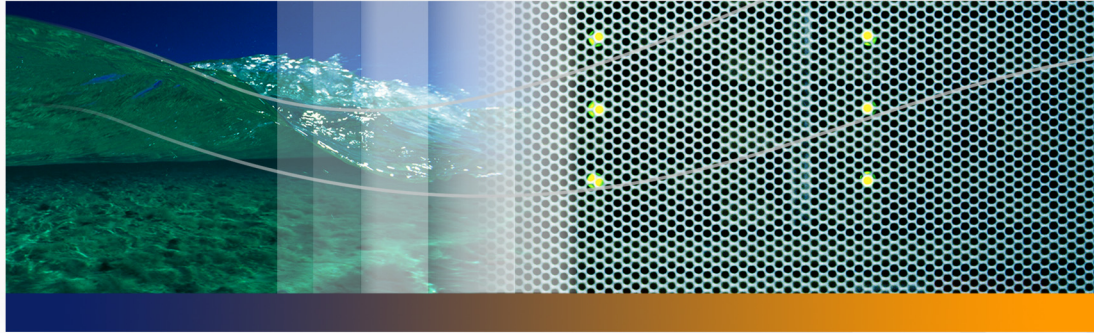


Peregrine Systems, Inc.

# Reporting Data Store 5.2 for AssetCenter



## Administration Guide



© Copyright 2005 Peregrine Systems, Inc.

PLEASE READ THE FOLLOWING MESSAGE CAREFULLY BEFORE INSTALLING AND USING THIS PRODUCT. THIS PRODUCT IS COPYRIGHTED PROPRIETARY MATERIAL OF PEREGRINE SYSTEMS, INC. ("PEREGRINE"). YOU ACKNOWLEDGE AND AGREE THAT YOUR USE OF THIS PRODUCT IS SUBJECT TO THE SOFTWARE LICENSE AGREEMENT BETWEEN YOU AND PEREGRINE. BY INSTALLING OR USING THIS PRODUCT, YOU INDICATE ACCEPTANCE OF AND AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THE SOFTWARE LICENSE AGREEMENT BETWEEN YOU AND PEREGRINE. ANY INSTALLATION, USE, REPRODUCTION OR MODIFICATION OF THIS PRODUCT IN VIOLATION OF THE TERMS OF THE SOFTWARE LICENSE AGREEMENT BETWEEN YOU AND PEREGRINE IS EXPRESSLY PROHIBITED.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. This book, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc. This document refers to numerous products by their trade names. In most, if not all, cases these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

Peregrine Systems, AssetCenter, AssetCenter Web, BI Portal, Dashboard, Get-It, Peregrine Mobile, and ServiceCenter are registered trademarks of Peregrine Systems, Inc. or its subsidiaries.

Microsoft, Windows, Windows 2000, SQL Server, and names of other Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation. Oracle is a registered trademark of Oracle Corporation. DB2 is a registered trademark of International Business Machines Corp. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). This product also contains software developed by: Sun Microsystems, Inc., Netscape Communications Corporation, and InstallShield Software Corporation. If additional license acknowledgements apply, see the appendix. This product includes code licensed from RSA Data Security.

This product includes software developed by Business Objects, S.A. Portions, copyright 1995 - 2004, Business Objects, S.A. All rights reserved.

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](mailto: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](mailto: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](http://www.peregrine.com)



# Contents

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 . . . . .	24
	Hierarchy table schema definition . . . . .	25
	Direct mapping table schema definition . . . . .	26
	Slowly changing dimension (SCD) keys schema definition . . . . .	28
	Aggregate keys schema definition . . . . .	29
	RDS dimension attributes . . . . .	29
	RDS InitRun command . . . . .	30
	Synchronizing AssetCenter record deletions with RDS. . . . .	31
Chapter 2	Using Connect-It to Handle RDS Data Synchronization . . . . .	33
	RDS data synchronization process . . . . .	33
	Asset types in the AMASSET_D table . . . . .	34
	Starting and stopping the RDS scenarios . . . . .	35
	Configuring the connectors . . . . .	38
Chapter 3	Customization Workflow . . . . .	43
	Workflow overview . . . . .	43
	Overview: Planning and information gathering . . . . .	44
	Overview: Stopping programs and processes . . . . .	46
	Overview: Backing up crucial files. . . . .	46
	Overview: Editing the RDS schema . . . . .	47

	Overview: Editing the Connect-It scenario. . . . .	48
	Overview: Editing the universe . . . . .	48
	Overview: Editing security profiles . . . . .	49
	Overview: Restarting programs and processes . . . . .	49
Chapter 4	RDS Customization . . . . .	51
	RDS schema customization. . . . .	51
	Adding a new table . . . . .	52
	Adding a new field . . . . .	55
	AssetCenter deletion workflow . . . . .	57
	AssetCenter VB script action definition . . . . .	58
	Content of an amOutputEvent record. . . . .	59
	Importing predefined deletion workflow into AssetCenter. . . . .	59
	Extracting record deletion information . . . . .	60
	Marking deleted records in the RDS. . . . .	60
Chapter 5	Connect-It Scenario Customization . . . . .	63
	Using CIT Scenario Builder tools to view an RDS scenario . . . . .	64
	Adding new mapping . . . . .	66
	Adding mapping for a new table . . . . .	66
	Adding a new field to an existing mapping . . . . .	76
	Test your customized RDS scenario . . . . .	80
	Scheduling automatic data synchronization. . . . .	80
Chapter 6	Business Objects Universe Customization . . . . .	83
	Editing the universe . . . . .	83

	Adding a new table . . . . .	83
	Adding a new field . . . . .	88
	Modifying an existing field . . . . .	92
	Checking database integrity . . . . .	93
	Exporting the universe . . . . .	94
	Editing security profiles . . . . .	96
Appendix A	RDS System Tables . . . . .	101
	Tracking RDS schema metadata . . . . .	101
	RDS_UNIQUECOLUMN table . . . . .	101
	RDS_SCDCOLUMN table . . . . .	102
	RDS environment tables . . . . .	103
	RDS_DBINFO table . . . . .	103
	RDS Security System Table - RDS_USER . . . . .	103
	RDS System tables for tracking ETL processing . . . . .	104
	RDS_CIT_LOG . . . . .	104
	RDS_SEC_SYNC table . . . . .	105
	RDS_SEQUENCE table . . . . .	105
	RDS_ETLSYNC_LOG table . . . . .	106
	Other useful tables. . . . .	106
	RDS time dimension table . . . . .	106
Appendix B	Copyright Notices . . . . .	109
	Notices . . . . .	109
	Index . . . . .	119



# About this Guide

This document is for advanced users who are going to customize the Reporting Data Store (RDS). If you intend to customize the `rdsac_et1.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 AssetCenter triggers customization.
Chapter 5, Connect-It Scenario Customization	Discusses how to customize the RDS Connect-It scenario, <code>rds_ac.scn</code> , for AssetCenter.
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 found 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
<a href="#">BI Portal User Guide</a>	Provides base reports and describes how to create and work with both base and custom reports.
<a href="#">BI Portal Installation Guide</a>	Describes how to install and configure the application and Web servers for Reporting Data Store.
<a href="#">BI Portal Administration Guide</a>	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.
<a href="#">RDS for ServiceCenter Administration Guide</a>	Provides information about customizing the RDS for ServiceCenter.
<a href="#">BI Portal Release Notes</a>	Includes last-minute enhancements, known issues, and closed issues.
<a href="#">WebIntelligence User's Guide</a>	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
<b>Bold</b>	Information that you must type exactly as shown appears in bold. The names of buttons, menus, and menu options also appear in <b>bold</b> .
<i>Italics</i>	Variables and values that you must provide are in <i>italics</i> . New terms and book titles also are in <i>italics</i> .
Monospace	Code or script examples, output, and system messages are in a monospace font. <pre>var msgTicket = new Message( "Problem" ); ... msgTicket.set( "_event", "epmc" );</pre> <p>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.</p> <p>Filenames, such as <code>login.asp</code>, appear in a monospace font.</p>

## 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
<b>Important:</b>	Information that is required to complete a task
<b>Note:</b>	Information that is of general interest
<b>Tip:</b>	Information that can make a task easier or faster
<b>Warning:</b>	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 Reporting Data Store and Schema

## CHAPTER

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 AssetCenter record deletions with RDS on page 31

**Note:** This section provides an introduction to the Reporting Data Store (RDS). If you customize the RDSac\_et1.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 AssetCenter transaction tables
- Aggregates of fact tables
- Hierarchy tables
- RDS system tables

### Dimension tables

Most dimension table fields are directly mapped from AssetCenter tables through Connect-It. Dimension tables add additional RDS ETL (Extract, Transform and Loading) attributes. AssetCenter 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 AssetCenter 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 AssetCenter transaction tables

BI Portal uses AssetCenter transaction tables for detail reporting and as search criteria for one AssetCenter application. These tables are directly mapped from AssetCenter 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 are built from the fullpath field in structured data. For example, the AMEMPLDEPT table fullpath field list  
Company/Department/Sub-department/Employee.

## 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 AssetCenter 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

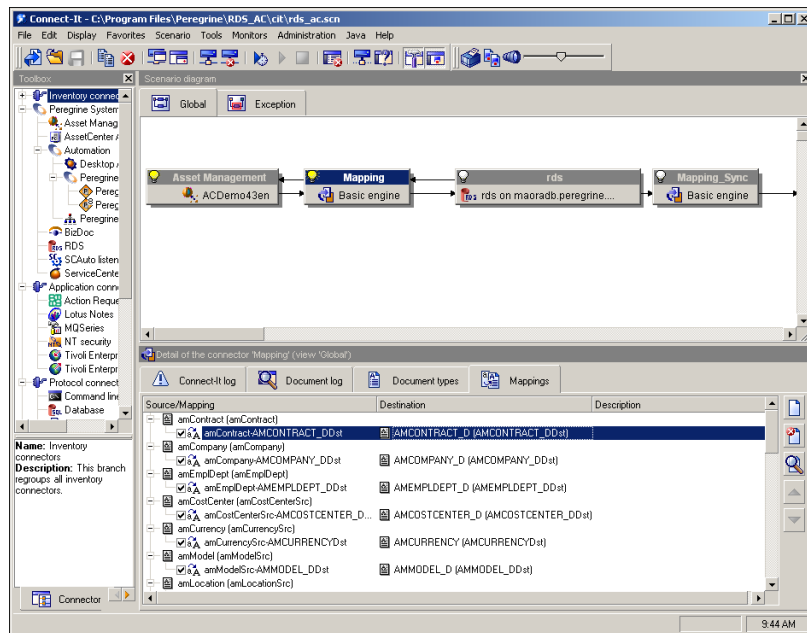
The `RDSac_et1.xml` provides the RDS schema data. The `rds_et1.xsd` defines the XML schema syntax. All these files are stored in `/conf` folder of the RDS installation directory.




## RDS ETL mapping information

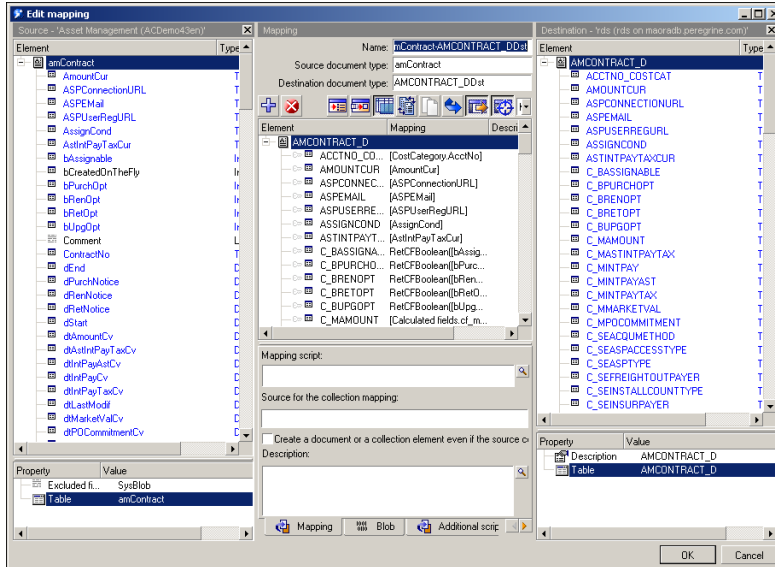
The Connect-It RDS scenario (`rds_ac.scn`) describes the metadata for the extraction, transformation and mapping. You can launch Connect-It Scenario Builder to open `rds_ac.scn` from the `/cit` folder of the RDS installation directory. You can use the Edit function to view any Source/Mapping and review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

The following diagram shows the mapping for the `amContract` table.



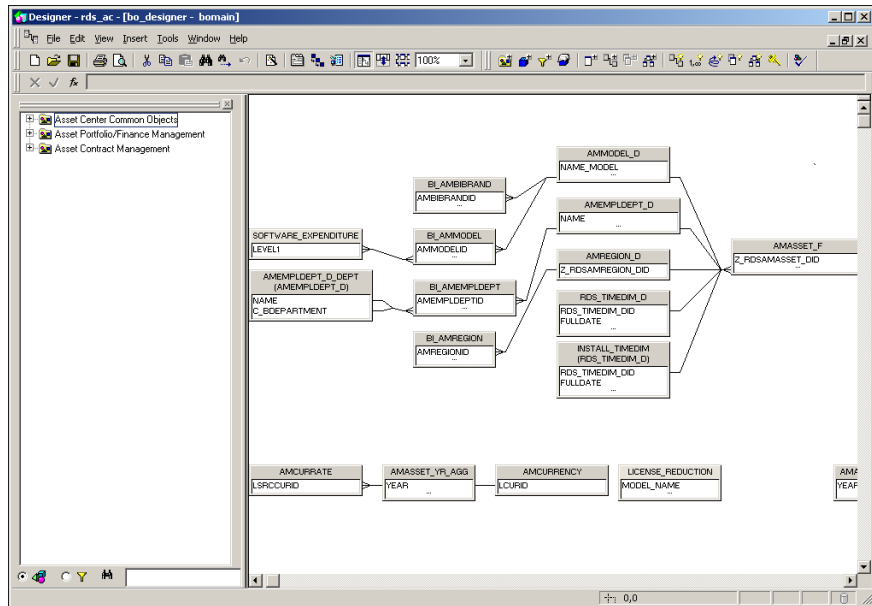
After you click **Edit a mapping**, the Edit Mapping window opens. 

The AssetCenter 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.



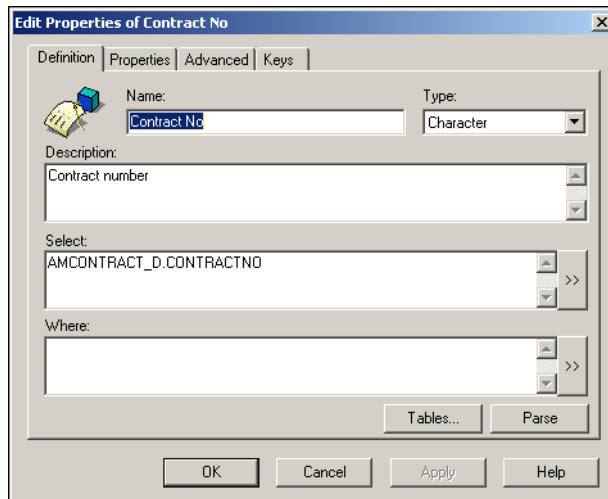
## RDS semantic layer — universe

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.



You can use the Business Objects designer tool to produce a PDF file of the universe metadata. The SupportFiles\AssetCenter\universe folder on the RDS for AssetCenter installation CD contains the rds\_ac.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 RDSac\_et1.xml file. This schema is built based on the XML schema definition defined in rds\_et1.xsd. These two files normally reside in the directory, /conf, under the RDS installation root directory. The detail of the rds\_et1.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_et1.xsd`.

The following example shows the RDS schema definition for the **amContract** table in AssetCenter.

```

<dimension name="AMCONTRACT" rdsVersion="5.2">
  <dimensionTableName>
    AMCONTRACT_D
  </dimensionTableName>
  <dataSourceTableName>
    amContract
  </dataSourceTableName>
  <dimensionTableFields>
    <dimensionTableField name="FIELD1" type="char" size="27"/>
    <dimensionTableField name="CONTRACTNO" type="char" size="20"/>
    <dimensionTableField name="LCNTRID" type="float"/>
    .
    .
  </dimensionTableFields>
  <uniqueKeys name="amcontract_unique">
    <uniqueKey fieldName="LCNTRID_KEY" srcFieldName="lCntrId"
      srctype="float" size="60" seqIndex="1" defaultValue="no match"/>
  </uniqueKeys>
  <facts>
    <fact name="AMCONTRACT_F">
      <factKeys>
        <factKey name="Z_RDSAMCONTRACT_DID"/>
        <factKey name="Z_RDSAMCOMPANY_DID" fieldName="LCPYID_KEY"
          tableName="AMCOMPANY_D" matchFieldName="LCPYID_KEY"/>
        <factKey name="Z_RDSAMCOSTCENTER_DID"
          fieldName="LCOSTID_KEY" tableName="AMCOSTCENTER_D"
          matchFieldName="LCOSTID_KEY"/>
        <factKey name="START_TIMEID" fieldName="DSTART"
          tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
        <factKey name="END_TIMEID" fieldName="DEND"
          tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
      </factKeys>
      <factMeasures>
        <factMeasure name="REF_FACTOR1" srcType="src"/>
        <factMeasure name="REF_FACTOR2" srcType="src"/>
        <factMeasure name="TSDURATION" srcType="src"/>
        .
        .
      </factMeasures>
    </fact>
  </facts>
  <scdKeys>
    <scdKey rdsFieldName="STATUS" srcFieldName="STATUS"
      srctype="char"/>
  </scdKeys>
  <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 AMCONTRACT.

```

<facts>
  <fact name="AMCONTRACT_F">
    <factKeys>
      <factKey name="Z_RDSAMCONTRACT_DID"/>
      <factKey name="Z_RDSAMCOMPANY_DID" fieldName="LCPYID_KEY"
        tableName="AMCOMPANY_D" matchFieldName="LCPYID_KEY"/>
      <factKey name="Z_RDSAMCOSTCENTER_DID"
        fieldName="LCOSTID_KEY" tableName="AMCOSTCENTER_D"
        matchFieldName="LCOSTID_KEY"/>
      <factKey name="START_TIMEID" fieldName="DSTART"
        tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
      <factKey name="END_TIMEID" fieldName="DEND"
        tableName="RDS_TIMEDIM_D" matchFieldName="FULLDATE"/>
    </factKeys>
    <factMeasures>
      <factMeasure name="REF_FACTOR1" srcType="src"/>
      <factMeasure name="REF_FACTOR2" srcType="src"/>
      <factMeasure name="TSDURATION" srcType="src"/>
      <factMeasure name="TSLESSORNOTICE" srcType="src"/>
      <factMeasure name="TSNOTICE" srcType="src"/>
      <factMeasure name="TSPURCHNOTICE" srcType="src"/>
      <factMeasure name="TSRENNOTICE" srcType="src"/>
      <factMeasure name="TSDEFRENDUR" srcType="src"/>
      <factMeasure name="TSRETNOTICE" srcType="src"/>
      <factMeasure name="MINTPAYAST" srcType="src"/>
      <factMeasure name="MPOCOMMITMENT" srcType="src"/>
      <factMeasure name="MAMOUNT" srcType="src"/>
      <factMeasure name="PDEFRLF" srcType="src"/>
      <factMeasure name="MINTPAY" srcType="src"/>
      <factMeasure name="MINTPAYTAX" srcType="src"/>
      <factMeasure name="PINTRENTPERCENT" srcType="src"/>
      <factMeasure name="PDEFRENPERCENT" srcType="src"/>
      <factMeasure name="MASTINTPAYTAX" srcType="src"/>
      <factMeasure name="MMARKETVAL" srcType="src"/>
    </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 AssetCenter *amModel* table.

```
<associates>
  <associate name="AMMODEL_BRAND_ASS" rdsVersion="5.2"
    tableName="AMMODEL" fieldName="BARCODE_MODEL"
    tableName2="AMBIBRAND" fieldName2="BARCODE"/>
</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">
    <!-- 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_et1.xsd`.

The following is an example of a direct mapping table for the AssetCenter *amProject* table.

```
<directMapping name="AMPROJECT" rdsVersion="5.2"
dataSourceTableName="amProject">
    <directMappingFields>
        <directMappingField name="CODE" type="char" size="20"/>
        <directMappingField name="DEND" type="date"/>
        <directMappingField name="DSTART" type="date"/>
        <directMappingField name="DTLASTMODIF" type="date"/>
        <directMappingField name="LBUDGCNTRID" type="float"/>
        <directMappingField name="LPROJID" type="float"/>
        <directMappingField name="LSUPERVID" type="float"/>
        <directMappingField name="REF" type="char" size="16"/>
        <directMappingField name="STATUS" type="char" size="42"/>
        <directMappingField name="TITLE" type="char" size="40"/>
    </directMappingFields>
    <directMappingIndexes>
        <directMappingIndex name="UNIQUAMPROJECT_IDX">
            <DirectMappingIndexKey fieldName="LPROJID"
srcFieldName="lProjID" srctype="float" seqIndex="1"/>
        </directMappingIndex>
    </directMappingIndexes>
```

## 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 AssetCenter `amContract` table in the RDS database.

```
<scdKeys>
  <scdKey rdsFieldName="STATUS" srcFieldName="STATUS"
    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 AssetCenter `amAsset` table.

```
<aggregateKeys>
  <aggregateKey name="DACQUISITION" type="date"/>
</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 AssetCenter, attributes are updated as:
  - Z\_RDSACTIVESTATUSIND = 'N'.
  - Z\_RDSDELETEDDATE is set to the timestamp value at the time the record was deleted in AssetCenter.
  - 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 RDSac\_et1.xml file take effect.

Usage	RDSInitRun RDS-Schema-XML-File	Modification-Indicator
Sample	RDSInitRun (the default xml file is set as rdsac_et1.xml)	
Sample	RDSInitRun rdsac_et1.xml	
Sample	RDSInitRun testdirect.xml	m
Sample	RDSInitRun testdim.xml	m
Sample	RDSInitRun rdsac_et1.xml	time
Sample	RDSInitRun rdsac_et1.xml	m_upgrade

In the sample commands, m means modified.

## Synchronizing AssetCenter record deletions with RDS

To synchronize records deleted from the AssetCenter database with those in the RDS, BI Portal uses the Record Deletion Synchronization mechanism to improve the general performance for marking records for deletion in RDS.

Your BI Portal installation CD has the necessary import script file that provide all the AssetCenter tables that are known to the RDS AssetCenter schema. These tables have a deletion workflow defined. Each deletion workflow triggers record deletion event logging with a special tag for BI Portal in AssetCenter's output event table.

While using AssetCenter, record deletions are appropriately logged to AssetCenter's event table.

During the RDS scheduled data synchronization point, Connect-It extracts these events from AssetCenter to the RDS AssetCenter BIRECORDDELETE table, and then sends them to the RDS ETL process to handle.

During the last part of the RDS ETL process, the outstanding deletion events in BIRECORDDELETE table are reviewed, and their corresponding records in the RDS AssetCenter database are found and marked with the *inactive* status.

When the ETL process completes, Connect-It removes all the BI Portal deletion-specific event records from AssetCenter's event table.

At the end of the RDS data synchronization cycle, the expected outcome is that all event records of category="BI Deletion" in AssetCenter's amOutputEvent table are deleted from the database. The BIRECORDDELETE table in the RDS AssetCenter database should also be empty.

The deletion workflow sequentially follows these steps.

- 1 AssetCenter updates the record deletion event log.

- 2 Connect-It extracts the event deletion information from AssetCenter and adds it to the RDS BIRECORDDELETE table.
- 3 The RDS ETL data synchronization process reviews the outstanding deletion information stored in RDS, then properly marks the corresponding records as being deleted.





# 2 Using Connect-It to Handle RDS Data Synchronization

## CHAPTER

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 35
- Configuring the connectors on page 38

---

## 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 AssetCenter 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_ac.scn` and `rds_acuser.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_ac.scn` and `rds_acuser.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_ac.ini` and `rds_acuser.ini` files before starting the services. These files reside in the `/cit` folder in the RDS installation root directory.
- Start the `rds_ac` 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_ac.log` file contains the result of this activity. The log file resides in the `RDS/Logs` directory.
- Start the `rds_acuser` service to populate the `RDS_ACUSER` table in the RDS database with the operator information from AssetCenter. 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.

## Asset types in the AMASSET\_D table

The following information lists the record types that are synchronized to the `AMASSET_D` table in the RDS database and the type of asset data you can expect to see in the reports.

- Only Portfolio items with one associated asset will be pushed to the RDS database. In AssetCenter, Portfolio items that have a

Model.Nature.seMgtConstraint = Unique asset tag will be pushed. When viewing each portfolio item record, this value is found in the General Tab > Category Section > Management constraint Field = Unique asset tag.

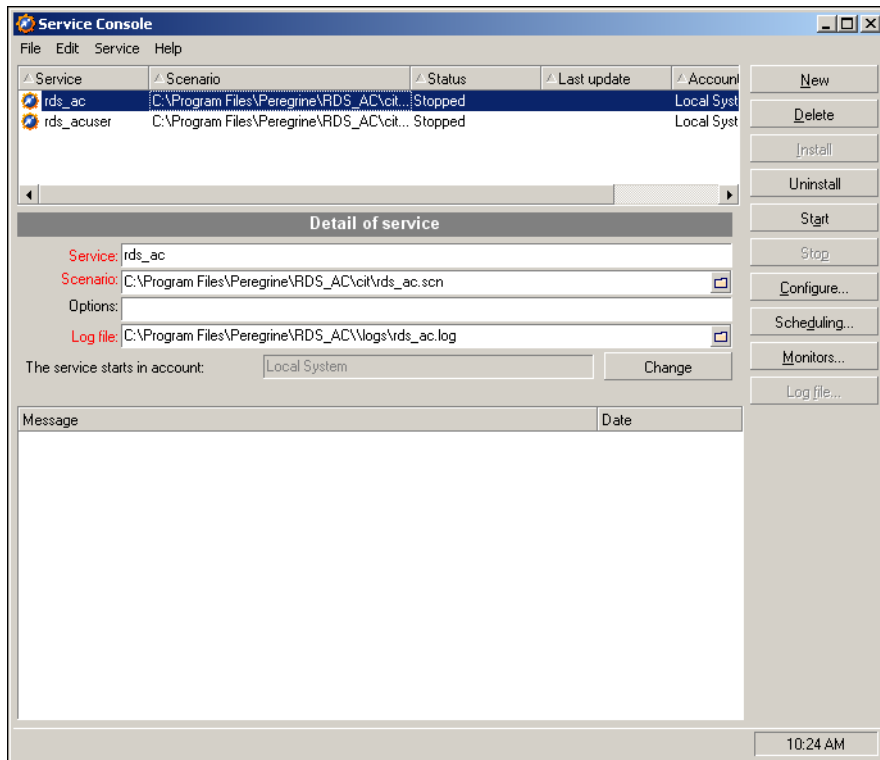
- Computer records from the amComputer table with one associated Portfolio item will be pushed to the AMASSET\_D table.
- Software installation records from the amSoftInsta11 table with one associated Portfolio item will be pushed to the AMASSET\_D table.

---

## Starting and stopping the RDS scenarios

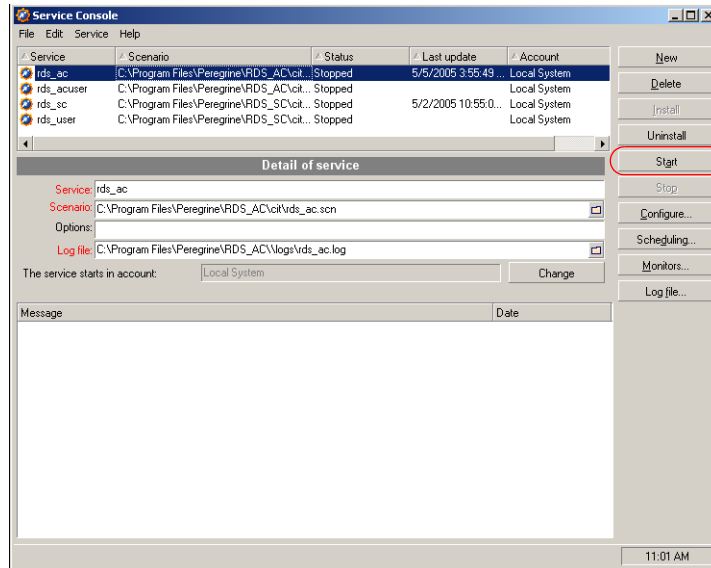
Use the Connect-It console application to start and stop the RDS scenarios. The following figure shows two services, rds\_ac and rds\_acuser.

**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 start the service:

- 1 Select the service name from the service panel.
- 2 Click **Start**.

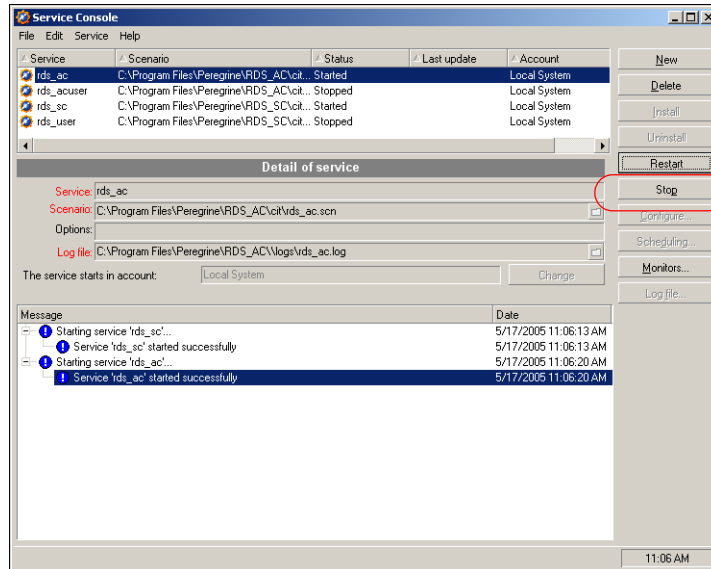


The data synchronization activities are recorded in the log file, `rds_ac.log`. Information is continuously appended to this file while the service is active. Peregrine recommends that you check the log file to monitor the status of your RDS data synchronization.

**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.



## 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.

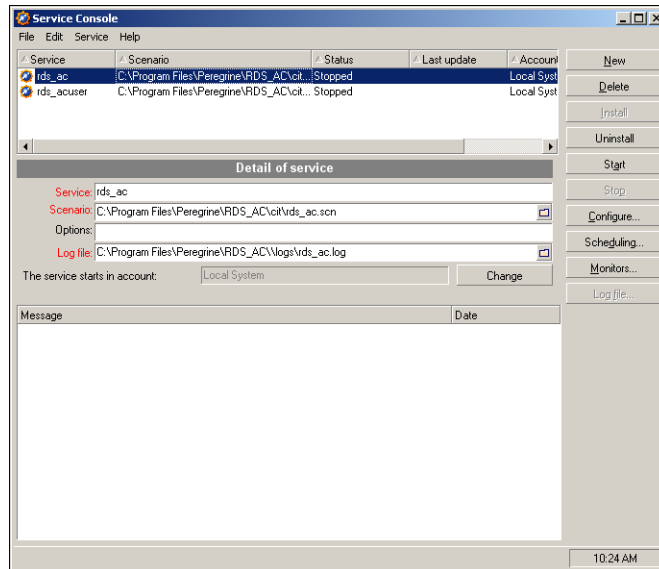
Scenario	Description
rds_ac	Synchronizes data between AssetCenter and the RDS database.
rds_acuser	Synchronizes the operator data between AssetCenter and the RDS_ACUSER table in the RDS database.

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.

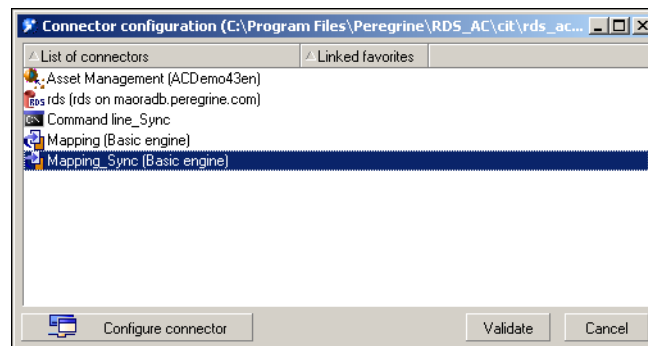
**Warning:** When using RDS multiple data source deployment, run only one RDS user scenario. See the [Administration Guide](#) for more information about multiple data sources.

To manually configure the scenarios:

- 1 Open the Connect-It Service Console.



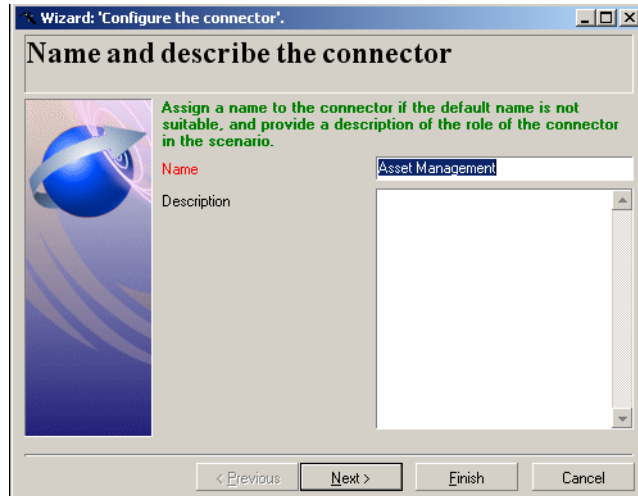
- 2 Highlight the rds\_ac scenario.
- 3 Click **Configure** to open the Connector Configuration window and display a list of connectors.



- 4 Configure the following connectors one at a time.

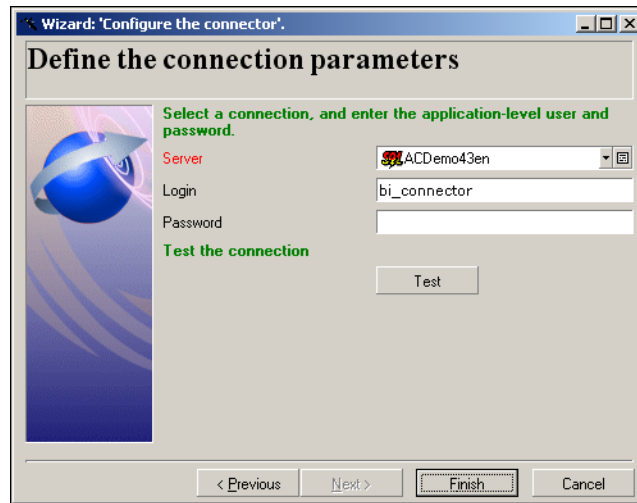
Asset Management  
RDS

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



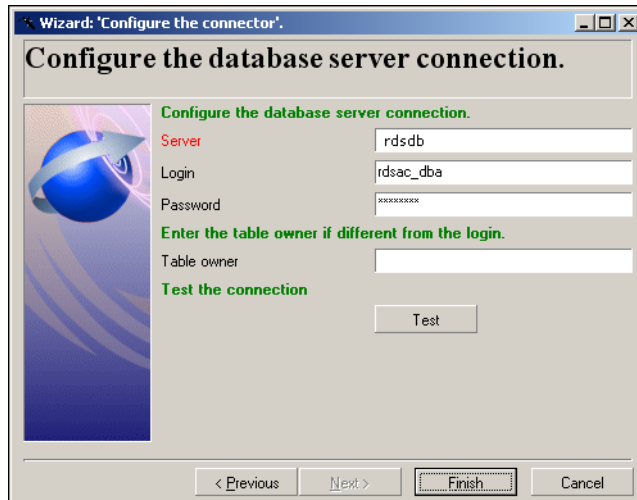


- 6 Using the **Configure the connector** wizard, click Next and enter the following information.



- a Type the AssetCenter server host name.
- b Type the AssetCenter default login, **bi\_connector**.
- c Type the AssetCenter 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.



- 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\_ac scenario, click **Validate** to close the Connector configuration window.
  - 9 Select the rds\_acuser scenario, then click **Configure**.
  - 10 Repeat [Step 4 on page 40](#) through [Step 8](#) in this section for the rds\_acuser scenario.



# 3 Customization Workflow

## CHAPTER

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_acuser` 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 AssetCenter, that data needs a place to reside. The RDS needs to have additional tables and fields created to accept new AssetCenter tables, 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 AssetCenter 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 AssetCenter 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 AssetCenter 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 Deletion Workflows need to be added in AssetCenter?

## Considerations for modifying and deleting existing data structures

**Important:** Deletions are strongly discouraged in the RDS. If a field has been deleted in AssetCenter, 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 Deletion Workflows in AssetCenter 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\_ac scenario and the rds\_acuser 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
rdsac_et1.xml	The RDS schema file
rds_ac.scn	The Connect-It scenario file
rds_ac.unv	The Business Objects Universe file
rdms_keywords.xml	Your database reserved words list
rds_acuser.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: `rdsac_et1.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 AssetCenter as an administrator and edit your Deletion Workflows 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 AssetCenter to the RDS.

Open the `rds_ac.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 AssetCenter. 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 AssetCenter 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_ac.unv` file, located in the directory under the Business Objects installation root directory:

```
\nodes\
```

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_ac` 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 RDS Customization

## CHAPTER

This chapter discusses the following topics:

- RDS schema customization on page 51
- AssetCenter deletion workflow on page 57

---

## RDS schema customization

You can modify the `RDS_acet1.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 AssetCenter (AC) database on which RDS is deployed. This section covers three different customization examples:

- New tables are added to the AssetCenter database
- New fields are added to existing AssetCenter 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_acet1.xml` schema file.

**Step 2** Use an XML editor or a text editor to make the necessary changes to the `RDS_acet1.xmlschema` 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.

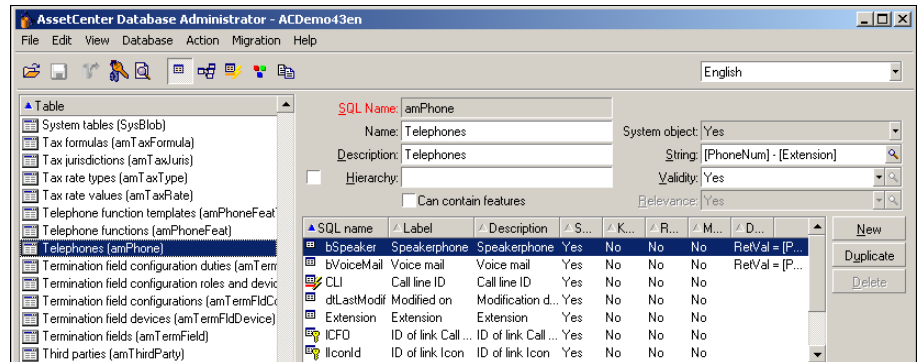
## Adding a new table

When you add a new table in AssetCenter 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, *amPhone*, already is in the AssetCenter database. This table, *amPhone*, has ten fields with `1PhoneId` as the primary index.

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

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



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

### To add the amPhone 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, amPhone is a directMapping table, so you need to add an instance of a directMapping table into the schema file.

```
<directMapping name="AMPHONE" rdsVersion="5.2"
  dataSourceTableName="amPhone">
<directMappingFields>
  <directMappingField name="LPHONEID" type="float"/>
  <directMappingField name="DTLASTMODIF" type="date"/>
  <directMappingField name="PHONENUM" type="char" size="30"/>
  <directMappingField name="EXTENSION" type="char" size="5"/>
  <directMappingField name="C_BVOICEMAIL" type="char"
    size="10"/>
  <directMappingField name="C_BSPEAKER" type="char" size="10"/>
  <directMappingField name="CLI" type="char" size="40"/>
  <directMappingField name="LICONID" type="float"/>
  <directMappingField name="LCFO" type="float"/>
  <directMappingField name="LITEMID" type="float"/>
</directMappingFields>
<directMappingIndexes>
  <directMappingIndex name="UNIQAAMPHONE_IDX">
    <DirectMappingIndexKey fieldName="LPHONEID" srcFieldName=
      "lPhoneId" srctype="float" seqIndex="1"/>
  </directMappingIndex>
</directMappingIndexes>
</directMapping>
```

- b** Using a text editor, open the RDS schema file, RDS\_acet1.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\_AC.
    - b** Type the following commands:

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

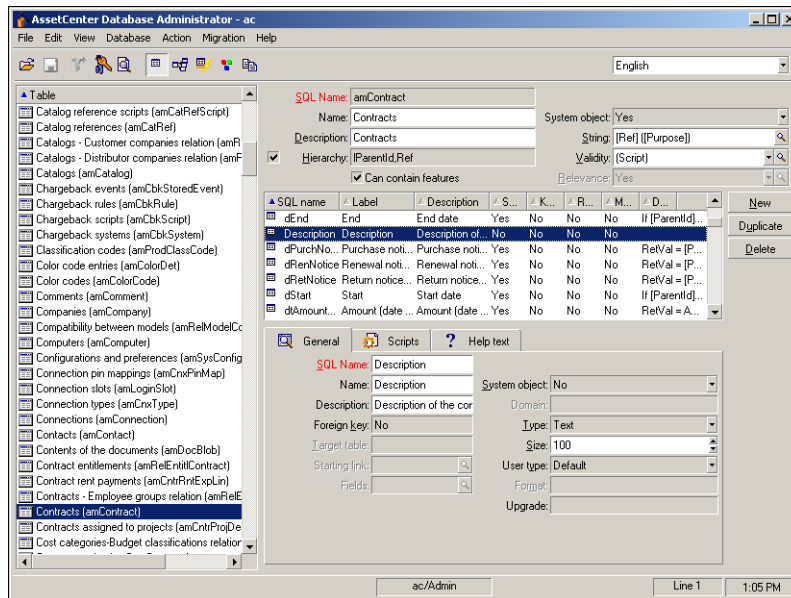
When the RDS initialization batch tool finishes initializing the RDS database, the new table is created.

- 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, `Description`, added to an existing `AssetCenter` table, `amContract`.



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 `AssetCenter` `amContract` table is `Description`. Its counterpart RDS field name is `DESCRIPTION`.

## To add a new field:

- 1 Add the field definition for Description as DESCRIPTION in the RDS\_acet1.xml schema file.

```
<dimension name="AMCONTRACT" rdsVersion="5.2">
<dimensionTableName>
AMCONTRACT_D
</dimensionTableName>
<dataSourceTableName>
amContract
</dataSourceTableName>
<dimensionTableFields>
<dimensionTableField name="CONTRACTNO" type="char" size="20"/>
...(many dimensionTableField definitions are omitted)
<dimensionTableField name="C_MPOCOMMITMENT" type="char" size="40"/>
<dimensionTableField name="REF_FACTOR1" type="float"/>
<dimensionTableField name="REF_FACTOR2" type="float"/>
<dimensionTableField name="DESCRIPTION" type="char" size="100"/>
</dimensionTableFields>
<uniqueKeys name="contact_unique">
...
</uniqueKeys>
<facts/>
<scdKeys/>
<aggregateKeys/>
</dimension>
```

- 2 Run the RDS initialization batch tool to add Description to the existing CONTRACT\_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_AC.
```

- 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 CONTRACT\_D table.

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



## AssetCenter deletion workflow

Each table in the AssetCenter database that RDS references requires a defined deletion workflow. While you can interactively define the deletion workflows in the AssetCenter user interface, importing the definitions into the AssetCenter database facilitates RDS deployment. The import script contains the following collection of script files.

```

bi.lst and bi.str (bi.lst is the lead script file for the import operation)
biworkflow.scr and biworkflow.scx
stdactiv.txt
stdevent.txt
stdrelac.txt
stdstart.txt
stdtrans.txt
stdwactn.txt
stdwf.txt
stdchoic.txt, stdrole.txt, stdalrm.txt
A collection of VB script files in subdirectory, script_memComment, one for each
dimension table

```

Refer to [Importing AssetCenter script files](#) in the [Installation Guide](#) for more information.

Because the contents of these script file are cross-referenced, it is important that that you do not manually edit these files.

You need to define deletion workflow for both dimension and direct mapping tables. Define the deletion workflow for the following 25 tables.

AMWFSCHEME key #	AMWFACTIVITY key #	AC table name	Unique Key field in AC
BI001	BI001001	amPortfolio	PortfolioItemId
BI002	BI001002	amPeriod	PeriodId
BI003	BI001003	amModel	ModelId
BI004	BI001004	amBrand	BrandId
BI005	BI001005	amLocation	LocaId
BI006	BI001006	amCompany	CpyId
BI007	BI001007	amEmpDept	EmpDeptId

AMWFSCHEME key #	AMWFACTIVITY key #	AC table name	Unique Key field in AC
BI008	BI001008	amContract	lCntrId
BI009	BI001009	amBudgCenter	lBudgCntrId
BI010	BI001010	amCostCenter	lCostId
BI011	BI001011	amBudgLine	lBudgLineId
BI012	BI001012	amExpenseLine	lExpLineId
BI013	BI001013	amCntrRent	lCntrRentId
BI014	BI001014	amCurrency	lCurId
BI015	BI001015	amCurRate	lCurRateId
BI016	BI001016	amAstProjDesc	lAstProjDescId
BI017	BI001017	amProjcet	lProjId
BI018	BI001018	amStock	lStockId
BI019	BI001019	amWorkOrder	lWorkOrderId
BI020	BI001020	amItemListVal	lItemListValId
BI021	BI001021	amNature	lNatureId
BI022	BI001022	amCostCategory	lCostCatID
BI023	BI001023	amCountry	lCountryId
BI024	BI001024	amBudgetCategory	lBudgCatId
BI025	BI001025	amBudget	lBudgId

## AssetCenter VB script action definition

This script file defines how the field memMessage of an amOutputEvent record is constructed for a deleted record of a specific table. The following example defines the amOutputEvent record that corresponds to a deleted amPortfolio record.

```
Dim lErr as long
Dim hrOE as long

hrOE = AmCreateRecord("amOutputEvent")
lErr = AmSetFieldStrValue(hrOE, "Category", "BI Deletion" )
lErr = AmSetFieldStrValue(hrOE, "memMessage", "<BIDeletion
table=amPortfolio ID=" & [LPortfolioItemId] & "/>" )
lErr = AmInsertRecord(hrOE)
' release handle
lErr = AmReleaseHandle(hrOE)
```

In the predefined import scripts, each table has one action definition:

- The AssetCenter table name.
- The unique column name used inside the RDS AssetCenter database for the specified AssetCenter table.

In this example, the action definition is to track record deletion of the AMASSET\_D table in the RDS AssetCenter database. The AMASSET\_D table mainly maps to the amPortfolio table in the AssetCenter database. The key to finding the correct record in the AMASSET\_D table for deletion handling is the lPortfolioItemId field.

## Content of an amOutputEvent record

When a record of an AssetCenter table with deletion workflow defined is being deleted, a corresponding record is added to the amOutputEvent table with the following content.

Field Name	Description	Example
CATEGORY	The type of the event	BI Deletion
DTCREATED	The date and time when the event is created	03/07/2005 15:17:15
MEMMESSAGE	Detail of the deletion event	<BIDeletion table=amPortfolio ID=20105/> means that the amPortfolio record with lPortfolioItemId field =20105 is being deleted.

**Note:** The rds\_ac.scn scenario creates BI\_Connector as the user who accesses the AssetCenter database. The user rights for BI\_Connector allow for deletion of amOutputEvent records. If you want to use a different user other than BI\_Connector, make sure that the designated user has BI\_Admin rights assigned. This is necessary to ensure that processed event records are properly deleted from the AssetCenter amOutputEvent table.

## Importing predefined deletion workflow into AssetCenter

The script file that you need to import for the workflow is on the RDS for AssetCenter installation CD. The file path is:

RDS-AC\SupportFiles\AssetCenter\import\bi.1st. The deletion workflow definition and the BI-specific user profile and rights definition are in the lead

script file, `bi.lst`. Refer to [Importing AssetCenter script files](#) in the [Installation Guide](#) for more information.

After you import the deletion workflow definitions, record deletions of the specified AssetCenter tables are logged in the `amOutputEvent` table in the AssetCenter database for the RDS ETL process to consume.

## Extracting record deletion information

You extract record deletion information from the AssetCenter database to the RDS AssetCenter database. During the RDS ETL update cycle, Connect-It processes the record deletion events logged in `amOutputEvent`.

- Connect-It filters the records in the `amOutputEvent` table of the **BI Deletion** category.
- For each selected `amOutputEvent` record, Connect-It extracts the needed information from the source event record to create a corresponding `BIRECORDDELETE` record in the RDS with the following content mapping.

AssetCenter <code>amOutputEvent</code> Field	RDS <code>BIRECORDDELETE</code> Field
<code>DTCREATED</code>	<code>TIMESTAMP</code>
<code>Table=&lt;&gt; in memMessage</code>	<code>TABLENAME</code>
<code>Id=&lt;nnn&gt; in memMessage</code>	<code>KEYVAL1</code>

## Marking deleted records in the RDS

Near the end of the RDS ETL process, the content of the `BIRECORDDELETE` table is inspected. If the table is not empty, then all the records in the table are processed. RDS does the following tasks for each record found in the `BIRECORDDELETE` table.

- Finds the record, as keyed by the value of `KEYVAL1`, in the RDS table corresponding to the specified AssetCenter table; for example, the value of `TABLENAME`.
- Marks the record as inactive by updating the `Z_RDSACTIVESTATUSIND` field from "Y" to "N".
- Updates the deletion timestamp of the record; for example, sets the field `Z_RDSDELETEDDATE` to the value of the `TIMESTAMP` field of the corresponding `BIRECORDDELETE` record.

These records are removed from the table when all the records in the BIRECORDDELETE table are processed. At the end of the RDS ETL process, Connect-it tracks the data synchronization status. The existing BIDeletion-specific records found in amOutputEvent table are then removed.





# 5 Connect-It Scenario Customization

## CHAPTER

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_ac.scn`, for AssetCenter. The RDS uses this scenario to perform data synchronization. This chapter also covers how to add a new data 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 64
- Adding new mapping on page 66
- Scheduling automatic data synchronization on page 80

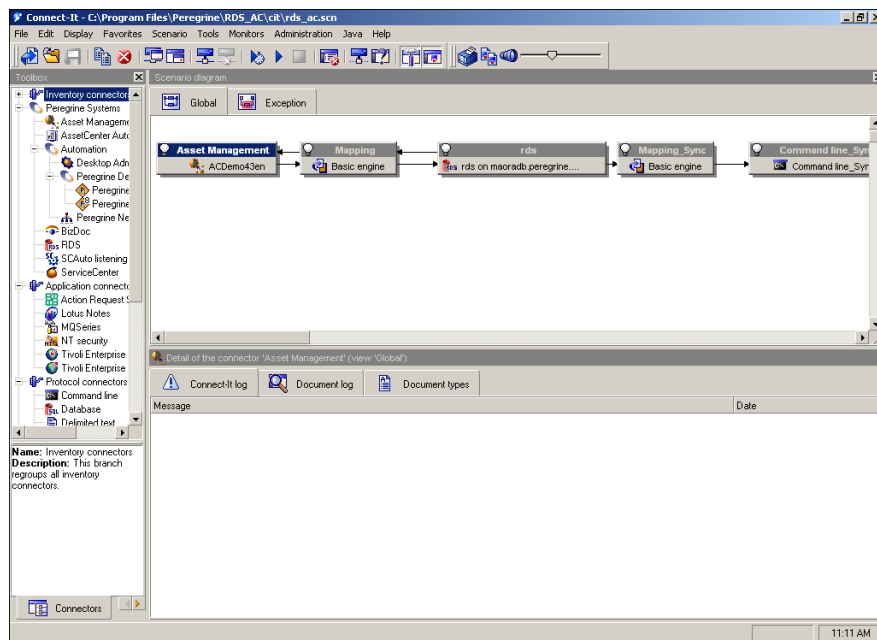
## Using CIT Scenario Builder tools to view an RDS scenario

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, select **Open** and navigate to `rds_ac.scn` (for example, `C:\Program Files\Peregrine\RDS_AC\cit\rds_ac.scn`) to open the Scenario diagram window.

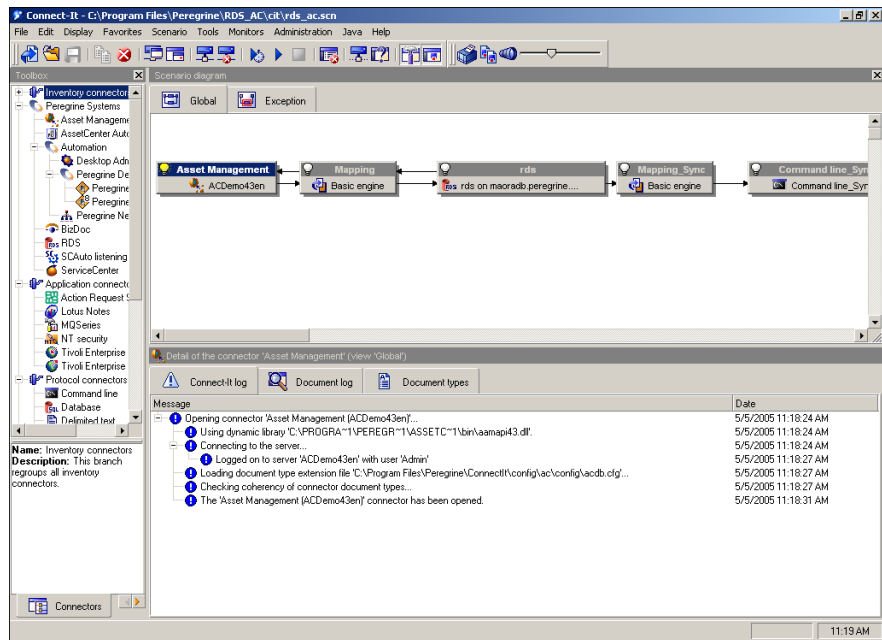




- 2 Highlight the Asset Management connector, then right-click and select **Open connector** in the pop-up menu.

This instructs Connect-It to open the data source, AssetCenter, 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 AssetCenter (the source) against those defined in the scenario. If it finds any inconsistency between the scenario and AssetCenter, the message is flagged with a red stop sign.



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

- Add or remove document types.
- Update scheduler settings.

## Adding new mapping

You must add a new data mapping to the existing scenario (`rds_ac.scn`) if you customized your AssetCenter 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.

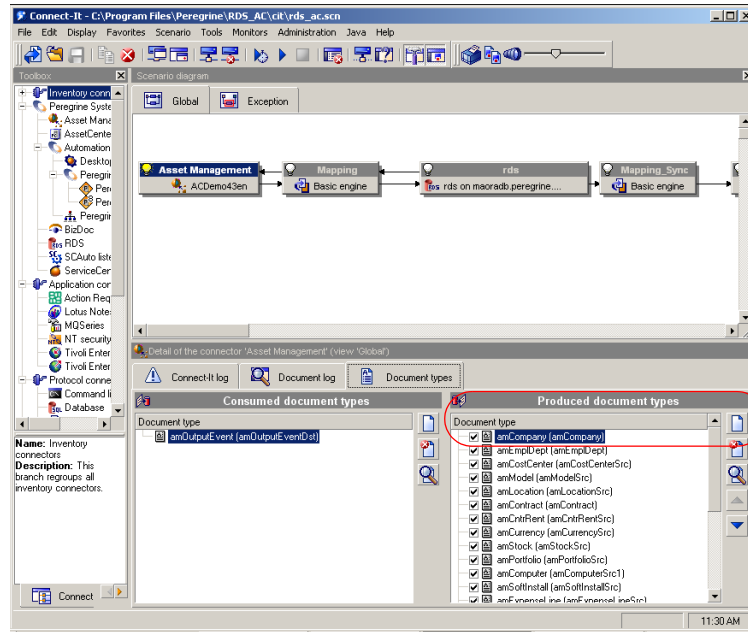
### Adding mapping for a new table

As the example in [Chapter 4, RDS Customization](#) shows, when you add a new table, `amPhone`, in AssetCenter, you need additional customization to make the data available for reports. The `amPhone` table is created in the RDS database at RDS initialization time. The table has ten fields, with `1PhoneId` the key to the table. This section presents the steps to customize the RDS scenario file, `rds_ac.scn`, to include the mapping information for the new table so that at RDS data synchronization time, the data in the AssetCenter `amPhone` table is populated to the RDS `amPhone` table.

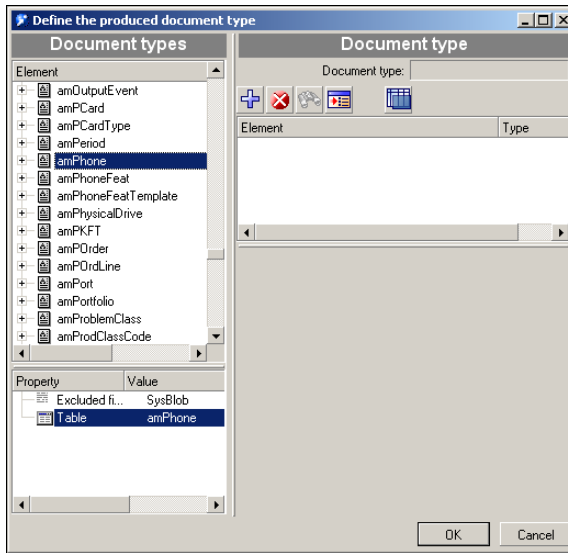
#### To add mapping for a new table:

- 1 From your Connect-It Scenario Builder tool, open `rds_ac.scn`.
- 2 Find and select the connector node with the **Asset Management** label, then right-click and select **Open connector** in the pop-up menu.
- 3 Select the Document types tab.

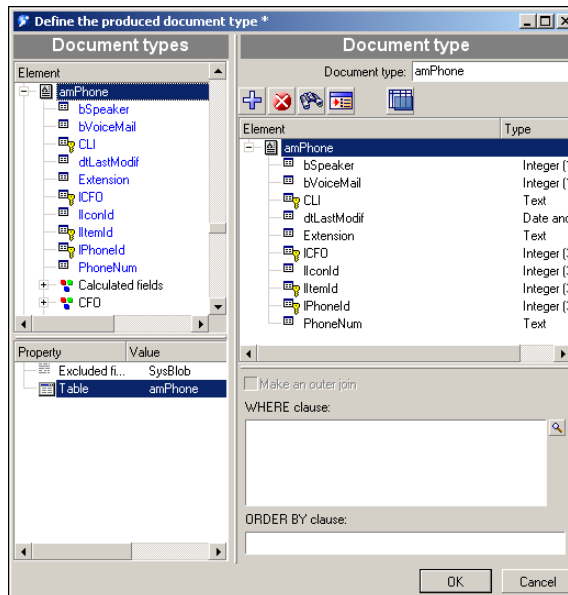
#### 4 Click **Create** under the Produced document types pane.



- In the **Define the produced document type** window, select the new table, **amPhone**, from the Document types list box.

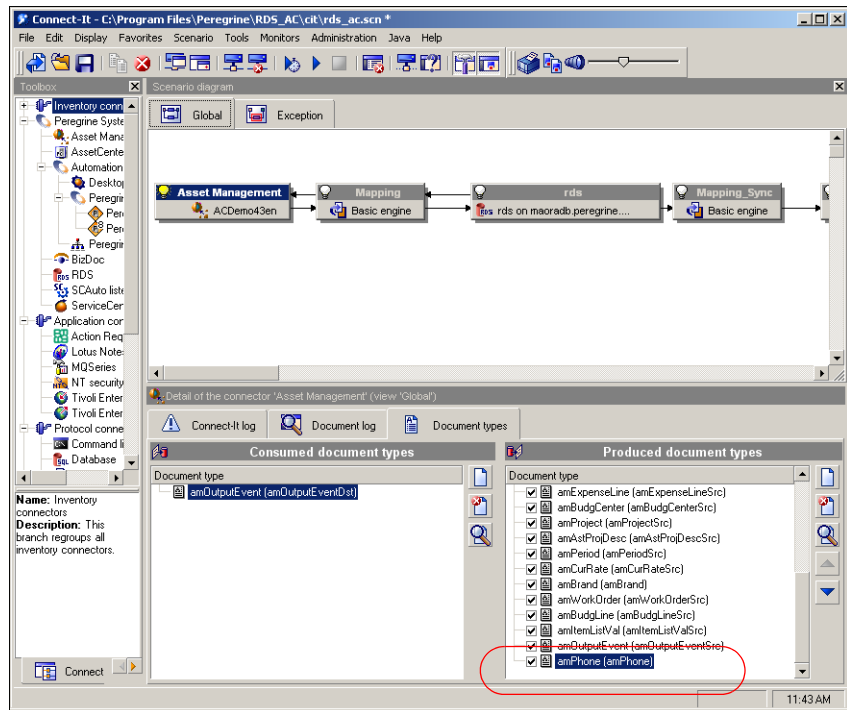



- Click **Add** to add a new table mapping to the Document types frame. 

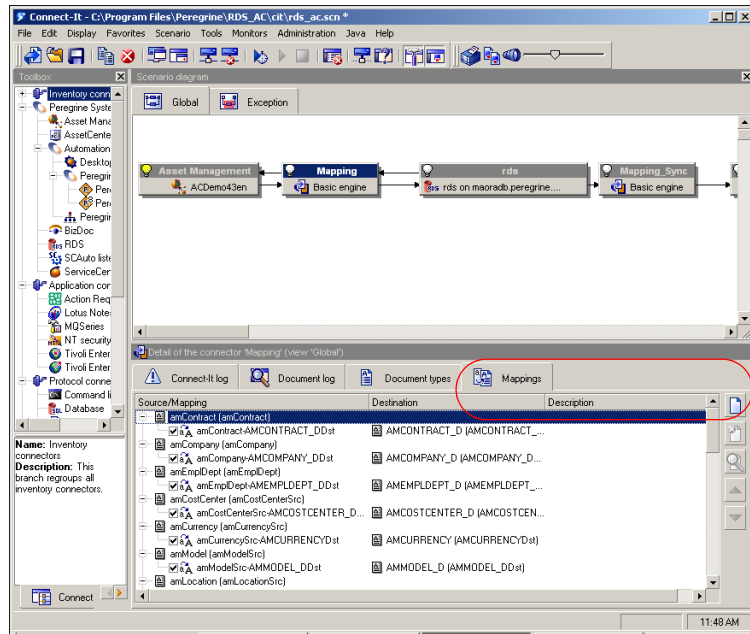


- Click **OK** to add the new table as a new document type and dismiss this window.

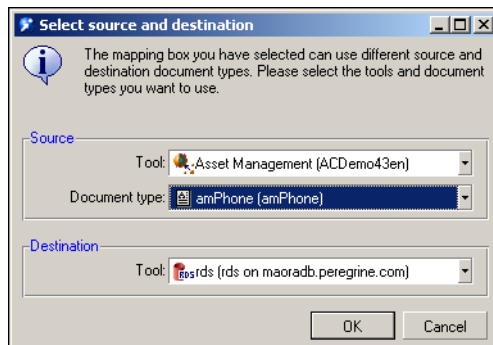
The amPhone table appears as a new document type in the Produced document types pane.



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

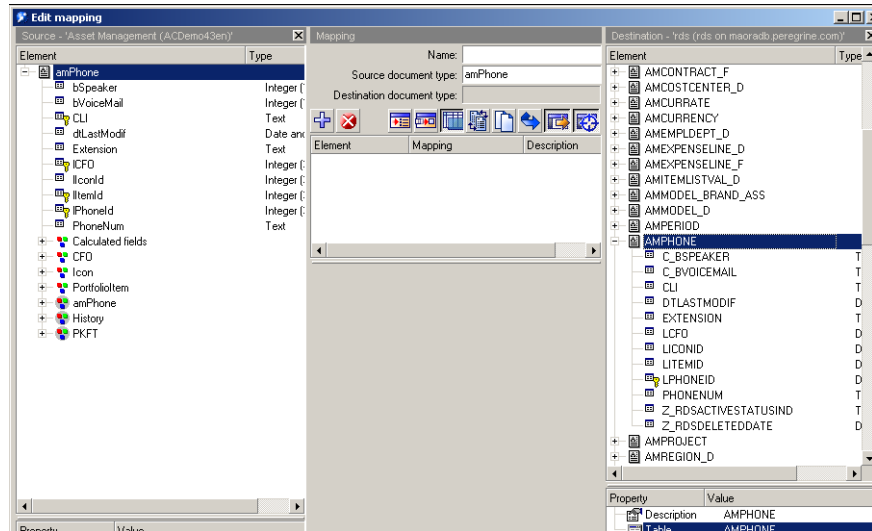



- 9 Select **amPhone** from the Document type drop-down list.



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

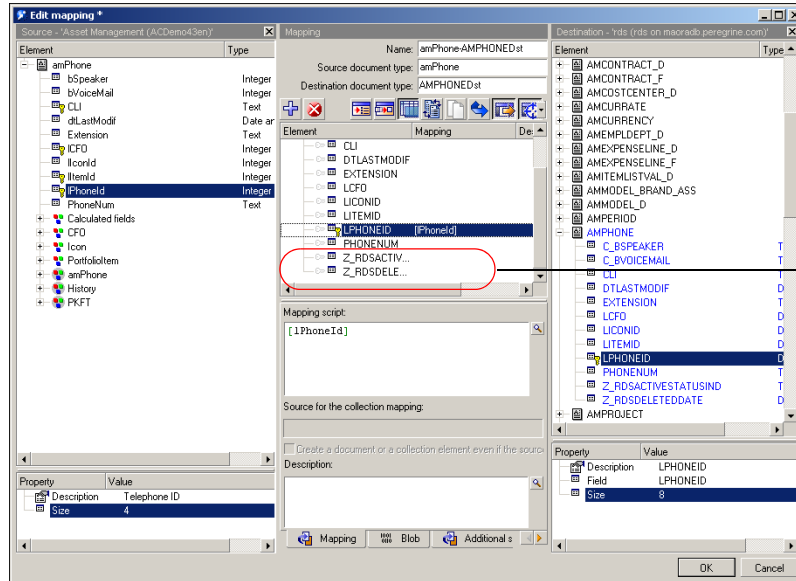
After Connect-It finishes processing the new mapping definition, a new mapping window opens for amPhone.



- 11 To edit the mapping for amPhone, type **amPhone** in the Name field.
- 12 Select **AMPHONE** 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, 1PhoneID of amPhone maps to LPHONEID of AMPHONE, so selecting 1PhoneId and dragging the selection to LPHONEID causes the mapping [1PhoneId] to be created visually.



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

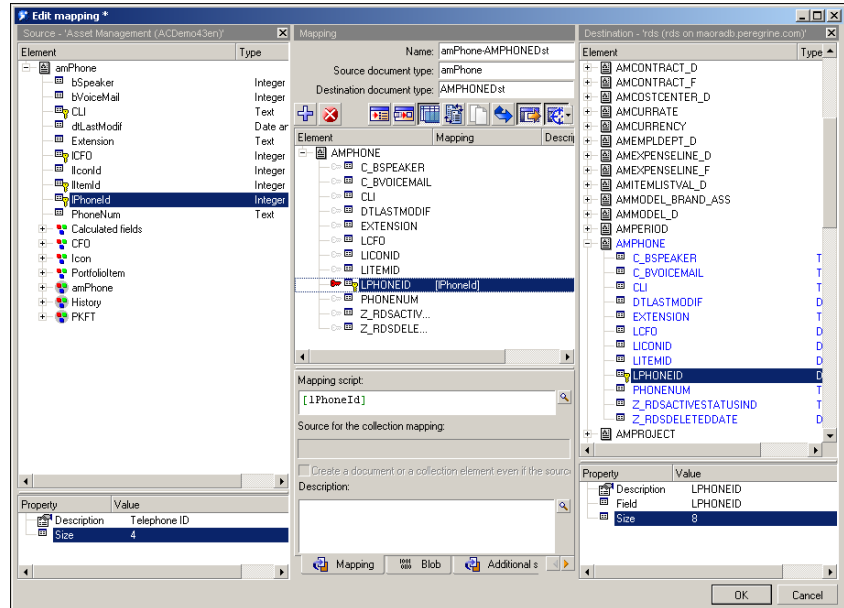
When you finish adding all field mappings, you need to define the primary key field.



## 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 LPHONEID field.

In this example, the primary key for AMPHONE is 1PhoneID. It turns red.



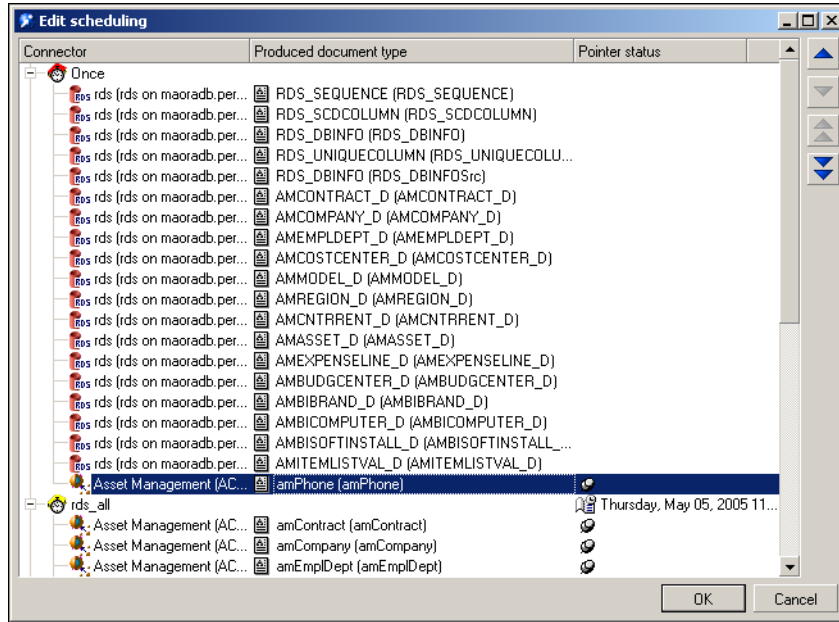
You defined the primary key.

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

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

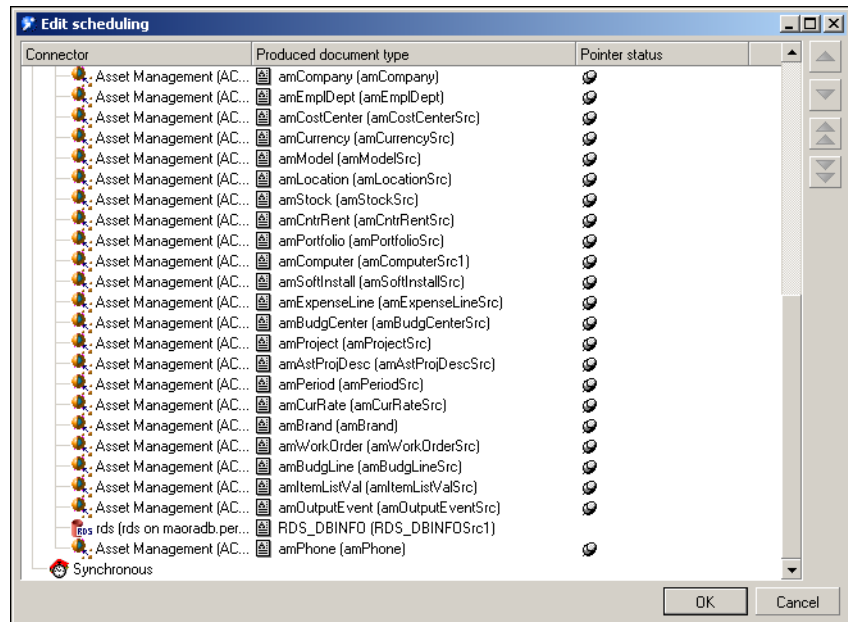
## To add the new table to the scheduling process:


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

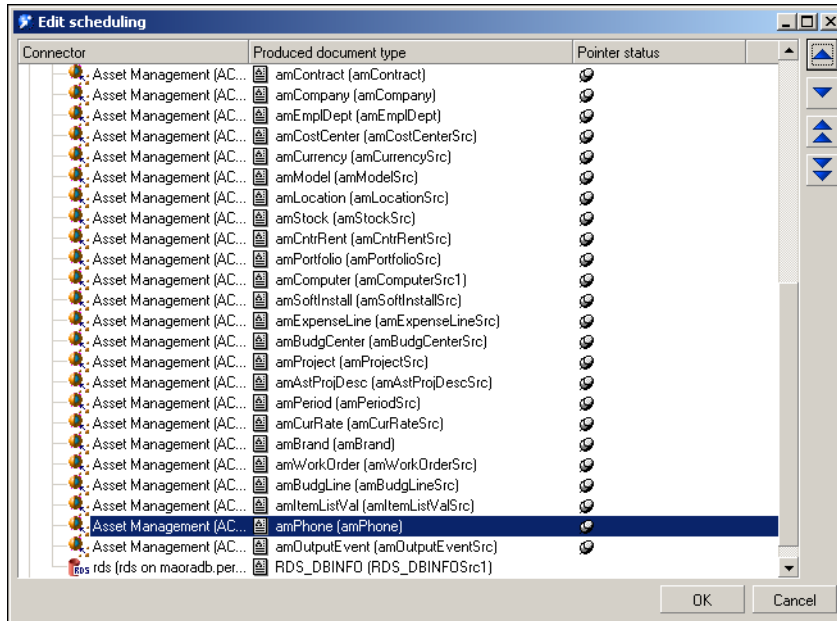


- 2 Scroll down the list of table names for connector Once.
- 3 Select amPhone in the Produced document type column.

- 4 Drag-and-drop the amPhone entry from connector Once to connector rds\_a11.



- 5 Highlight the new mapping, and using the Move Up icon, position the listing before `amOutputEvent (amOutputEventSrc)`. 



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


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

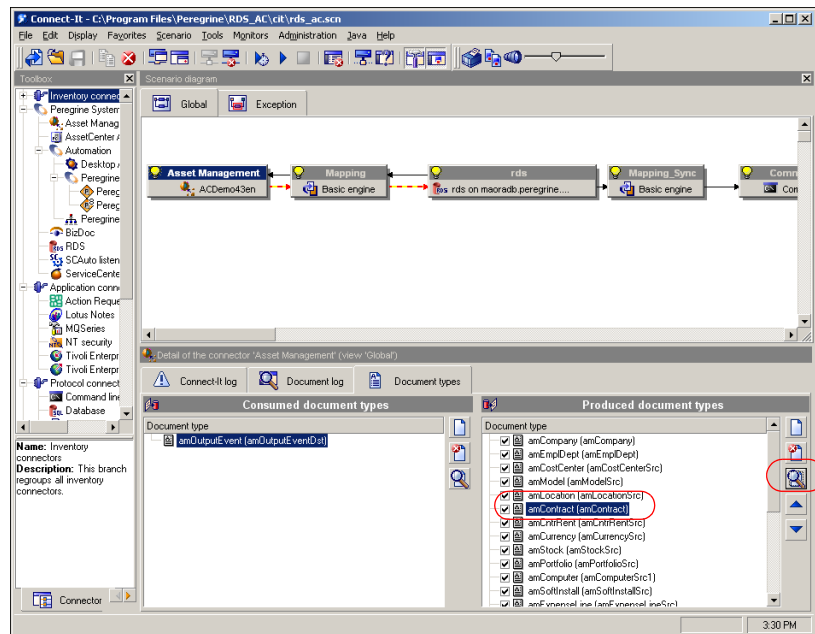
You are ready to use the scenario to perform data synchronization.


## Adding a new field to an existing mapping

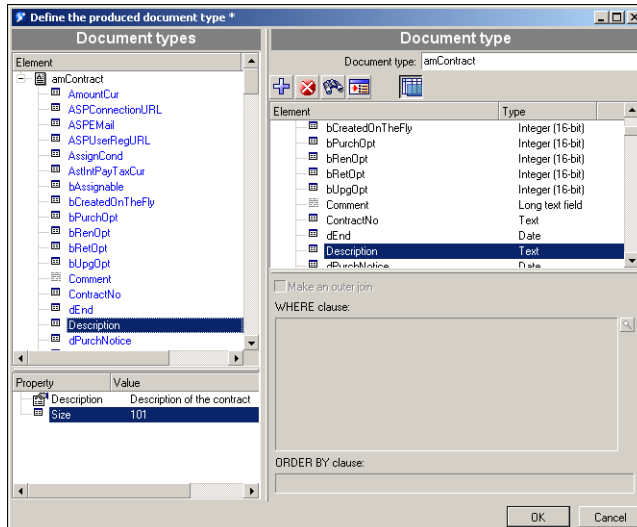
In [Chapter 4, RDS Customization](#), you added a new field, `Description`, to the `amContract` table in Assetcenter and added the `DESCRIPTION` field to the `AMCONTRACT_D` table in the RDS. For the Connect-It scenario to populate this new field in the RDS with data, you must modify the RDS scenario to include mapping information for the new field. The following example shows how to add the new field, `Description`, to an existing table mapping in `rds_ac.scn`.

## To add a new field to rds\_ac.scn:

- 1 Open `rds_ac.scn` in your Connect-It Scenario Builder tool.
- 2 Find and select the **Asset Management** connector node, then right-click to display **Open connector** from the pop-up menu.
- 3 Add the new Description field to the **amContract** document type.
  - a Click the **Document types** tab.
  - b Select the `amContract (amContract)` element in the Produced document type pane.
  - c Click **Edit** to edit the detail fields of the contacts table. 

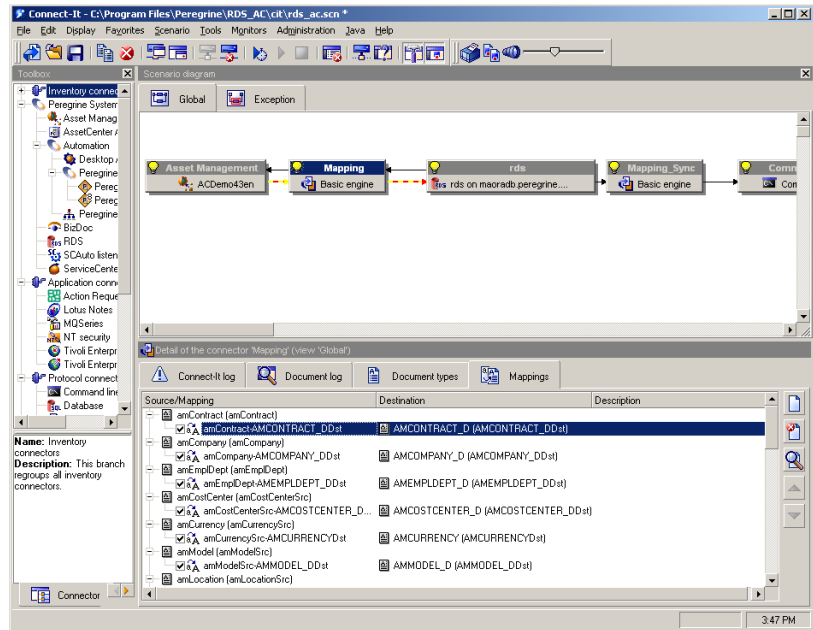


- 4 Edit the document type, amContract, and add the new field Description.
  - a Select the Description field in the Element list box for the table in the left pane of the window and click **Add** to add the new field to amContract. 
  - b Click **OK** to dismiss the edit document type window.




- 5 Add a new field mapping for Description by selecting the **Mapping** connector.
  - a Click the **Mappings** tab.
  - b Select the mapping relationship of amContract to AMCONTRACT\_D.

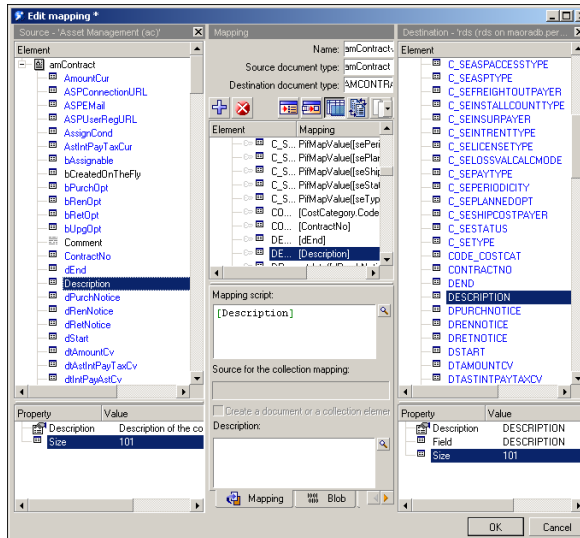
- c Click **Edit** to open the Edit mapping window for this mapping. 



- d Select the DESCRIPTION field from the Destination element list box.

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

- f Select the **Description** field from source element list box, then drag-and-drop it on the **DESCRIPTION** element in the Mapping element list box.
- g Click **OK** to finish the mapping definition.



## 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.

## Scheduling automatic data synchronization

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

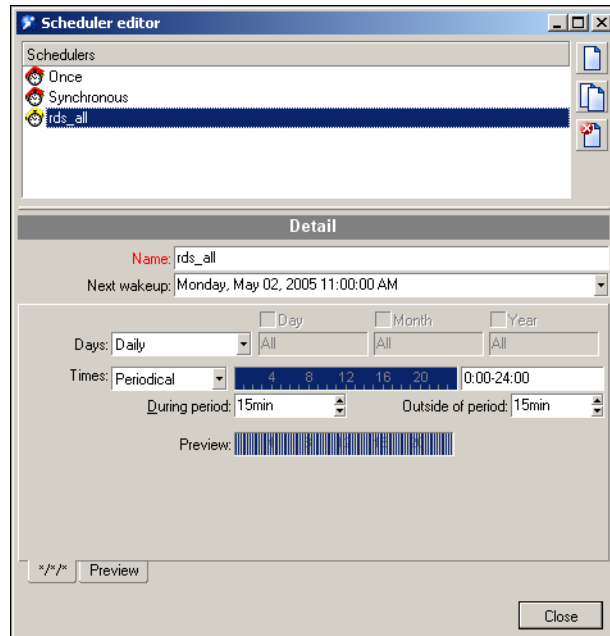
Because some of the synchronization tasks require more system resources than others, Peregrine recommends that you do not reconfigure them to occur more frequently than the default time intervals.

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_ac` scenario.
- 3 Click **Scheduling** to open the Connect-It Scheduling window.
- 4 Click **Edit Schedulers** to open the Connect-It Scheduler Editor.



The following defines a synchronization schedule.

Schedule	Description
<code>rds_all</code>	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_acuser` interval using these same steps. If you change the `rds_acuser` interval in the Connect-It scenario, you must also change the BI

Portal Administrative setting for the **User Synchronization Interval** to match the `rds_acuser` interval. You change the **User Synchronization Interval** using BI Administration in the Administration module of the portal.

The screenshot shows the 'BI Portal Settings' window with a left-hand navigation menu and a main settings area. The 'Admin' section is expanded, showing various sub-options. The main area contains a list of settings with input fields and a 'Restore' button. The 'User Synchronization Interval' is highlighted with a red circle and has a value of 900. Below the settings are 'Back', 'Save', 'Restore', and 'Next' buttons.

Setting Name	Value
Data Security Refresh Interval	3600
RDS Log Table Purge Interval	3600
<b>User Synchronization Interval</b>	<b>900</b>
BO Admin Server Refresh Interval	1800

Refer to the [BI Portal Administration Guide](#) for more information.



# 6 Business Objects Universe Customization

CHAPTER

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 83
- Editing security profiles on page 96

---

## 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_ac.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_ac.unv` file to the Business Objects server.

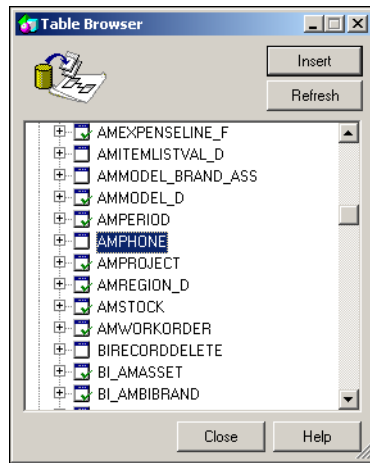
### Adding a new table

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

#### To add a new table:

- 1 Click **Insert > Tables** to open the Table Browser dialog box.

- 2 Select the AMPHONE table and then click **Insert**.

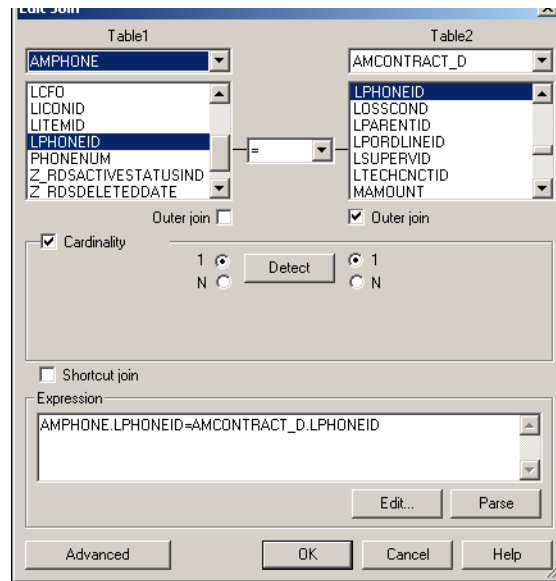


- 3 Click **Close** to close the window.

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.



- 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 AMPHONE table on the field LPHONEID.
- 6 In Table 2, select the RDS table and the field linked to this table. In this example, it is AMCONTRACT\_D, using the field LPHONEID.

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 contract to exist without the LPHONEID, use an outer join on the contract side for this example. Click the appropriate check box.

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

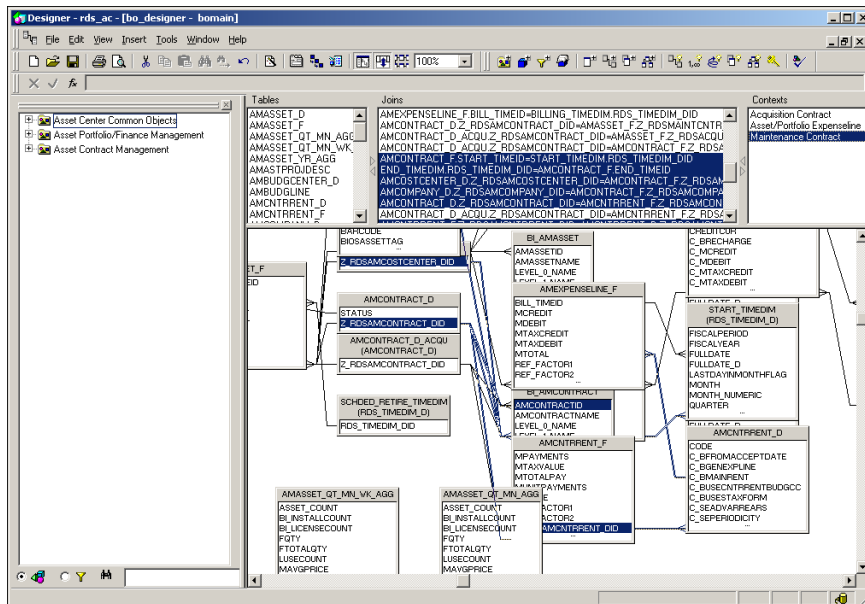
- 7 Click **Detect** to determine the Cardinality of your join, then read the text that appears beneath the button that describes the join. If this is not correct, change the Cardinality as appropriate.
- 8 When you are satisfied with all the details of the link, click OK.

You see a line drawn from AMPHONE to AMCONTRACT\_D. You can right-click on this line and select Join Properties at any time to edit the settings again.

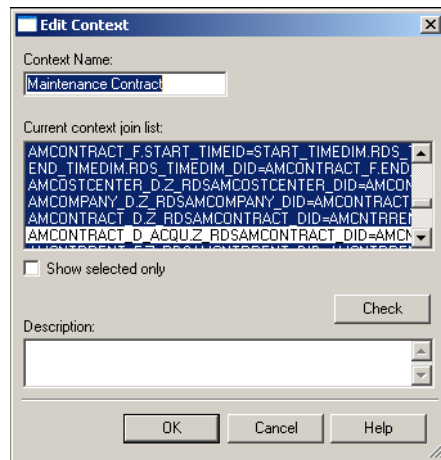
After creating the new link, you must make sure that the new join relationships are added to the existing Context.

### To edit the Context:

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



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



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

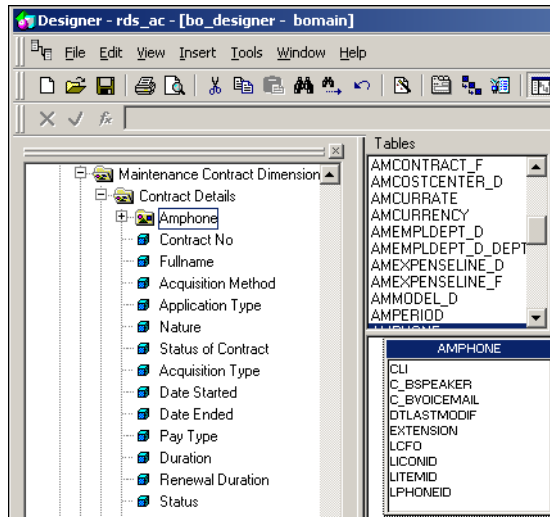
**Tip:** Hold the Ctrl key when adding the link so that you keep the existing links.

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 Amphone folder.

#### To add this table to the object browser:

- Expand the folders to open the **Asset Contract Management > Asset Contract Dimensions > Maintenance Contract Dimensions > Contract Details** folder.

- Click and drag the AMPHONE table from the right pane to the **Contract Details** folder.



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

- 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 Lphoneid as Unique identifier that links a contract to a phone (CUSTOM OBJECT). A quick search on (CUSTOM OBJECT) in the description shows all personalized items.

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

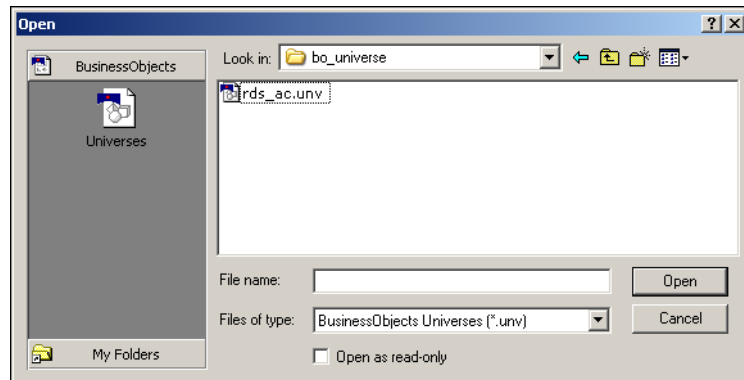


### To edit the rds\_ac.unv file:

- 1 Log on to your Business Objects Designer Tool as a user with Designer rights.
- 2 Make a copy of your universe file (rds\_ac.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\_ac.unv).



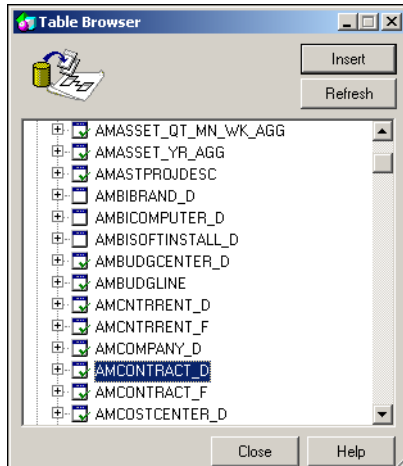
- 4 In the Open dialog box, select the universe file (rds\_ac.unv) and click **Open**.

In this example, you add one new field to the AMCONTRACT\_D table.

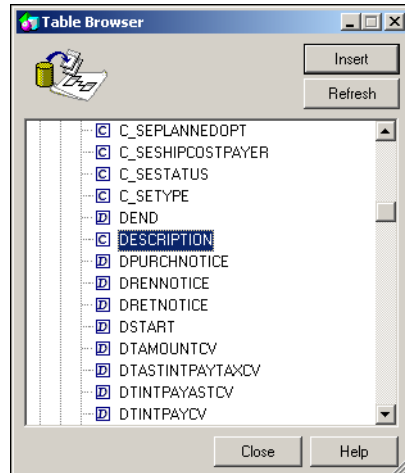
### To add a new field:

- 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 AMCONTRACT\_D to expand the table and see individual fields.

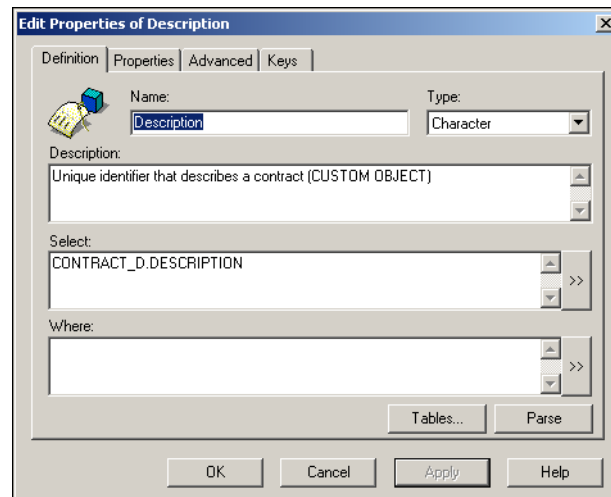


- 3 Click the new field (in this case, DESCRIPTION) and then click **Insert** to add the field to the universe.
- 4 Click **Close** to close the window.

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 **Asset Contract Management > Asset Contract Dimensions > Maintenance Contract Dimensions > Contract Details** folder.
- 2 Right-click the **Contract Details** folder and select **Object**.



- 3 In the Edit Properties window, type a name in the Name text box, for example **Description**.
- 4 In the Type drop-down list box, select a type, for example, **Character**.
- 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 Description as Unique identifier that describes a contract (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, CONTRACT\_D.DESCRPTION.
- 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.

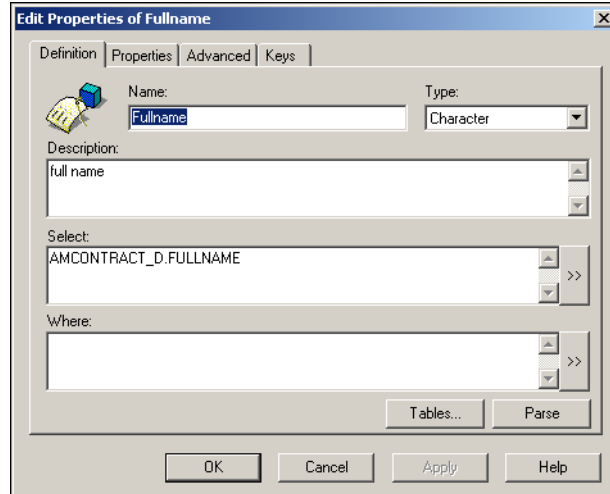
## Modifying an existing field

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

### To modify an existing field:

- 1 Browse to the field in the left pane.

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



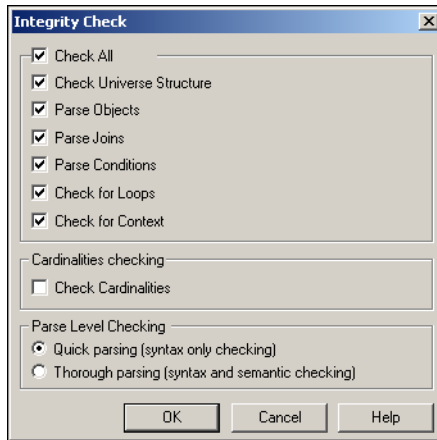
- a In the Edit Properties dialog box, edit the Name, data Type, Description, and database field as needed.
  - b In this example, change Fullname to My Fullname.
  - c In the Select text box, change the statement to AMCONTRACT\_D.MY\_FULLNAME.
  - d Click **Parse** to test the changes.
- 3 If successful, 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**.



- 2 Click **Check All**.

- 3 Click **OK**.

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

- 4 Correct any problems as needed.

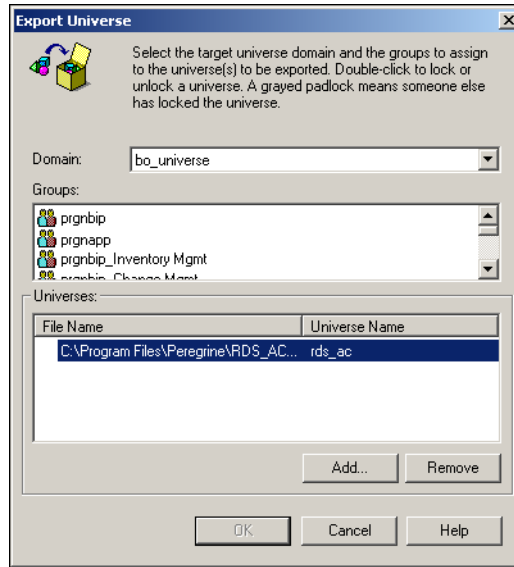
- 5 Save your changes.

## 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**.



- 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:

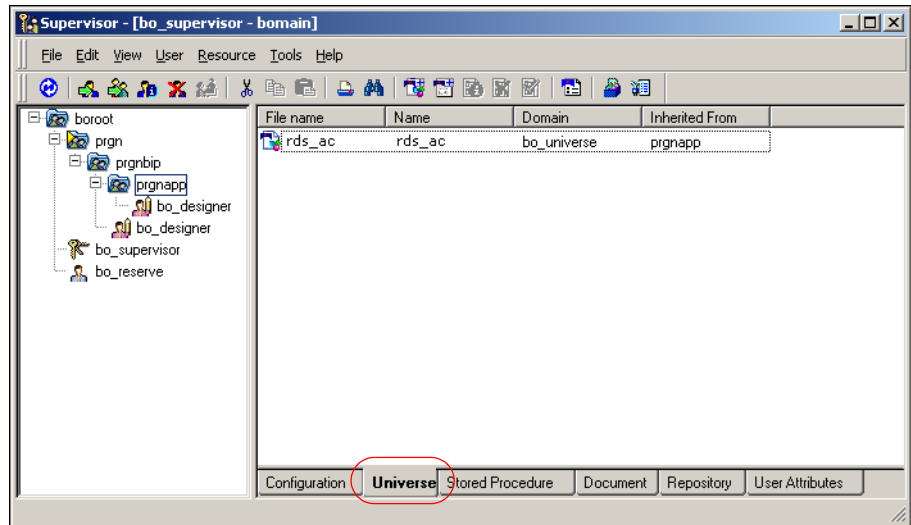
- 1 Log on to the Business Objects Supervisor as a user with Supervisor rights.
- 2 Select the group at the level you want to set your restrictions. If you want to restrict all users, go to your main application group. If you want to impose different restrictions for different sets of users, you can: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

### To set security restrictions on the sub-group:

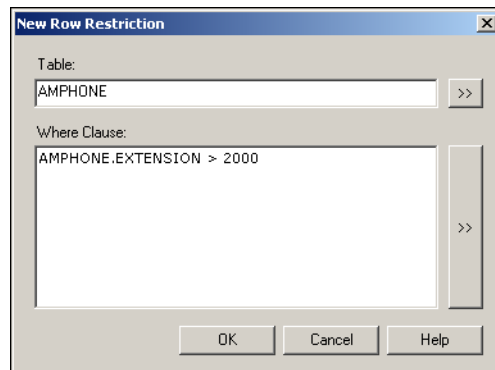
- 1 Highlight the sub-group.



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



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



- 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 AMPHONE.

- 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 AMPHONE table that refer to telephone extensions less than 2000 (AMPHONE . EXTENSION > 2000).

- 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.

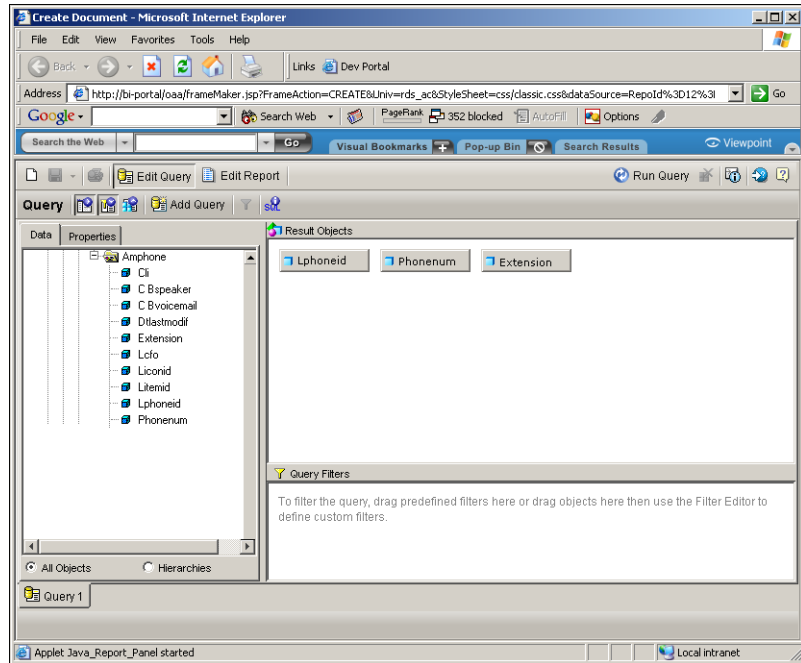
#### To test the restriction:

- 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.

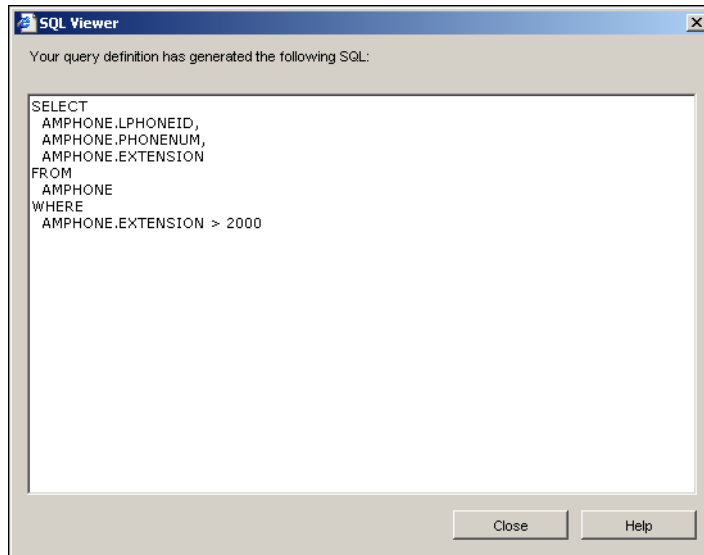
- 2** Click the **Reporting** icon.
- 3** Click **Create**.

#### 4 Click the rds\_ac universe to launch the Report Designer tool.



- 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.



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 103
- RDS System tables for tracking ETL processing on page 104
- Other useful tables on page 106

---

## 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 <b>Note:</b> 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..

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.
CAPABILITIES		VARCHAR2(500)	
APP_NAME		VARCHAR2(10)	

Name	Null?	Type	Description
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:

Copyright © 2001, Thai Open Source Software Center Ltd, Sun Microsystems. All rights reserved.

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

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

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.

Neither the names of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS 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 REGENTS OR 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.

## Jcrypt.java

This product includes software developed by Eric Young ([ey@mincom.oz.au](mailto:ey@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](mailto: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/>](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. Accepting 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

identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.



# Index

## PEREGRINE

### A

- aggregate keys 29
- aggregate of fact tables 15
- aggregate table 16
- aggregateKeys 29
- amAsset table 34, 35
- amOutputEvent 59, 60
- AssetCenter record deletion 31
- associate table 15, 16, 24
- automatic data synchronization 80

### B

- BIRECORDDELETE 15, 31, 60
- bridge table 15, 16

### C

- configure scenarios 39
- Connect-It console
  - configure scenarios 39
  - start scenarios 37
  - stop scenarios 38
- Connect-It scenario
  - add a new field to rds\_sc.scn 77
  - add mapping for a new table 66
  - add the new table to the scheduling process 74
  - Builder tool 64
  - customize 63
  - data synchronization process 80
  - edit 48
  - primary key field 73

- schedule synchronization 80

- customer support 10

### D

- data structures
  - add 45
  - modify and delete 45
- data synchronization 31, 33, 80
- deletion workflow 31, 57
- dimension attributes 29
- dimension table 14, 16, 21, 24
- direct mapping table 15, 16, 26, 27
- directMapping table 27
- document groups 13, 14

### E

- ETL mapping information 17
- extracting record delete information 60

### F

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

### H

- hierarchy table 15, 25

### I

- import script files 57, 59

### J

- join table 15

**M**

marking deleted records 60  
 metadata 16

**N**

naming conventions 16

**P**

Peregrine Systems customer support 10  
 planning the customization 44  
 primary key field 73

**R**

rdms\_keywords.xml 47  
 RDS scenario 64  
 RDS scenarios 33  
 RDS schema customization  
   add a new field 55  
   add a new table 52  
 RDS schema, editing 47  
 RDS time dimension table 106  
 rds\_ac 34  
 rds\_ac scenario 38, 39, 46  
 rds\_ac.ini 34  
 rds\_ac.log 34  
 rds\_ac.scn 33, 34, 47, 48, 63  
 rds\_ac.unv 47, 83, 88, 89  
 RDS\_acetl.xml 51, 52, 54, 56  
 rds\_acuser 34  
 rds\_acuser scenario 38, 42, 46  
 rds\_acuser.ini 34  
 rds\_acuser.scn 33, 34, 47  
 RDS\_CIT table 104  
 RDS\_CIT\_LOG STATE\_FLAG table 105  
 RDS\_CIT\_LOG table 104  
 RDS\_DBINFO table 103  
 rds\_etl.xml 16  
 rds\_etl.xsd 16, 20, 27  
 RDS\_ETL\_WORK table 104  
 RDS\_ETLSYNC\_LOG table 106  
 rds\_sc.scn 64  
 RDS\_SCDCOLUMN table 102  
 RDS\_SEC\_SYNC table 105  
 RDS\_SEQUENCE table 105  
 RDS\_UNIQUECOLUMN table 101

RDS\_USER table 103  
 RDSac\_etl.xml 13, 16, 20, 30  
 rdsac\_etl.xml 47  
 RDSInitRun command 30  
 record deletion  
   amOutputEvent 59  
   extract information 60  
   import script files 59  
   marking the records 60  
   VB script action definition 58  
   workflow 57  
 record deletion, AssetCenter 31

**S**

schedule Connect-It scenarios 80  
 schema data 16  
 schema definition 23, 27  
 script files 57, 59  
 security profiles, editing 49  
 start RDS scenarios 37  
 stop RDS scenarios 38  
 synchronize  
   automatic scheduling 80  
   data 33  
   record deletion 31  
 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  
 typographical conventions 9

**U**

universe 19



- add a field to the object browser 91
- add a new field 90
- add a new table 83
- add a table to the object browser 87
- edit 89
- edit a security profile 96
- edit the Context 86
- export to Business Objects 95
- modify an existing field 92
- test user rights 98
- validate the database links 94

universe, editing 48

## V

VB script, record deletion 58

## W

workflow

- back up critical files 46
- close programs and processes 46
- deletion 31, 57
- edit Connect-It scenario 48
- edit security profiles 49
- edit the RDS schema 47
- edit the universe 48
- planning 44
- restart programs and processes 49

## X

XML schema

- definition 29
- syntax 16

## Z

- Z\_RDSACTIVESTATUSIND 30
- Z\_RDSCREATEDDATE 29
- Z\_RDSDELETEDDATE 30
- Z\_RDSLASTMODDATE 29, 30
- Z\_RDSOBSOLETEIND 30
- Z\_RDSTRANSENDDATE 30
- Z\_RDSTRANSLASTIND 30





