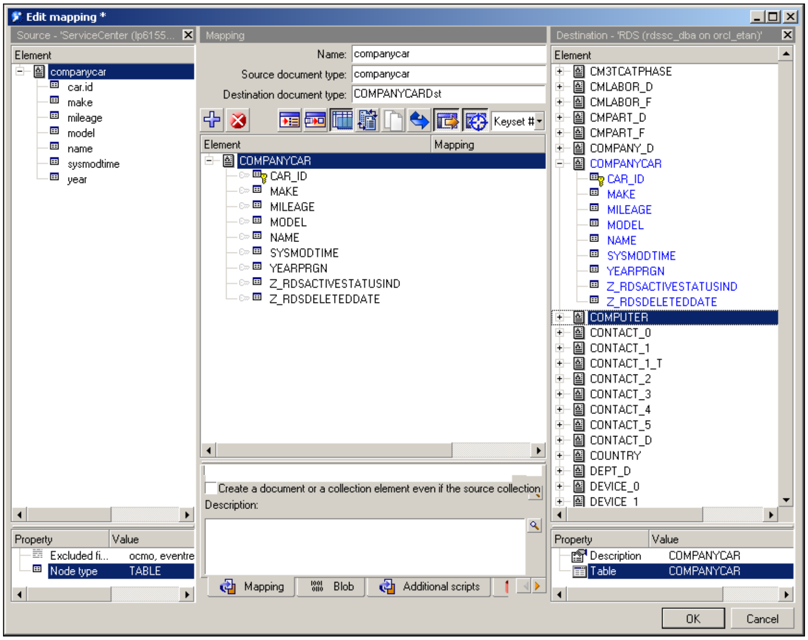
Property	Value
Field	MY_COST_
Size	31

c Select the `mycost.center` field from the `contacts` Source Element list.

d Drag-and-drop the `mycost.center` field to the Destination Element, MY_COST_CENTER.

This completes the mapping of `mycost.center` of the ServiceCenter `contacts` table to `my_cost_center` of the RDS `CONTACT` table.

e Click **OK** to save the new mapping information.



Edit mapping *

Source - 'ServiceCenter (lp6155...

Mapping

Destination - 'RDS (rdssc_dba on orcl_etan)'

Name: companycar

Source document type: companycar

Destination document type: COMPANYCARDst

□ Create a document or a collection element even if the source collection

Description:

Property	Value
Excluded fi...	ocmo, eventre
Node type	TABLE

Property	Value
Description	COMPANYCAR
Table	COMPANYCAR

Scheduling automatic data synchronization

The RDS has a set of pre-defined Connect-It scenario schedulers to run different synchronization tasks.

Because some of the synchronization tasks require more system resources than others, Peregrine recommends that you do not reconfigure them to occur more frequently than the default time intervals.

To schedule data synchronization, use the Connect-It Scheduler Editor.

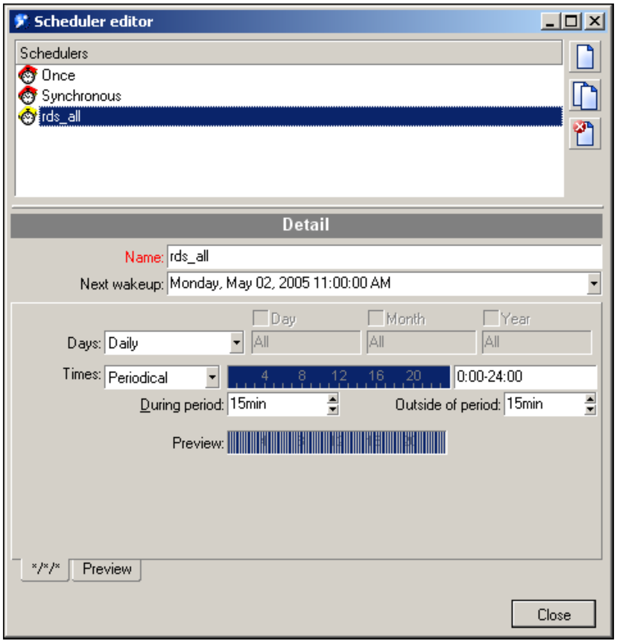
To schedule data synchronization using the Connect-It Scheduler Editor:

1 Click **Start** > **Programs** > **Peregrine** > **Connect-It** > **Service Console**.

2 Select the `rds_sc` scenario.

3 Click **Scheduling** to open the Connect-It Scheduling window.

4 Click **Edit Schedulers** to open the Connect-It Scheduler Editor.



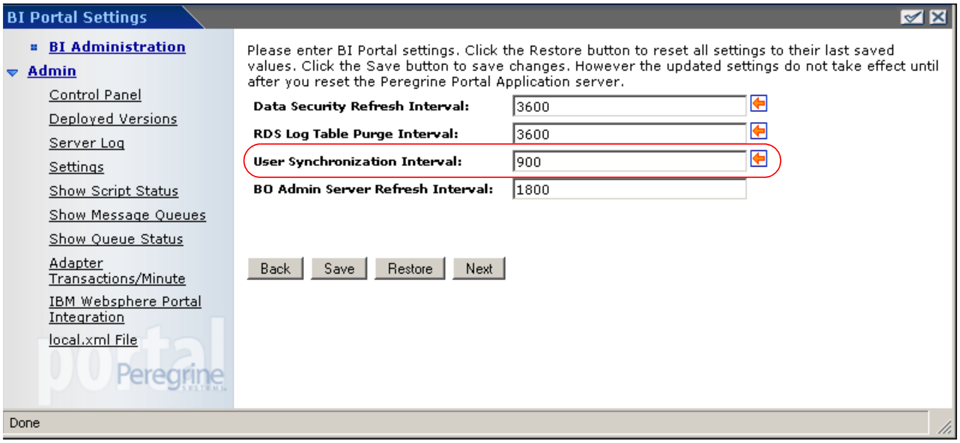
Name: rds_all
Next wakeup: Monday, May 02, 2005 11:00:00 AM
Days: Daily
Day: All
Month: All
Year: All
Times: Periodical
0:00-24:00
During period: 15min
Outside of period: 15min

The following defines a synchronization schedule.

Schedule	Description
`rds_all`	Synchronizes new and updated records at the default interval of once a day at midnight.

For more information about using the Connect-It Scheduler Editor, see the **Connect-It** documentation.

You can change the `rds_user` interval using these same steps. If you change the `rds_user` interval in the Connect-It scenario, you must also change the BI Portal Administrative setting for the **User Synchronization Interval** to match the rds_user interval. You change the **User Synchronization Interval** using BI Administration in the Administration module of the portal.



Please enter BI Portal settings. Click the Restore button to reset all settings to their last saved values. Click the Save button to save changes. However the updated settings do not take effect until after you reset the Peregrine Portal Application server.

Data Security Refresh Interval: 3600

RDS Log Table Purge Interval: 3600

User Synchronization Interval: 900

BO Admin Server Refresh Interval: 1800

Refer to the BI Portal Administration Guide for more information.

CHAPTER 6 Business Objects Universe Customization

This chapter explains how to use the Business Objects Designer tool to customize a universe file. The topics covered in this chapter are:

- Editing the universe on page 91
- Editing security profiles on page 103

Editing the universe

Once you customize the Reporting Data Store (RDS), you need to modify the metadata layer to expose this data to your end user. You do this by using the Business Objects Designer tool to customize the `rds.unv` file. The examples used in this section explain adding new table, adding a new field, and modifying an existing field. You then need to check the integrity of the data and export the `rds.unv` file to the Business Objects server.

Adding a new table

This section provides an example of how to add a new table to a universe file (`rds.unv`).

To add a new table:

1. Click **Insert** > **Tables** to open the Table Browser dialog box.
2. Select the `COMPANYCAR` table and then click **Insert**.

3 Click **Close** to close the window.

The table now appears in the right pane of the main window as a table object.

Next, you need to link this table to your existing tables. This allows reports to use multiple tables successfully.

4 Click **Insert** > **Join** to open the Edit Join dialog box.

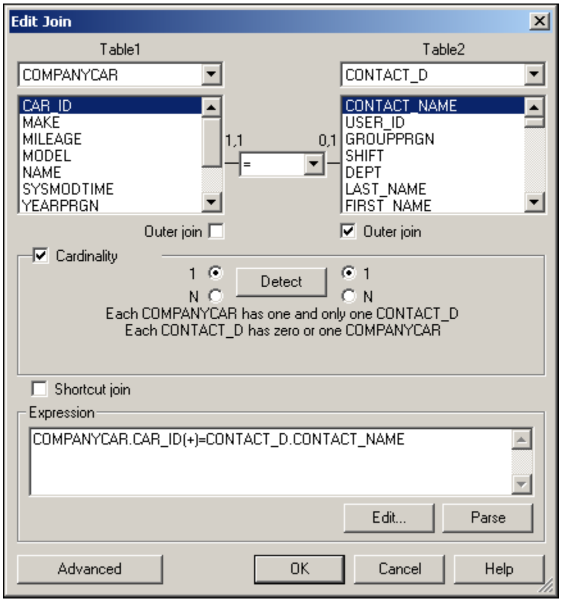


Table1: COMPANYCAR

CAR_ID
MAKE
MILEAGE
MODEL
NAME
SYSMODTIME
YEARPRGN

1,1 = 0,1

Table2: CONTACT_D

CONTACT_NAME
USER_ID
GROUPPRGN
SHIFT
DEPT
LAST_NAME
FIRST_NAME

Outer join □ ☑ Outer join

☑ Cardinality

1 ● ● 1
N ○ ○ N

Each COMPANYCAR has one and only one CONTACT_D
Each CONTACT_D has zero or one COMPANYCAR

□ Shortcut join

Expression

COMPANYCAR.CAR_ID(+)=CONTACT_D.CONTACT_NAME

5 In Table 1, select the first table of the join from the drop-down list and then click the field that linked this table from the list. In this example, link from the `COMPANYCAR` table on the field `CAR_ID`.

6 In Table 2, select the RDS table and the field linked to this table. In this example, it is `CONTACT_D`, using the field `CAR_ID`.

Between the two tables, there is a drop-down list box that specifies the qualifier for the join. If you are not sure which to use, default to the equal sign (=).

Beneath each table, there is a check box for an Outer Join. Use outer joins when you want your query to return all records from the outer table, regardless of whether or not there is a matching record in the other table. Inner joins only return records when there is a match in both tables. Since it is possible for a contact to exist without a company car, use an outer join on the contact side for this example. Click the appropriate check box.

For more information on Inner vs. Outer Joins, consult your DBA or database documentation.

7 Click **Detect** to determine the Cardinality of your join, then read the text that appears beneath the button that describes the join. If this is not correct, change the Cardinality as appropriate.

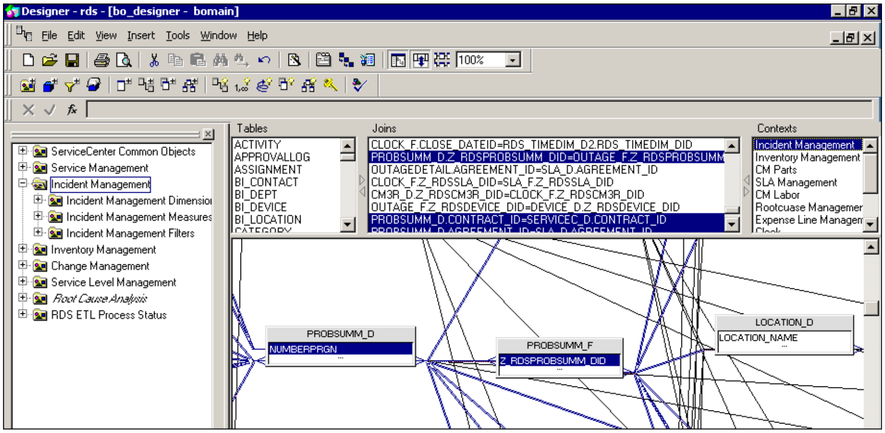
8 When you are satisfied with all the details of the link, click OK.

You see a line drawn from `COMPANYCAR` to `CONTACT_D`. You can right-click on this line and select Join Properties at any time to edit the settings again.

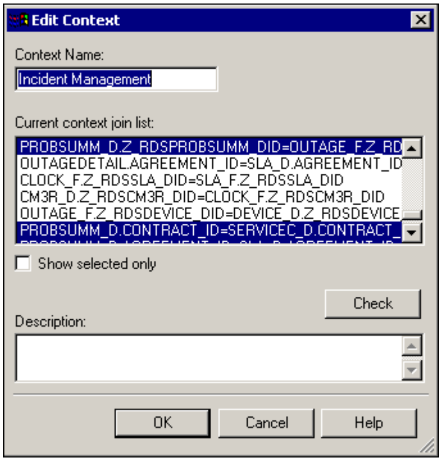
After creating the new link, you must make sure that the new join relationships are added to the existing Context.

To edit the Context:

1 From the Business Objects Designer tool, click **View** > **List Mode**.


Designer - rds - [bo_designer - bomain]
File Edit View Insert Tools Window Help
100%
ServiceCenter Common Objects
Service Management
Incident Management
Incident Management Dimensio
Incident Management Measures
Incident Management Filters
Inventory Management
Change Management
Service Level Management
Root Cause Analysis
RDS ETL Process Status
Tables
ACTIVITY
APPROVALLOG
ASSIGNMENT
BI_CONTACT
BI_DEPT
BI_DEVICE
BI_LOCATION
Joins
CLOCK_F.CLOSE_DATEID=RDS_TIMEDIM_D2.RDS_TIMEDIM_DID
PROBSUMM_D.Z_RDSPROBSUMM_DID=OUTAGE_F.Z_RDSPROBSUMM
OUTAGEDETAIL.AGREEMENT_ID=SLA_D.AGREEMENT_ID
CLOCK_F.Z_RDSSLA_DID=SLA_F.Z_RDSSLA_DID
CM3R_D.Z_RDSCM3R_DID=CLOCK_F.Z_RDSCM3R_DID
OUTAGE_F.Z_RDSDEVICE_DID=DEVICE_D.Z_RDSDEVICE_DID
PROBSUMM_D.CONTRACT_ID=SERVICEC_D.CONTRACT_ID
Contexts
Incident Management
Inventory Management
CM Parts
SLA Management
CM Labor
Rootcuase Managemer
Expense Line Managem
PROBSUMM_D
NUMBERPRGN
PROBSUMM_F
Z_RDSPROBSUMM_DID
LOCATION_D
LOCATION_NAME

2. Highlight the Context you want to edit and right-click **Context Properties** to open the Edit Context dialog box.



Context Name: Incident Management

Current context join list:

```
PROBSUMM_D.Z_RDSPROBSUMM_DID=OUTAGE_F.Z_RD
OUTAGEDETAIL.AGREEMENT_ID=SLA_D.AGREEMENT_ID
CLOCK_F.Z_RDSSLA_DID=SLA_F.Z_RDSSLA_DID
CM3R_D.Z_RDSCM3R_DID=CLOCK_F.Z_RDSCM3R_DID
OUTAGE_F.Z_RDSDEVICE_DID=DEVICE_D.Z_RDSDEVICE
PROBSUMM_D.CONTRACT_ID=SERVICEC_D.CONTRACT_
```

Description:

3. Add the new link into the existing context and click **OK**.

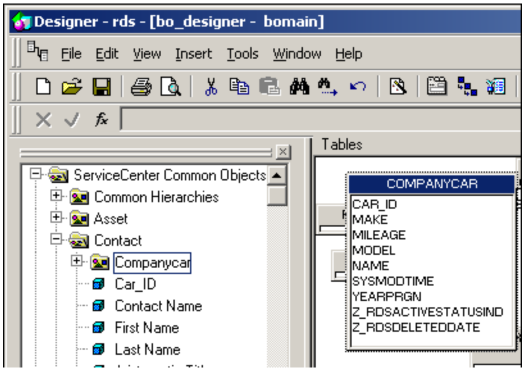
Tip: Hold the Ctrl key when adding the link so that you keep the existing links.

You need to add this table to the object browser (right pane). This makes the fields available to be selected for a report. This can be done individually, or all at once. For this example, create a folder under the Contacts folder.

To add this table to the object browser:

1. Expand the folders to open the **ServiceCenter Common Objects** > **Contact** folder.

2 Click and drag the COMPANYCAR table from the right pane to the Contact folder.



This automatically creates a folder containing all the fields in that table.

3 To rename any object, right-click it and select **Class Properties** (for folders) or **Object Properties** (for fields). Edit the name or description and click **OK**.

Note: As a best practice, Peregrine recommends that you include a common string of text in the descriptions of all universe objects you add or edit so that a quick search can identify them during future upgrades. For example, you can describe `Car ID` as `Unique identifier that links a contact to a company car (CUSTOM OBJECT)`. A quick search on `(CUSTOM OBJECT)` in the description shows all personalized items.

4 To hide any field from end-users, right-click the object and select **Hide Item(s)**.

5 To move any object, click and drag it to the desired folder.

Adding a new field

This section provides an example of how to add a new field to a universe file (`rds.unv`).

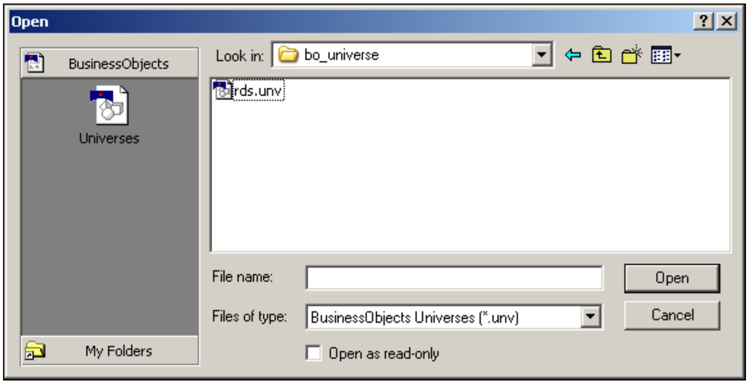
To edit the rds.unv file:

1 Log on to your Business Objects Designer Tool as a user with Designer rights.

2 Make a copy of your universe file (`rds.unv`). This is in the `<BO_Installation>\nodes\<server>\<cluster>\universes\<domain>\` `directory`.

Note: The backup copy is in addition to the copy you made in the customization workflow. Edit this copy. Do not edit the production copy.

3 In the Designer window, click **File** > **Open** and navigate to the universe file (`rds.unv`).



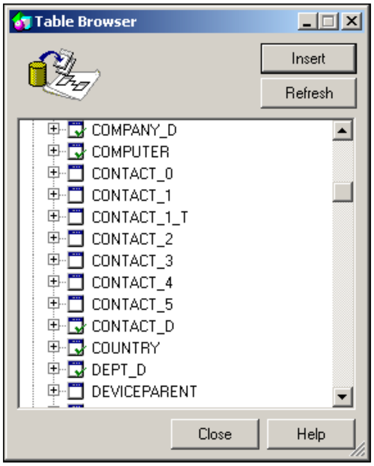
4 In the Open dialog box, select the universe file (`rds.unv`) and click **Open**.

In this example, you add one new field to the CONTACT_D table.

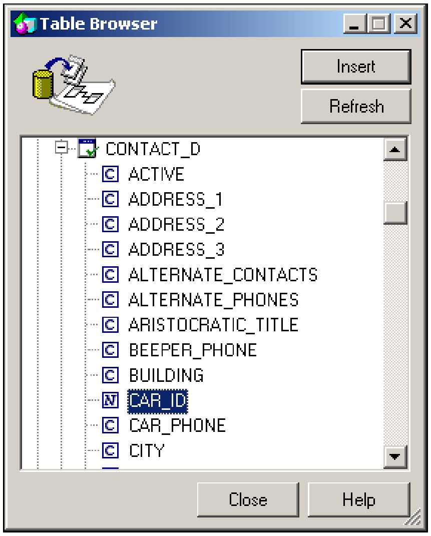
To add a new field:

1 From the Designer window, click **Insert** > **Table**.

The Table Browser dialog box opens and displays a list of all tables in the RDS.



2 Click the plus sign (+) next to `CONTACT_D` to expand the table and see individual fields.



```
CONTACT_D
  ACTIVE
  ADDRESS_1
  ADDRESS_2
  ADDRESS_3
  ALTERNATE_CONTACTS
  ALTERNATE_PHONES
  ARISTOCRATIC_TITLE
  BEEPER_PHONE
  BUILDING
  CAR_ID
  CAR_PHONE
  CITY
```

3 Click the new field (in this case, `CAR_ID`) and then click **Insert** to add the field to the universe.

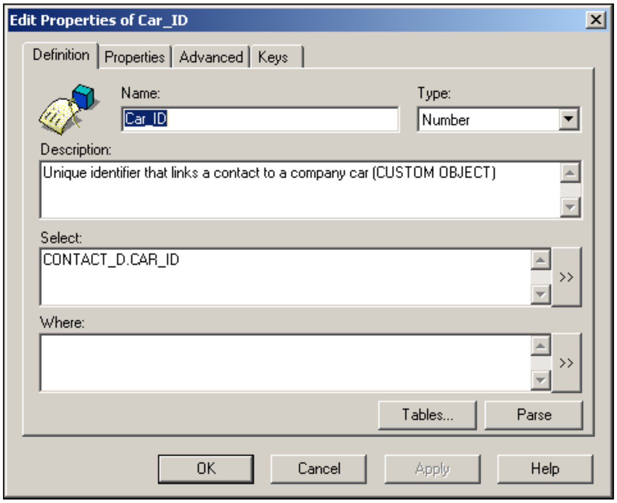
4 Click **Close** to close the window.

Next, you need to add this field to the object browser (right pane). This makes the field available to be selected for a report.

To add this field to the object browser:

1 In the Designer window, expand the folders to open the **ServiceCenter Common Objects** > **Contact** folder.

2 Right-click the **Contact** folder and select **Object**.



Edit Properties of Car_ID

Name: Car_ID

Type: Number

Description: Unique identifier that links a contact to a company car (CUSTOM OBJECT)

Select: CONTACT_D.CAR_ID

Where:

3 In the Edit Properties window, type a name in the Name text box, for example `Car ID`.

4 In the Type drop-down list box, select a type, for example, `Number`.

5 Type a description for the field in the Description text box.

Note: As a best practice, Peregrine recommends that you include a common string of text in the descriptions of all universe objects you add or edit, so that a quick search can identify them during future upgrades. For example, label the description for `Car ID` as `Unique identifier that links a contact to a company car (CUSTOM OBJECT)`. In the future, you can search on `(CUSTOM OBJECT)` in the description to reveal all personalized items.

6 In the Select text box, either browse for the field or type in the field in the form `<tablename>.<fieldname>`; for example, `CONTACT_D.CAR_ID`.

7 If necessary, you can add filters in the Where text box.

8 Click **Parse** to test your object.

If the test is successful, click **OK** to apply changes and close this dialog. Otherwise, re-enter any erroneous entries.

9 If necessary, click and drag the new object to a different folder or location.

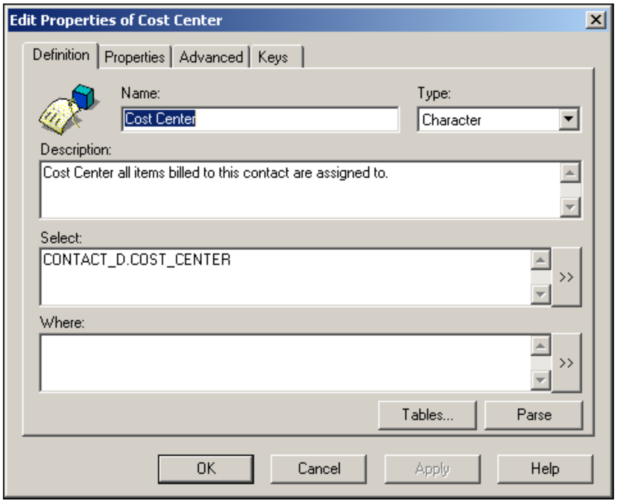
Modifying an existing field

This section provides an example of how to modify an existing field in a universe file (`rds.unv`).

To modify an existing field:

1 Browse to the field in the left pane.

2 Right-click and select **Object Properties**.



Name: Cost Center

Type: Character

Description: Cost Center all items billed to this contact are assigned to.

Select: CONTACT_D.COST_CENTER

Where:

a In the Edit Properties dialog box, edit the Name, data Type, Description, and database field as needed.

b In this example, change `Cost Center` to `My Cost Center`.

c In the Select text box, change the statement to `CONTACT_D.MY_COST_CENTER`.

d Click **Parse** to test the changes.

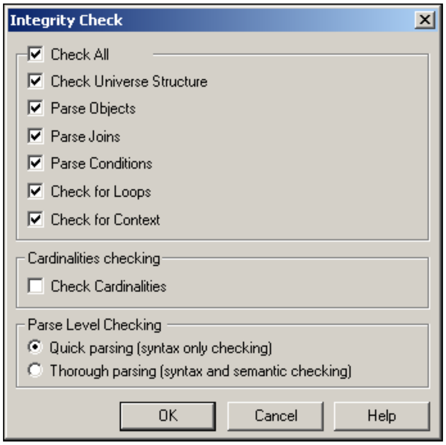
3 If successful, click **OK** to save the changes; otherwise, make the necessary changes in the dialog box and repeat the test.

Checking database integrity

Once you make all your changes, you need to do an integrity check to make sure all database links are valid.

To do an integrity check:

1 Click **Tools** > **Check Integrity**.



☑ Check All
☑ Check Universe Structure
☑ Parse Objects
☑ Parse Joins
☑ Parse Conditions
☑ Check for Loops
☑ Check for Context

Cardinalities checking
☐ Check Cardinalities

Parse Level Checking
● Quick parsing (syntax only checking)
○ Thorough parsing (syntax and semantic checking)

2 Click **Check All**.

3 Click **OK**.

The utility runs though the Universe and suggests possible problem areas.

4 Correct any problems as needed.

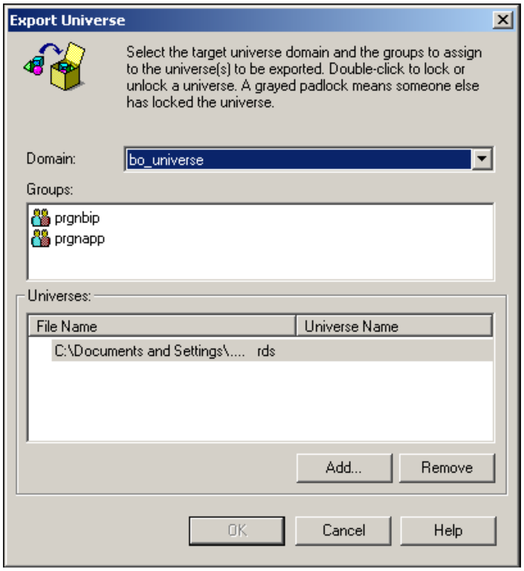
5 Save your changes.

Exporting the universe

Once you complete your changes and verify the integrity of the database, you are ready to export the changes to the Business Objects server.

To export the universe to the Business Objects server:

1 Click **File** > **Export**.



Export Universe

Select the target universe domain and the groups to assign to the universe(s) to be exported. Double-click to lock or unlock a universe. A grayed padlock means someone else has locked the universe.

Domain: bo_universe

Groups:
prgnbip
prgnapp

Universes:

File Name	Universe Name
C:\Documents and Settings\....	rds

2 In the Export Universe dialog box, click the appropriate Domain from the Domain drop-down list.

3 In the Groups list, select an application group; in this example, `prgnapp`.

Note: Export the universe to the application group.

4 Click **OK** to export the universe.

You see a message indicating a successful export.

Note: If you changed any existing objects, Peregrine recommends that you check existing reports on your server that may reference those objects. You may need to edit the query to reflect your new changes.

Editing security profiles

This is an optional procedure, only required if you set up row level security in Business Objects. For additional information, see the BI Portal Administration Guide.

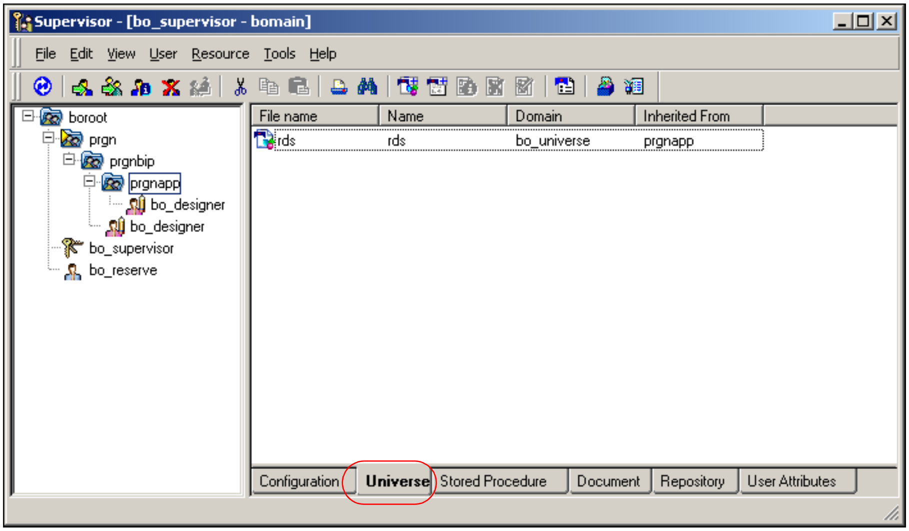
To edit a security profile:

1. Log on to the Business Objects Supervisor as a user with Supervisor rights.
2. Select the group at the level you want to set your restrictions. If you want to restrict all users, go to your main application group. If you want to impose different restrictions for different sets of users, you can:
   - Create groups for specific restrictions under the application group.
   - Add users to groups you created.

To set security restrictions on the sub-group:

1. Highlight the sub-group.

2 From the Supervisor window, click the **Universe** tab at the bottom of the window.



File name	Name	Domain	Inherited From
rds	rds	bo_universe	prgnapp

3 From the **Universe** tab, right-click the `rds` universe file, select **Properties, then** click **Rows** > **Add** to create a new restriction.

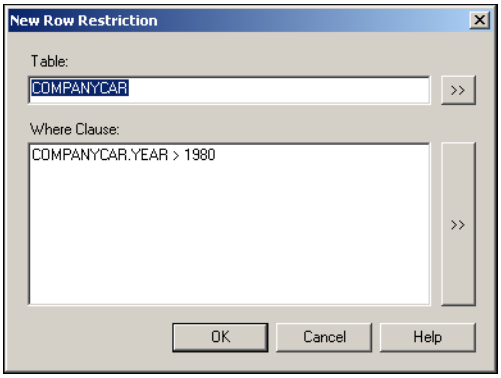


Table: COMPANYCAR

Where Clause: COMPANYCAR.YEAR > 1980

a Type in a table name or click the symbol (>>) beside the Table field to browse for the table you want to add restrictions to.

In this example, use `COMPANYCAR`.

b In the **Where Clause** field, type an SQL clause or click the symbol (>>) to build one with the formula editor. For help with SQL syntax, contact your DBA or database documentation.

This example restricts access to all records in the `COMPANYCAR` table that refer to cars built before 1980 (`COMPANYCAR.YEAR > 1980`).

4 Click **OK** when done.

5 Click **Check All** to verify the validity of the SQL you entered.

The Status field changes to OK.

6 Click **OK** to close this dialog box.

Note: Changes are instantaneous. You do not need to save or reset the server for changes to take effect.

To test the restriction:

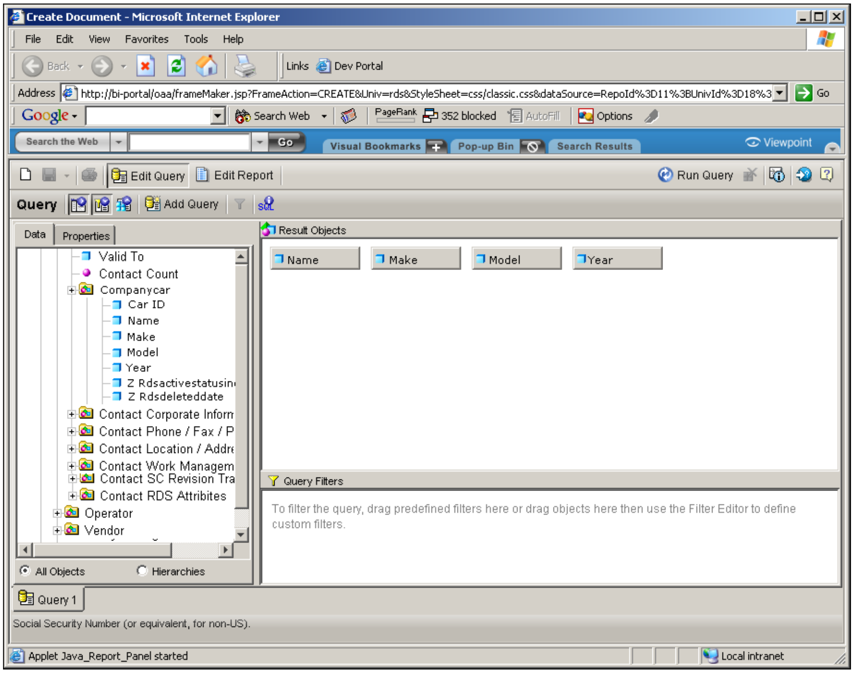
1 Log on to BI Portal as a user with report creation rights.

Note: Make sure this user belongs to the group that you placed the restriction on.

2 Click the **Reporting** icon.

3 Click **Create**.

4 Click the `rds` universe to launch the Report Designer tool.



Address: http://bi-portal/oaa/frameMaker.jsp?FrameAction=CREATE&Univ=rds&StyleSheet=css/classic.css&dataSource=RepoId%3D11%3BUnivId%3D18%3

○ All Objects ○ Hierarchies

To filter the query, drag predefined filters here or drag objects here then use the Filter Editor to define custom filters.

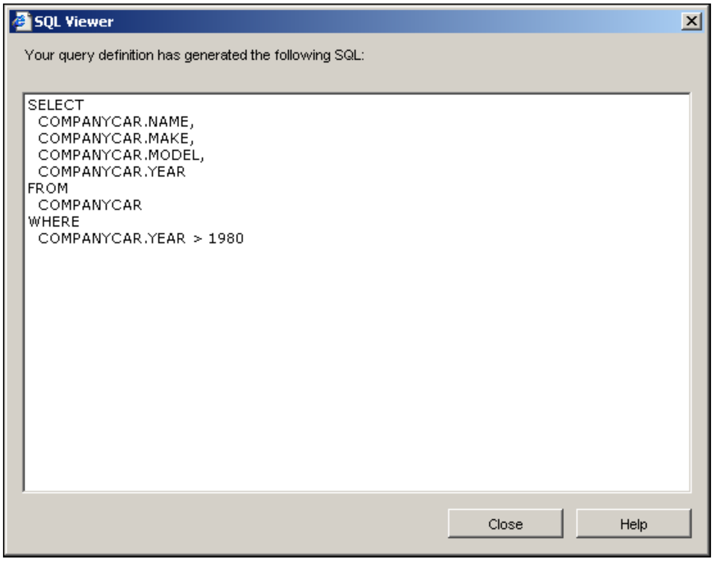
Social Security Number (or equivalent, for non-US).

Applet Java_Report_Panel started

5 Navigate to the appropriate object in the universe and drag fields into the Result Objects pane.

6 Click **View SQL** to see the SQL statement behind the report.

The restrictive clause you added is at the bottom of the report.



Your query definition has generated the following SQL:

```
SELECT
  COMPANYCAR.NAME,
  COMPANYCAR.MAKE,
  COMPANYCAR.MODEL,
  COMPANYCAR.YEAR
FROM
  COMPANYCAR
WHERE
  COMPANYCAR.YEAR > 1980
```

To add additional access restrictions, repeat the steps in this section, as required.

A RDS System Tables

APPENDIX

This appendix provides descriptions and samples of the system tables found in the RDS. The topics include:

- Tracking RDS schema metadata
- RDS environment tables on page 111
- RDS System tables for tracking ETL processing on page 112
- Other useful tables on page 114

Tracking RDS schema metadata

The following RDS tables track RDS schema metadata.

- `RDS_UNIQUECOLUMN`
- `RDS_SCDCOLUMN`

RDS_UNIQUECOLUMN table

The `RDS_UNIQUECOLUMN` table tracks all the unique columns for the reporting tables. It also includes unique fields for the ServiceCenter and AssetCenter source files. The RDS schema creation program uses this information to create unique indexes for RDS reporting tables. The RDS Connect-It scenarios use the unique field values to decide about adding new records or updating the existing records in the ETL process.

The RDS_UNIQUECOLUMN table columns in an Oracle database contain the following information.

Name	Null?	Type	Description
LRDS_UNIQUECOLUMN_ID	Not NULL	NUMBER(10)	Unique system key.
TABLE_NAME	Not NULL	VARCHAR2(80)	RDS Report table name. RDS table type as one of following values: ASSOCIATE DIMENSION
TABLE_TYPE	Not NULL	VARCHAR2(80)	DIRECT_MAPPING
DS_TABLE_NAME	Not NULL	VARCHAR2(30)	Source file name. File type as on the following values: DATA_SOURCE (as a source file) PARENT_DIM **Note:** Use PARENT_DIM information for cascading delete. When the record in the table (that is, the value of DS_TABLE_NAME) is deleted from the data source, the corresponding record in the RDS table (that is, the value of TABLE_NAME) is marked as inactive.
SRC_UNIQUECOLUMN	Not NULL	VARCHAR2(80)	
RDS_UNIQUECOLUMN	Not NULL	VARCHAR2(80)	
TYPE_UNIQUECOLUMN	Not NULL	VARCHAR2(30)	
TABLE_UNIQUECOLUMN	Not NULL	VARCHAR2(10)	

RDS_SCDCOLUMN table

The RDS_SCDCOLUMN table tracks all the SCD columns for the dimension tables. The RDS Connect-It scenario uses this information to decide which updating to use: the updating as simple overwrite updating or SCD Type Two updating in the ETL process. The SCD Type Two updating creates a new record and marks the existing record as old in dimension tables. Refer to data warehouse technical sources for more information about SCD Type Two.

The RDS_SCDCOLUMN table columns in an Oracle database contain the following information

Name	Null?	Type	Description
LRDS_SCDCOLUMN_ID	Not NULL	NUMBER(10)	Unique system key.
TABLE_NAME	Not NULL	VARCHAR2(80)	RDS dimension table name.
SRC_SCDCOLUMN	Not NULL	VARCHAR2(80)	ServiceCenter source field name.
RDS_SCDCOLUMN	Not NULL	VARCHAR2(80)	RDS column name.
TYPE_SCDCOLUMN	Not NULL	VARCHAR2(30)	Column type.
TABLE_SCDNUMBER	Not NULL	NUMBER(10)	The sequence number of the SCD column if there are more than one SCD columns.

RDS environment tables

This section describes the tables that store data about the RDS environment.

RDS_DBINFO table

The RDS_DBINFO table stores the RDS deployment environment. The following table shows sample data.

Z_RDS_DBINFO_ID	OSTYPE	DATABASETYPE	DATABASENAME	DATABASEURL	USERID	USERPASSWORD
1	Windows	Oracle		jdbc:oracle:oci8:@rds	rds_dba	passw0rd

RDS Security System Table - RDS_USER

The RDS_USER table stores all the user information for BI Portal. If contact based authentication is turned on, the data source can be the ServiceCenter *operator* table and *contacts* table.

The RDS_User table contains the following information.

Name	Null?	Type	Description
RDS_USER_ID		FLOAT	Unique system key.
DS_UNIKEY_NAME		VARCHAR2(120)	The unique key value. For the operator file, the value is the operator name field value.

Name	Null?	Type	Description
CAPABILITIES		VARCHAR2(500)	
APP_NAME		VARCHAR2(10)	
DS_TYPE		VARCHAR2(30)	
EMPLID		VARCHAR2(140)	The join field value for contact with operator.
ROLEGROUP_ID		FLOAT	RDS_USER_ID from the operator record, which is linked with the contact.
Z_RDSLASTMODDATE		DATE	Time the record was last modified.
Z_RDSACTIVESTATUSIND		VARCHAR2(1)	The user record active status; Y or N.
Z_RDSDELETEDATE		DATE	Record deleted date time.

RDS System tables for tracking ETL processing

The following tables track RDS ETL processing status.

- RDS_CIT
- RDS_CIT_LOG
- RDS_ETL_WORK
- RDS_ETLSYNC_LOG
- RDS_SEC_SYNC
- RDS_SEQUENCE

The RDS_CIT and RDS_ETL_WORK tables are used for the RDS ETL (Extracting, Transforming and loading) process as RDS internal tracking system tables.

RDS_CIT_LOG

RDS_CIT_LOG has the following columns.

- ID
- STATE_FLAG
- CIT_TIME

RDS_CIT_LOG STATE_FLAG

The RDS_CIT_LOG STATE_FLAG shows the ETL process status as one of the following values.

Value	Description
CIT_START	RDS Connect-It scenario initial sync or periodic sync starts.
RDS_SYNC_START	RDS fact tables population starts.
RDS_AGG_START	RDS aggregation tables population starts.
RDS_HIER_START	RDS hierarchy tables population starts.
COMPLETED	ETL process is completed.

The STATE_FLAG shows COMPLETED after the ETL process is done.

RDS_SEC_SYNC table

The RDS_SEC_SYNC table records the last time a different BI Portal user performed a synchronization.

RDS_SEQUENCE table

The RDS_SEQUENCE table stores the next available sequence value for the unique system keys for all the RDS tables.

RDS_ETLSYNC_LOG table

The RDS_ETLSYNC_LOG provides very useful information for the ETL sync volume. This table contains the following information.

Name	Null?	Type	Description
RDS_ETLSYNC_LOG_ID	NOT NULL	NUMBER	Unique system key
TABLE_NAME	NOT NULL	VARCHAR2(80)	Populated table name
TABLE_ACTION	NOT NULL	VARCHAR2(80)	ETL process updating action codes: Add Update Update with SCD Delete Log
SYNC_RECORD_COUNT	NOT NULL	NUMBER	Total record number of the updating described by the action code
SYNC_DATETIME		DATE	The sync date time
Z_RDSCREATEDDATE		DATE	The track record creation date time

Other useful tables

This section describes other system tables that provide useful information.

RDS time dimension table

The RDS time dimension table is a unique and powerful dimension in the RDS. The RDS time dimension table provides the mechanism for BI Portal reporting based on days, months, quarters and years. All RDS aggregation tables end with _AGG. Table name conventions are based on weeks, quarters and months, and years. RDS_TIMEDIM_D stores all the calendar day from 1995 to 2010. The start-year and end-year values are set by RDS installer. The values are saved in RDS \conf directory, rds.properties file as:

```
rds.startDate=1995
rds.endDate=2010
```

The RDS time dimension table columns contains the following information.

Name	Null?	Type
RDS_TIMEDIM_DID	Not NULL	NUMBER
FULLDATE		VARCHAR2(20)
WEEK		VARCHAR2(10)
MONTH		VARCHAR2(20)
QUARTER		VARCHAR2(4)
YEAR		VARCHAR2(10)
FISCALPERIOD		VARCHAR2(4)
LASTDAYINMONTHFLAG		CHAR(1)
FULLDATE_D		DATE
MONTH_NUMERIC		NUMBER

B Copyright Notices

APPENDIX

Peregrine Systems acknowledges the copyrights belonging to the following third parties. (This appendix constitutes a continuation of the copyright page.)

Notices

iso-relax.jar

The following software may be included in this product: `iso-relax.jar` v2002/07/07; This software may be obtained under the terms of the license below:

MIT license

Copyright © Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

relaxngDatatype.jar

The following software may be included in this product: `relaxngDatatype.jar` v1.0; This software may be obtained under the terms of the license below:

Jcrypt.java

This product includes software developed by Eric Young (*eay@mincom.oz.au*).

JAXP

See the W3C license and the Apache version 2 license below.

W3C

W3C IPR SOFTWARE NOTICE

Copyright © 2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires “Notice of any changes or modifications to the W3C files, including the date changes were made.” Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL binding, the pragma prefix can no longer be 'w3c.org'; in the case of the Java binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the W3C Software Copyright Notice and License could be found at
http://www.w3.org/Consortium/Legal/copyright-software-19980720

Copyright © 1994-2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/

This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and modify this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: “Copyright © [$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. *http://www.w3.org/Consortium/Legal/*”

3. Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED “AS IS,” AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Apache Version 1.1

The Apache Software License, Version 1.1

Copyright © 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

> “This product includes software developed by the Apache Software Foundation (*http://www.apache.org/*).”

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names “Xalan” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact *apache@apache.org*.

5. Products derived from this software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright © 1999, Lotus Development Corporation., *http://www.lotus.com*. For more information on the Apache Software Foundation, please see *<http://www.apache.org/>*.

Apache Version 2

Apache License, Version 2.0, January 2004, *http://www.apache.org/licenses/*

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions.**

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution**. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions**. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability**. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. A**ccepting Warranty or Additional Liability**. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own

Index

A

- aggregate keys 30
- aggregate of fact tables 15
- aggregate table 16
- aggregateKeys 30
- associate table 15, 16, 25
- automatic data synchronization 87

B

- BIRECORDDELETE 15
- bridge table 15, 16

C

- configure scenarios 40
- Connect-It console
  - configure scenarios 40
  - start scenarios 38
  - stop scenarios 39
- Connect-It scenario
  - add a new field to rds_sc.scn 78
  - add mapping for a new table 68
  - add the new table to the scheduling process 77
  - Builder tool 66
  - customize 65
  - data synchronization process 82
  - edit 50
  - primary key field 76
  - schedule synchronization 87
  - update mapping information for a field 83
- customer support 10

D

- data structures
  - add 47
  - modify and delete 47
- data synchronization 32, 35, 87
- device table 22
- dimension attributes 31
- dimension table 14, 16, 21, 25
- direct mapping table 15, 16, 27, 29
- directMapping table 28
- document groups 13, 14

E

- ETL mapping information 17

F

- fact table 15, 16, 23
- fact table definition 25
- FactType table 24

H

- hierarchy table 15, 26

J

- join table 15

L

- log file 38

M

- metadata 16

N

naming conventions 16

P

Peregrine Systems customer support 10
planning the customization 46
primary key field 76

R

rdms_keywords.xml 49
RDS scenario 66
RDS scenarios 35
RDS schema customization
  add a new field 57
  add a new table 54
  change an existing field 59
RDS schema, editing 49
RDS time dimension table 114
rds.unv 49, 91, 95, 96
rds_ac.log 38
RDS_CIT table 112
RDS_CIT_LOG STATE_FLAG table 113
RDS_CIT_LOG table 112
RDS_DBINFO table 111
RDS_etl.xml 13, 16, 20, 32, 53, 54, 56, 58
rds_etl.xml 16, 49
rds_etl.xsd 16, 20, 28
RDS_ETL_WORK table 112
RDS_ETLSYNC_LOG table 114
rds_sc scenario 36, 39, 40, 48
rds_sc.ini 36
rds_sc.log 36, 38
rds_sc.scn 35, 36, 49, 50, 65, 66
RDS_SCDCOLUMN table 110
RDS_SEC_SYNC table 113
RDS_SEQUENCE table 113
RDS_UNIQUECOLUMN table 109
rds_user scenario 36, 39, 43, 48
RDS_USER table 111
rds_user.ini 36
rds_user.scn 35, 36, 49
RDSInitRun command 32
record deletion, ServiceCenter 32

S

schedule Connect-It scenarios 87
schema data 16
schema definition 22, 24, 28
security profiles, editing 51
ServiceCenter record deletion 32
ServiceCenter triggers 60
start RDS scenarios 38
stop RDS scenarios 39
synchronize
  automatic scheduling 87
  data 35
  record deletion 32
  sysmodtime field 55
sysmodtime 55
system table 15, 16

T

tables
  aggregate of fact 15
  associate 15
  bridge 15
  dimension 14
  direct mapping 15
  fact 15
  hierarchy 15
  join 15
  system 15
  transaction 15
technical support 10
transaction table 15
triggers
  add a post-delete 62
  delete 63
  predefined in ServiceCenter 60
typographical conventions 9

U

universe 19
  add a field to the object browser 98
  add a new field 97
  add a new table 91
  add a table to the object browser 94
  edit 95
  edit a security profile 103

- edit the Context 93
- export to Business Objects 102
- modify an existing field 100
- test user rights 105
- validate the database links 101

universe, editing 50

W

workflow

- back up critical files 48
- close programs and processes 48
- edit Connect-It scenario 50
- edit security profiles 51
- edit the RDS schema 49
- edit the universe 50
- planning 46
- restart programs and processes 51

X

XML schema

- definition 31
- syntax 16

Z

Z_RDSACTIVESTATUSIND 31, 32

Z_RDSCREATEDDATE 31

Z_RDSDELETEDDATE 31

Z_RDSLASTMODDATE 31, 32

Z_RDSOBSOLETEIND 32

Z_RDSTRANSENDDATE 31, 32

Z_RDSTRANSLASTIND 32

Peregrine
SYSTEMS.

6/9/05