User Manual

Data Extractor

SimCorp Solutions
Based on version 20.01
December 2019

SimCorp

PUBLISHED BY

SimCorp A/S
Weidekampsgade 16
2300 Copenhagen S
Denmark

Published	December 2019
Writer:	MMN
Based on:	20.01

The contents of this publication are for general information and illustrative purposes only and are used at the reader's own risk. SimCorp uses all reasonable endeavors to ensure the accuracy of the information. However, SimCorp does not guarantee or warrant the accuracy, completeness, factual correctness, or reliability of any information in this publication and does not accept liability for errors, omissions, inaccuracies, or typographical errors.

Electronic messaging for Collateral Management processes may access the MarginSphereTM Messaging Platform owned by AcadiaSoft, Inc., and for such access all terms and conditions set forth in the MarginSphereTM service agreement apply.

If you have any comments to the contents of this document or suggestions for topics to be included in a future updating of this document, please do not hesitate to send them by mail to: SCDimension.doc@SimCorp.com.

Contents

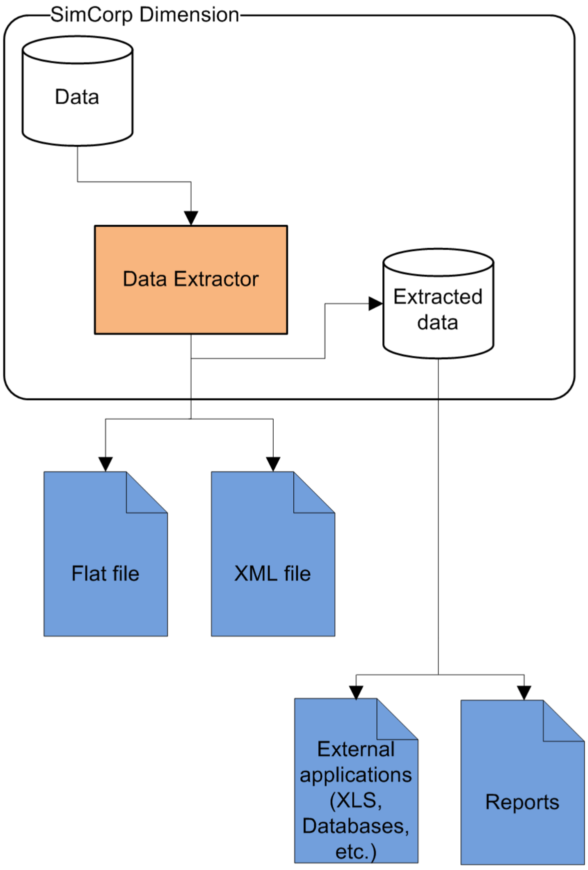
1 Data extractor explained

The data extractor is an integrated tool within SimCorp Dimension. Use the data extractor to extract data from the SimCorp Dimension user tables and place it into a single extract table. The extracted data is stored in separate Oracle® schemas, where it can be accessed for further processing.

You can extract or process the data to the following formats:

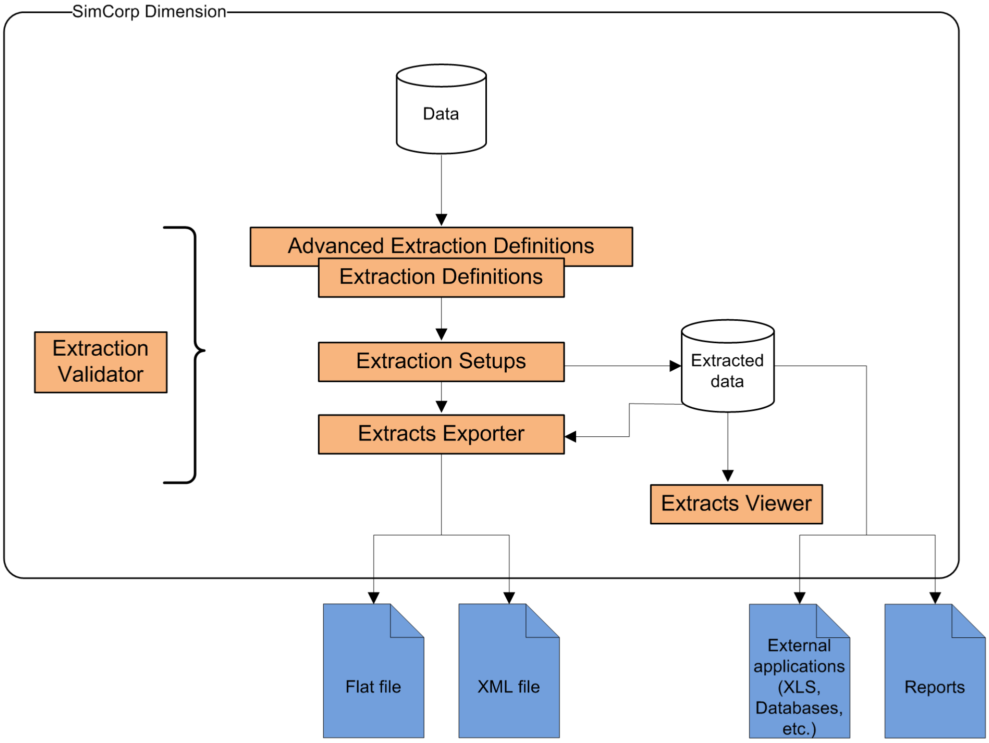
- Flat files
- XML files
- External applications (Microsoft Excel worksheets or other applications)
- Reports

You can also export data to the SimCorp Message Queue, the Communication Server, data warehouse schemas, or you can present the data as a screen view.



Components of the data extractor

The following components of the data extractor are used in defining the data and processing the extracts.



The components in the following table are used in the main workflow when you define and process a data extract.

Component	Description
Extraction Definitions	Use this window to define the data to be extracted from the database. See Create a basic extraction definition on how to select the data to be extracted.
Advanced Extraction Definitions	Use this window to define the data to be extracted from the database, with the ability to enter native SQL statements in the query. See Create an advanced extraction definition on how to select the data to be extracted, using your own SQL statements in the query.
Extraction Setups	Use this window to process the data extracts. This window is where you connect the extraction definition and the destination of the user schemas. Furthermore, this is where you execute the extracts, thus storing the data in the defined user schema. See Processing the data extracts on how to configure the **Extraction Setups** window to process the data extracts.
Extracts Exporter	Use this window to export data to a flat file or to an XML file. See Export data to flat file structure and Export data to XML file structure for more information.

Component	Description
Extracts Viewer	Use this window to display, print or export the data to a flat file via clipboard (only tab separated) when the data has been extracted and stored in a user schema. See Viewing extract table contents and structure.
Extraction Validator	Use this window to validate the functioning of the various elements of the data extractor that are saved in the installation. See Monitoring validity of data extractor objects for information on how to validate the various elements.

The following table describes additional components, which are used for specific purposes.

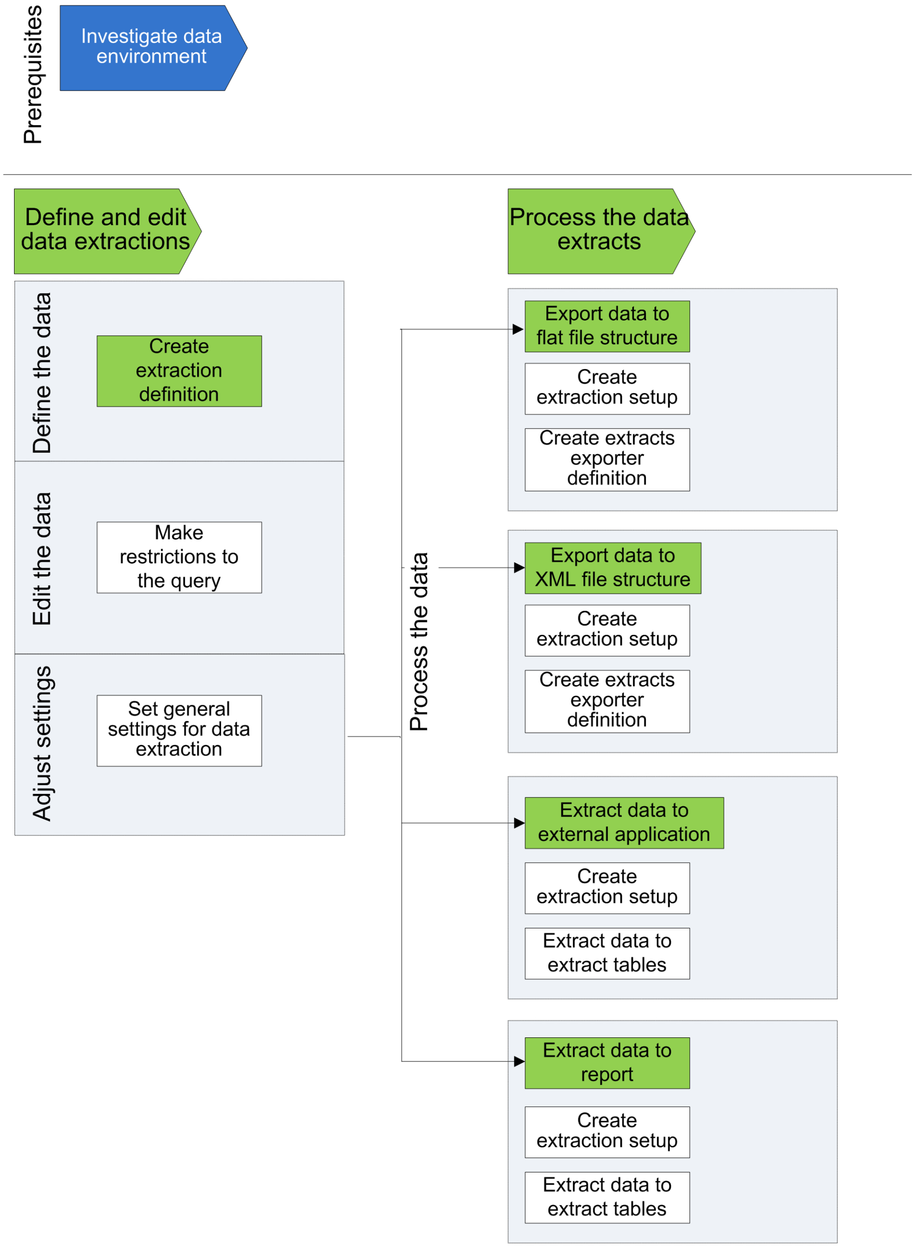
Component	Description
Extraction Formula Fields	Use this window to define the name and data type for fields that can be used in formulas. See Apply predefined formulas on how to apply predefined formulas to a data extraction when you are in the process of editing the data.
Extract Tables	Use this window to maintain extract tables in internal extract user and data warehouse schemas. See Viewing extract table contents and structure for more information.
Cleanup	Use this menu to delete un-referenced data extractor objects from the system. Un-referenced objects are objects that have no functional relations within the system. See Cleaning up data extractions and data extractor objects for information on how to clean up the various data extractor elements from your system.
Extract Fund Data	Use the data extraction fund event to execute a data extraction as a fund event, based on setups in the **Extended Fund Events** window. This fund event can be part of your fund event group. See the ***Fund Accounting Setup User Manual*** for more information.
Export Fund Data	Use the data export fund event to execute a data export as a fund event, based on setups in the **Extended Fund Events** window. This fund event can be part of your fund event group. See the ***Fund Accounting Setup User Manual*** for more information.

Find more documentation

The following SimCorp Dimension user manuals are useful when working with the Data Extractor:

- ***Batch Jobs User Manual***
- ***Communication Sever User Manual***
- ***Fund Accounting Setup User Manual***
- ***System Access User Manual***
- ***System Administrator's Manual***
- ***System Performance Reference Manual***
- ***Tools User Manual***

1.1 Workflow


Prerequisites
Investigate data environment
Define and edit data extractions
Define the data
Create extraction definition
Edit the data
Make restrictions to the query
Adjust settings
Set general settings for data extraction
Process the data extracts
Process the data
Export data to flat file structure
Create extraction setup
Create extracts exporter definition
Export data to XML file structure
Create extraction setup
Create extracts exporter definition
Extract data to external application
Create extraction setup
Extract data to extract tables
Extract data to report
Create extraction setup
Extract data to extract tables

Process	See
Prerequisites	Prerequisites
Define and edit data extractions	Defining and editing data extractions
Define the data	Define the data
Edit the data	Edit the data
Adjust settings	Adjust general settings for data extraction
Process the data	Processing the data extracts
Export data to flat file structure	Export data to flat file structure
Export data to XML file structure	Export data to XML file structure
Extract data directly to the SimCorp Message Queue	Extract data to the message queue
Extract data to external application	Extract data to an external application
Extract data to report	Extract data to a report
Extract data to a data warehouse schema	Extract data to a data warehouse schema

1.2 Prerequisites

Before the data extractor can be used in SimCorp Dimension, a number of tasks are required to be completed by an Oracle DBA on the Oracle database. These tasks are:

- Create/delete separate tablespaces in Oracle (can only be done with 3rd party tools, such as Oracle SQL*Plus)
- Create/delete external extract user schemas (only with 3rd party tools – schema can then be registered in SimCorp Dimension)

The data owner (SCDAT) must be granted with ANALYZE ANY privilege. If the privileges are not granted, it is not possible for SimCorp Dimension to gather statistics in the Data Extractor users table, because it belongs to another user.

For information on how to complete these tasks, see the ***System Administrator’s Manual***.

Before you extract data from SimCorp Dimension, investigate the data environment to determine which tables and fields you want to extract data from. See Database structure introduced for information about how to identify the required tables and fields in the system.

1.3 Database structure introduced

SimCorp Dimension is integrated with the Oracle database. The database consists of system tables and data tables.

The system tables are considered part of the program and are kept in a separate tablespace from the data tables.

The SimCorp Dimension tablespaces typically include:

- SCSYS — Application metadata
- SCDAT — Data owner's tablespace, which contains application data
- SCINDX — Data owner's indexes
- SCAUDIT — Application audit (historic) data
- SCAUDIX — Audit indexes

It is recommended to use database views when creating the data extractions. A database view is an SQL query that is stored in the SimCorp Dimension database. The database view collects fields from different database tables.

The main reasons for using database views in the data extracts:

- Maintainability — SimCorp ensures that the database views are working when the database structure changes.
- Easier to use — Database views require less joining and are more targeted towards the end-user.

Note

Consider database tables if you experience poor system performance of the data extracts.

This section describes the following concepts:

- Table groups
- Data in the window
- Tools for investigating the database
- Select statement hierarchy

1.3.1 Table groups

All user tables are assigned both a table group and a sub-table group. The table groups are organised based on the business model (modules), and the sub-table groups categorise the type of data stored in the tables.

Table Groups	Table Groups (cont.)	Table Groups (cont.)
Accounting	Holdings	Pricing

Table Groups	Table Groups (cont.)	Table Groups (cont.)
Archiving	Integration	Rebalancing
Audit trail for 4 eyes principle	Interface Specific	Reconciliation
Calculation figures	Limits	Reference data
Client management	Market data	Reporting
Client reporting	Market specific	Risk Management
Collateral	Orders	Securities
Compliance	Performance	System
Costs	Portfolio calculation	System access
Data extractor	Portfolio management	Transactions
Funds	Portfolios	

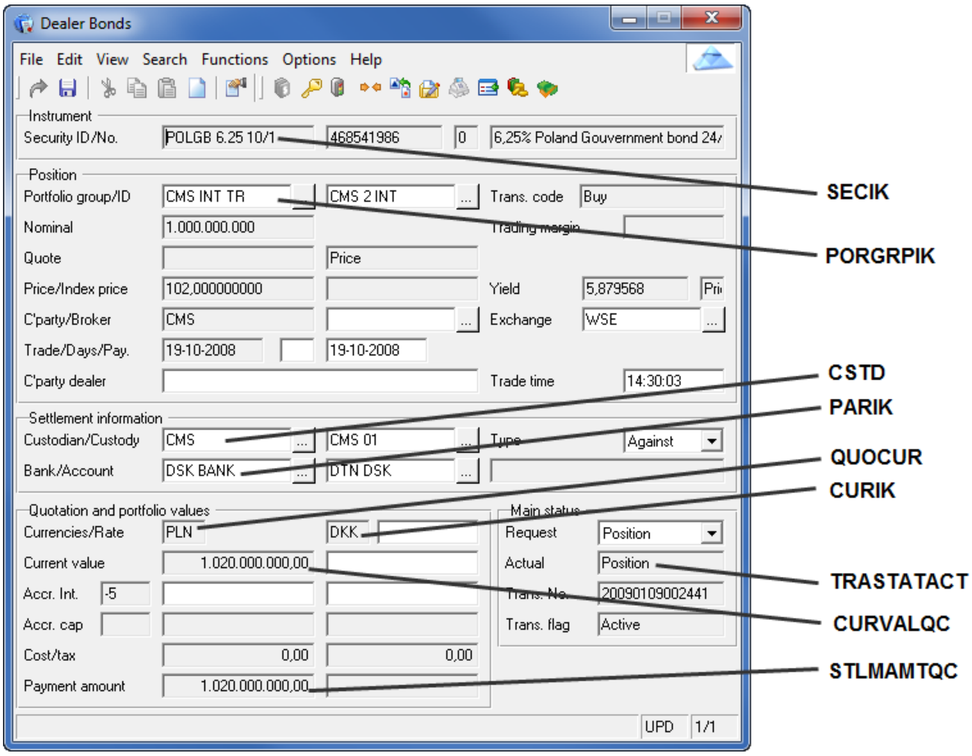
The sub-table groups are used to control the default audit configuration, and reflect the type of data stored in the tables:

- **Adjustable data** — Usually rather large data volumes (for example, Prices), which might be imported from other sources but still can be modified using standard functionality iwn SimCorp Dimension.
- **Free codes** — Similar to static data, but separated due to the large number of freecode tables.
- **Generated data** — System generated data that can be recreated and not directly manipulated by the users.
- **Not auditable data** — System generated data that will significantly decrease performance if audit trail is enabled for the table.
- **Static data and configuration** — Static data, setups, configuration, parameters, and reference data.

1.3.2 Data in the window

The data are displayed in varying combinations of data type in the windows of the system. For example, the **Dealer Bonds** window comprises:

- Static data — **Security ID/No.** (SECIK), **Bank/Account** (PARIK), and **Currencies** (from quotations) (QUOCUR)
- Reference data — **Currencies** (from portfolios) (CURIK)
- Transactions — **Actual** (TRASTATACT), **Current value** (CURVALQC) and **Payment amount** (STLMAMTQC)
- Holding keys — **Custodian/Custody** (CSTD) and **Portfolio group/ID** (PORGRPIK)



Instrument

Security ID/No. POLGB 6.25 10/1 468541986 0 6,25% Poland Gouvernment bond 24/

Position

Portfolio group/ID CMS INT TR CMS 2 INT Trans. code Buy

Nominal 1.000.000.000 Trading margin

Quote Price

Price/Index price 102,000000000 Yield 5,879568 Pri

C'party/Broker CMS Exchange WSE

Trade/Days/Pay. 19-10-2008 19-10-2008

C'party dealer Trade time 14:30:03

Settlement information

Custodian/Custody CMS CMS 01 Type Against

Bank/Account DSK BANK DTN DSK

Quotation and portfolio values

Currencies/Rate PLN DKK

Current value 1.020.000.000,00

Accr. Int. -5

Accr. cap

Cost/tax 0,00 0,00

Payment amount 1.020.000.000,00

Main status

Request Position

Actual Position

Trans. No. 20090109002441

Trans. flag Active

UPD 1/1

1.3.3 Tools for investigating the database

SimCorp Dimension provides the following tools to investigate the field and table structure in the system:

- Database overview tool
- Technical information

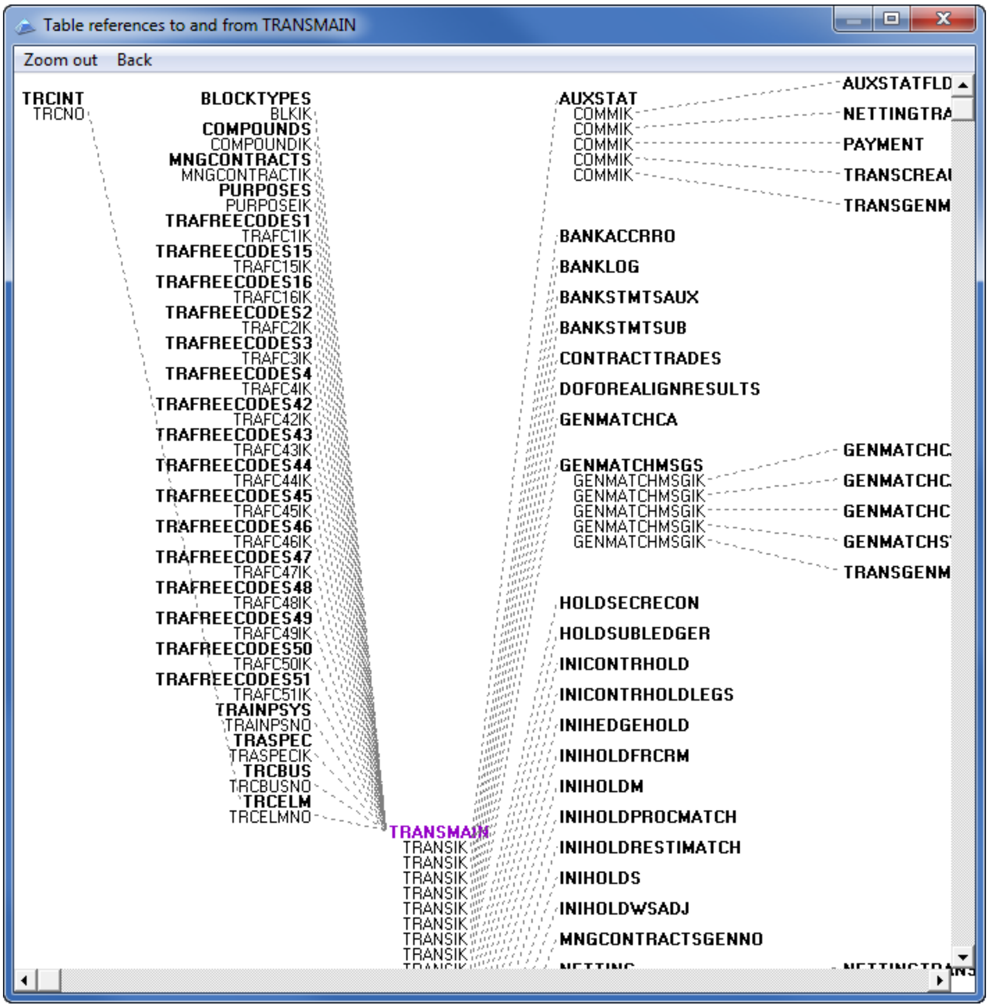
Use these tools to determine from which tables and fields to extract data.

1.3.3.1 Database overview tool

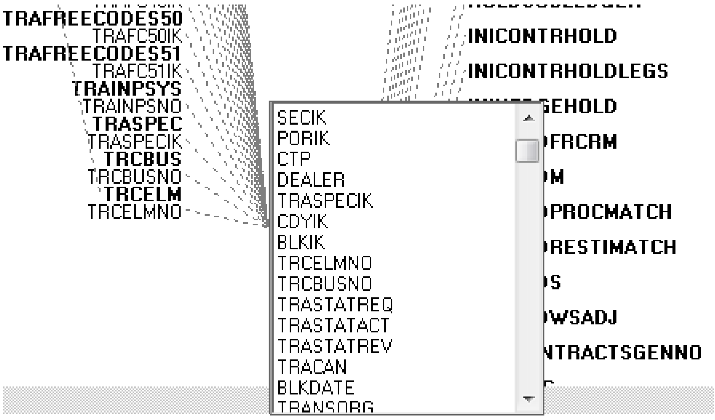
Use the **Database Overview** tool to see how the database is organised in SimCorp Dimension and to see how report views are constructed. This tool presents a representation of the relationship between tables in the system.

You can select a table group to restrict the query to only tables in that group. When a table group is selected, only tables and report views belonging to the group are displayed in the graph. If no table group is selected, all tables and report views are available for selection.

You can see the internal keys for the tables by pressing **Zoom in**.



Open the technical help for a selected field by clicking the table name and selecting a field name from the list.



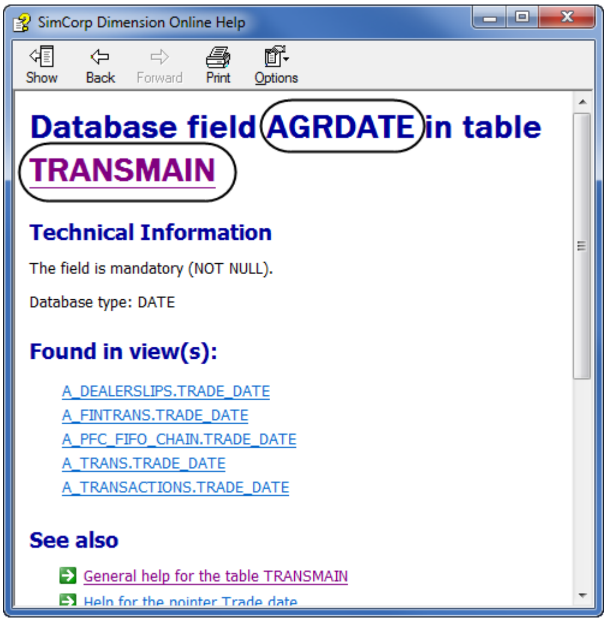
1.3.3.2 Technical information

The online help in SimCorp Dimension contains information about the tables, fields and database views in the SimCorp Dimension database.

To find the technical information for a field in the online help, right-click the field and select **Help**. In the online help, click **Technical Information**.

You can also browse for tables and report views in the table of contents of the help system. Expand the **Database Tables** or **Report Views** nodes in the online help (on the **Contents** tab).

The technical help topic provides you with the database field name and the table where the data is stored.



Database field AGRDATE in table TRANSMAIN

Technical Information

The field is mandatory (NOT NULL).

Database type: DATE

Found in view(s):

A_DEALERSLIPS.TRADE_DATE
A_FINTRANS.TRADE_DATE
A_PFC_FIFO_CHAIN.TRADE_DATE
A_TRANS.TRADE_DATE
A_TRANSACTIONS.TRADE_DATE

See also

General help for the table TRANSMAIN
Help for the pointer Trade date

1.3.4 Select statement hierarchy

The Data Extractor retrieves data from one or more tables by using an SQL query.

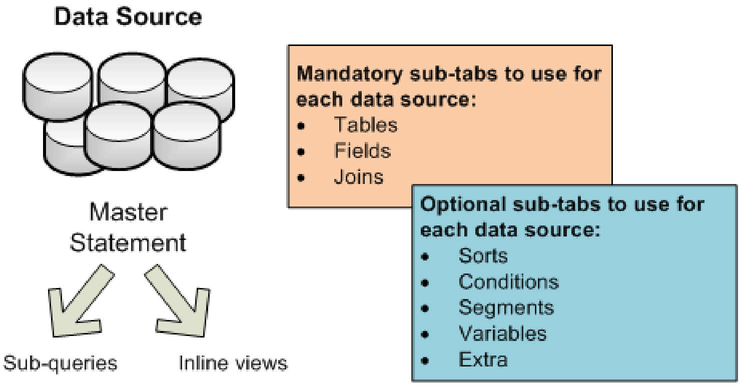
In SimCorp Dimension, the top-level SELECT statement in the query is the MASTER statement. An extraction definition always contains a MASTER statement at the top level. A query that is nested within another SQL statement is a sub-statement. There are two types of sub-statement:

- Inline view — Used in the **Tables** part of the parent statement

  See Create sub-statements to the master: Inline views

- Sub-query — Used in the **Joins** part of the parent statement

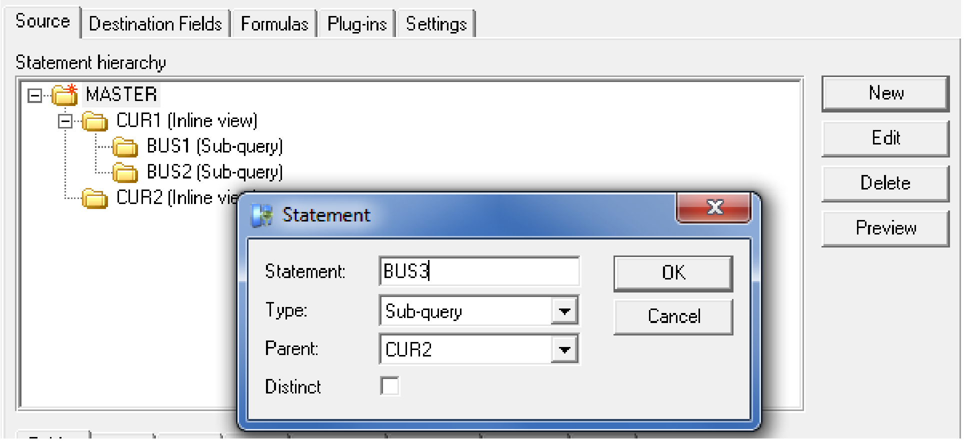
  See Create sub-statements to the master: Sub-queries



In SimCorp Dimension, the MASTER statement is used both to define source data and to define the destination fields. Inline views and sub-queries are used to define (and refine) source data.

The use of sub-statements can greatly improve the runtime performance of your extraction. Refer to ***System Performance Reference Manual*** for information on optimising performance.

The statement hierarchy is created and edited in **Extraction Definitions** window, in the **Source** tab, in the **Statement hierarchy** pane.



Statement: BUS3
Type: Sub-query
Parent: CUR2
Distinct ☐

Note that SimCorp terminology differs slightly from that of Oracle®.

SimCorp term	Oracle® term
Master	Query
Inline view	Sub-query in the FROM clause
Sub-query	Sub-query in the WHERE clause
SQL Expression	Expression included in the SELECT statement

2 Defining and editing data extractions

The first step to create a data extraction is to define the data to be extracted and to edit the query. Follow the workflow to create a data extraction.

Process	See
Identify and select the source of the data that you want to extract.	Section Define the data: • Create a basic extraction definition • Create an advanced extraction definition
Preview the data to ensure it is what you want.	Section View preview data, SQL query and execution plan
Edit and tune the data extraction by using the following:	Section Edit the data
• Create segments	Section Specify the records to export by using segments
• Create conditions	Section Refine the data extraction by using conditions
• Add complexity to the conditions	Section Add complex conditions to the extraction definition
• Add variables	Section Add variables for executing the extraction definition
• Group and sort the output	Section Group and present data in a sorted order
• Add formulas	Section Apply predefined formulas
• Include export translation	Section Add conversion rules using export translations
• Use plug-ins	Section Apply predefined business logic
• Aggregate by using inline views	Section Create sub-statements to the master: Inline views
• Modularise by using sub-queries	Section Create sub-statements to the master: Sub-queries
• Transform data using SQL expressions	Section Transform data by using SQL expressions
Combine multiple queries using Union	Section Combine multiple queries using Union
Define the general settings	Section Adjust general settings for data extraction

Process	See
Authorisation in Extraction Definitions	Section Authorisation in Extraction Definitions

Before you begin

Before you extract data from SimCorp Dimension, investigate the data environment to determine which tables and fields you want to extract data from. See Database structure introduced for information about how to identify the required tables and fields in the system.

If you are specifying your extraction definition by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

- See the ***Tools*** user manual for information on how to create a general segment.
- See Specify the records to export by using segments to apply the segment to the extraction definition.
- See Processing the data extracts to apply segments to the extraction setup.

If you are using formulas in your extraction definition, you must first create the formula. Formulas enable you to make calculations that are not pre-defined by SimCorp Dimension and that can be applied in multiple modules in the system.

- See the ***Tools*** user manual for information on how to create formulas.
- See Apply predefined formulas to apply the formula to the extraction definition.

2.1 Define the data

There are two approaches to defining the data to be extracted:

Create a basic extraction definition	A basic extraction definition allows you to use the built-in functionality in the **Extraction Definitions** window to define and edit an extraction definition.
Create an advanced extraction definition	An advanced extraction definition allows you to enter native SQL when defining and editing an extraction definition.

2.1.1 Create a basic extraction definition

Use the **Extraction Definitions** window to define the source of the data to extract and the destination in a basic extraction definition.

Note

Because it could take time to complete a valid extraction definition, you can select the **Incomplete definition** check box and save the extraction definition while it is in development.

1. In the **Extraction Definitions** window, select **File** > **New** and enter an ID and name for the extraction definition.
2. Select the tables or views from which to extract data by selecting an available field in the **Table** column on the **Source** > **Tables** tab and pressing F4.

   A list of all available tables and views appears. Select the required tables and views from the list and click **OK**.

   A. You are prompted to edit the table alias names. Edit the table alias names if required, and click **OK**.

   B. If there are possible joins between the tables, you are prompted to accept or alter the proposed joins. Select the relevant joins and click **OK**.

   The relevant fields that comprise the joins are added to the extraction definition.

   You can see the joins between tables on the **Source** > **Joins** tab, and you can see the fields used to create those joins on the **Source** > **Fields** tab.

3. Select the required fields from those tables that have been added to the extraction definition. There are two ways to add fields:
    - Automatic — Use the **Data Dictionary Browser**:

      This tool simplifies the process of linking between tables by providing graphical navigation and creating joins automatically.

      i. Select **Functions** > **Select Field via Data Dictionary Browser**.

      ii. Select the table from the **Table alias** drop down list, and select the relevant fields from amongst the possible fields for each table alias.
        - Select all those fields from which you want to extract data.
        - Select all those fields that are required to create the joins between the tables. Fields with a (+) appended to the name designate underlying fields that are linked to other tables. Select the expandable field and click **Expand** to expose those fields that link to other tables. Selecting these fields adds missing tables to the extraction definition and creates automatic joins.
        - If you are using formulas in your extraction definition, select all those fields that are required in the formula.
    - Manual — Use the **Source** > **Fields** tab:

      You can select fields from which to extract data and create the necessary joins manually.

      i. On the **Source** > **Fields** tab, select an available field in the **Table alias** column and press F4. A list of tables that have been added to the extraction definition appears. Select the table alias on which the required field is located and click **OK**.

      ii. Select the accompanying field in the **Field** column and press F4. A list of fields that are in the selected table appears. Select the required fields from which to extract data and click **OK**.

      iii. If desired, you can edit the field names in the **Field alias** column.

4. Modify your field selection. On the **Source** > **Fields** tab:

   A. Select whether some fields should use domain translation values:

   - If you select the **Domain translation** check box, the extracted data will contain the text representation of the field.
   - If you clear the **Domain translation** check box, the extracted data will contain the numerical representation of the field.

   B. If you want to aggregate the output of any given field, you can select from amongst the following aggregation methods in the **Aggr. Function** column:

   - **Average** — Returns the arithmetic mean on the source field set.
   - **Count** — Returns the aggregated number of the source field set.
   - **Maximum** — Returns the largest value of the source field set.
   - **Minimum** — Returns the smallest value of the source field set.
   - **Sum** — Returns the summation of the source field set.
   - **Variance** — Returns the sample variance of the source field set.
   - **Standard deviation** — Returns the square root of 'Variance'.

   For complex aggregations you may require inline views. See Create sub-statements to the master: Inline views for more information.

Note

If an aggregation function is applied to any of the fields, all fields without an aggregation function are added to the GROUP BY clause of the resulting SQL statement. This means that fields without an aggregation function will be grouped.

5. Verify and create the relevant joins on the **Source** > **Joins** tab.

   If your extraction definition has more than one table in the MASTER statement, or if there is more than one statement in the extraction definition, the tables must be joined. Joins are created in the following ways:

   - Automatic — When you add fields to the extraction definition by using the **Data Dictionary Browser**, tables are joined on IK fields automatically. To verify that all relevant joins are added, use the **Functions** > **Suggest Joins** tool to suggest and add missing joins.
   - Manual — On the **Source** > **Joins** tab, select an available field in the **From field** column and press F4. Select the relevant IK or REF field to join from. Likewise, select an available field in the **To field** column and press F4. Select the relevant IK or REF field to join to.
   - If required, make a left outer join.

     To make a left outer join, select the **Outer join** check box. The result of a left outer join for table A and B always contains all records of the **From field** (left table (A)), even if the join-condition does not find any matching record in the **To field** (right table (B)).
   - If required, adjust the operator of the join.

     In the **Operator** column, you can define the type of join, where the relation between the left table (A) and the right table (B) could be:

     - = (equal)
     - < (less than)
     - <= (less than or equal)
     - > (greater than)
     - >= (greater than or equal)
     - <> (not equal)

6. Define the destination fields for the extraction definition, where you design the output format of the data extraction.

   You can link the destination fields to a destination table layout in internal user and data warehouse schemas by selecting a destination table in the **Destination table** field. Information about the primary keys and the destination table layout is read when you select a destination table. The **Mandatory field** indicates the primary key fields, not null fields and other mandatory fields, and the field is cleared when a destination table is not selected or when the destination table is from an internal user schema.

   A. Optional: In the **Destination table** field, select a destination table in internal user or data warehouse schemas to link the destination fields.

   B. On the **Destination Fields** tab, in the **Source field** column, select from amongst the fields that have been added as source fields. Select those fields that you want in your output.

   C. In the **Field order** column, enter the order in which you want the data to be presented.

   D. If required, in the **SQL name** field, edit the SQL names to be used in the extraction. This field maps the source field to the target. By default, SimCorp Dimension inserts the original SQL name from the source.

   E. If required, you can override the values for **Data type**, **Field length**, and **Decimals** for each destination field by clearing the **Use data dictionary** check box and editing the appropriate fields.

      If the **Use data dictionary** check box is selected (recommended), then the field dimensions are copied from the internal data dictionary. It is recommended to use the default settings to prevent any irregularities between the data and the format of fields.

      If the data from the source field does not fit the edited format of the field in the extract table, you will not be able to validate the extraction definition and will be notified of the inconsistency when you try to save.

7. Save the extraction definition. This will check for errors and validate the extraction definition.

   You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

If required, you can now edit the extraction definition in order to refine the queries and add additional formatting. See Edit the dataEdit the datafor more information.

Defining the destination layout

You can define the destination layout in the **Extraction Definitions** window on the **Destination Fields** tab by using the following methods:

- Define a destination layout synchronised with source tables — Select the **Use data dictionary** check box. The fields are dependent on the source tables, and will be updated or removed automatically when loading/validating.
- Define a fixed destination layout — Clear the **Use data dictionary** check box. The fields are not automatically changed when loading/validating.
- Define a destination layout as a copy of an extract/data warehouse table layout — Select the **Read Table Layout** function to add a copy of the selected table layout. You must manually specify source fields and formula fields to map source data to destination fields. There is no link between the destination layout and the selected table.
- Link the destination fields to a destination table layout — Select a destination table (in internal user or data warehouse schemas) in the **Destination table** field on the **Destination Fields** tab. Information about the primary keys is read together with information about the destination table layout, when you select a destination table.

2.1.2 Create an advanced extraction definition

Use the **Advanced Extraction Definitions** window to define the source of the data to extract and the destination in an advanced extraction definition, where you can enter native SQL to refine your query.

Note

You can open a basic extraction definition and modify it in the **Advanced Extraction Definitions** window, but you cannot modify an advanced extraction definition in the basic **Extraction Definitions** window.

1. In the **Advanced Extraction Definitions** window, select **File** > **New** and enter an ID and name for the extraction definition.
2. In the **Definition type** drop down list, select **Advanced extraction definition**.
3. On the **Advanced Source** tab, enter the SQL statement in the **Query** field.

4. On the **Advanced Source** > **Fields** sub-tab, select the required fields for the extraction definition.
   - A. Select the required fields in the **Field alias** column.
   - B. Fill in the **Field order** field, so the sequence of fields on the **Fields** sub-tab is the same as they appear in the **Query** field.
   - C. Select the data type for each field alias.
   - D. Enter a field length for each field alias.

5. If required, add conditions or segments by using TABCOND syntax to the statement in the **Query** field.

   Use SQL KEYWORDS such as WHERE, OR, AND.

   Enter one line per table.

   The TABCOND syntax is:

   ```
   /*TABCOND,NAME,TABLE,TABLE ALIAS,SQL
   KEYWORD,DESCRIPTION,*/
   ```

   The configuration is automatically updated to the **Advanced Source** > **Table conditions** sub-tab.

6. Define the destination fields for the extraction definition, where you design the output format of the data extraction.

   You can link the destination fields to a destination table layout in internal user and data warehouse schemas by selecting a destination table in the **Destination table** field. Information about the primary keys and the destination table layout is read when you select a destination table. The **Mandatory field** indicates the primary key fields, not null fields and other mandatory fields, and the field is cleared when a destination table is not selected or when the destination table is from an internal user schema.

   A. Optional: In the **Destination table** field, select a destination table in internal or data warehouse user schemas to link the destination fields.

   B. On the **Destination Fields** tab, in the **Source field** column, select from amongst the fields that have been added as source fields. Select those fields that you want in your output.

   C. In the **Field order** column, enter the order in which you want the data to be presented.

   D. If required, in the **SQL name** field, edit the SQL names to be used in the extraction. This field maps the source field to the target. By default, SimCorp Dimension inserts the original SQL name from the source.

   E. If required, you can override the values for **Data type**, **Field length**, and **Decimals** for each destination field by clearing the **Use data dictionary** check box and editing the appropriate fields.

      If the **Use data dictionary** check box is selected (recommended), then the field dimensions are copied from the internal data dictionary. It is recommended to use the default settings to prevent any irregularities between the data and the format of fields.

      If the data from the source field does not fit the edited format of the field in the extract table, you will not be able to validate the extraction definition and will be notified of the inconsistency when you try to save.

7. Save the extraction definition. This will check for errors and validate the extraction definition.

   You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

If required, you can now edit the extraction definition in order to refine the queries and add additional formatting. See Edit the data for information on how to edit the extraction definition.

Defining the destination layout

You can define the destination layout in the **Advanced Extraction Definitions** window on the **Destination Fields** tab by using the following methods:

- Define a destination layout synchronised with source tables — Select the **Use data dictionary** check box. The fields are dependent on the source tables, and will be updated or removed automatically when loading/validating.
- Define a fixed destination layout — Clear the **Use data dictionary** check box. The fields are not automatically changed when loading/validating.
- Define a destination layout as a copy of an extract/data warehouse table layout — Select the **Read Table Layout** function to add a copy of the selected table layout. You must manually specify source fields and formula fields to map source data to destination fields. There is no link between the destination layout and the selected table.
- Link the destination fields to a destination table layout — Select a destination table (in internal user or data warehouse schemas) in the **Destination table** field on the **Destination Fields** tab. Information about the primary keys is read together with information about the destination table layout, when you select a destination table.

2.2 Using Oracle views to integrate external data

Oracle views in **Data Warehouse schemas** can be registered in SimCorp Dimension. You can create an Oracle view that is based on an external data source (for example, flat file or external databases) and then use the view in the **Extraction Definitions** window.

You must create and support the Oracle views.

You can use the Oracle views as both data sources and destinations in the Data Extractor. The Oracle view can be used as a local object that can reference other data sources, both local and remote.

Note

Oracle views are supported for **Data Warehouse schemas** only.

The following requirements must be implemented for the Oracle views that are used as destinations in the Data Extractor (that is, insert, update or delete records in the Oracle view):

- Each Oracle view must have a primary key. The primary key must be created in a DISABLE or DISABLE NOVALIDATE state (see Oracle product documentation).
- Each Oracle view must be updatable (see Oracle product documentation for more information about updating a join view).

To create an Oracle view based on remote tables:

1. Create a Database Link to the remote data source. For example:

   ```
   CREATE DATABASE LINK <dblink_name> CONNECT TO
   <remote_user> IDENTIFIED BY <remote_password> USING
   <connection_string>
   ```

2. Create a view using the <remote_table>@<dblink_name> syntax for remote tables. For example:

   ```
   CREATE OR REPLACE VIEW <view_name> (<field_list>,
   CONSTRAINT <pk_name> PRIMARY KEY (<pk_field_name>)
   DISABLE) AS SELECT <field_list> FROM <table_
   name>@<dblink_name>;
   ```

3. Open the **Database User Schemas Contents** window and register the view.
4. Use the view both as source and destination.

2.3 View preview data, SQL query and execution plan

The **Extraction Definition** window and **Extraction Setup** window have some useful functions that you can use to check the construction of your data extraction:

- **Preview Data**

  After you create and save a valid extraction definition, you can see an example of how the extracted data will appear. Previewing data is useful in order to examine whether the extraction definition fetches the required data.

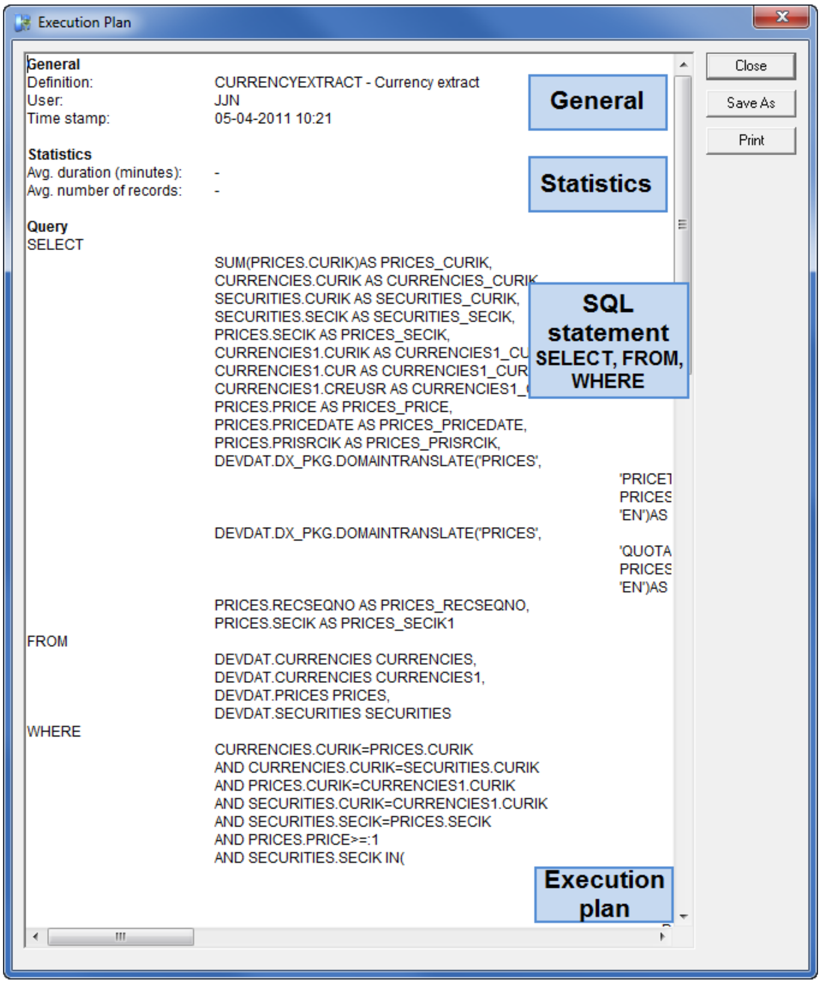
  Open **Functions** > **Preview Data**. The example consists of the first 500 records in the order of the sorting defined in the **Source** > **Sorts** tab.

- **Execution Plan**

  Open **Functions** > **Execution Plan**.

  The **Execution Plan** window shows the following information about the extraction definition:

  - Extraction definition ID and name
  - User who created the extraction definition
  - Time stamp
  - Statistics (Statistics are not shown in extraction definitions. They are generated from the extraction setup when the extraction definition is executed)
  - SQL query of the extraction definition
  - Execution plan of the extraction definition



```
General
Definition:                  CURRENCYEXTRACT - Currency extract
User:                        JJN
Time stamp:                  05-04-2011 10:21

Statistics
Avg. duration (minutes):     -
Avg. number of records:      -

Query
SELECT
                             SUM(PRICES.CURIK)AS PRICES_CURIK,
                             CURRENCIES.CURIK AS CURRENCIES_CURIK
                             SECURITIES.CURIK AS SECURITIES_CURIK,
                             SECURITIES.SECIK AS SECURITIES_SECIK,
                             PRICES.SECIK AS PRICES_SECIK,
                             CURRENCIES1.CURIK AS CURRENCIES1_CU
                             CURRENCIES1.CUR AS CURRENCIES1_CUR
                             CURRENCIES1.CREUSR AS CURRENCIES1_
                             PRICES.PRICE AS PRICES_PRICE,
                             PRICES.PRICEDATE AS PRICES_PRICEDATE,
                             PRICES.PRISRCIK AS PRICES_PRISRCIK,
                             DEVDAT.DX_PKG.DOMAINTRANSLATE('PRICES',
                                                                     'PRICET
                                                                     PRICES
                                                                     'EN')AS
                             DEVDAT.DX_PKG.DOMAINTRANSLATE('PRICES',
                                                                     'QUOTA
                                                                     PRICES
                                                                     'EN')AS
                             PRICES.RECSEQNO AS PRICES_RECSEQNO,
                             PRICES.SECIK AS PRICES_SECIK1
FROM
                             DEVDAT.CURRENCIES CURRENCIES,
                             DEVDAT.CURRENCIES CURRENCIES1,
                             DEVDAT.PRICES PRICES,
                             DEVDAT.SECURITIES SECURITIES
WHERE
                             CURRENCIES.CURIK=PRICES.CURIK
                             AND CURRENCIES.CURIK=SECURITIES.CURIK
                             AND PRICES.CURIK=CURRENCIES1.CURIK
                             AND SECURITIES.CURIK=CURRENCIES1.CURIK
                             AND SECURITIES.SECIK=PRICES.SECIK
                             AND PRICES.PRICE>=:1
                             AND SECURITIES.SECIK IN(
```

- **Direct Data Copy**

  The extract data is processed (selected and inserted) on the database layer, which improves the system performance of the data extraction process.

  Select **Functions** > **Check for Direct Data Copy**.

2.4 Remove unused tables and fields

In the **Extraction Definitions** window, you can remove the tables and fields on the **Source** tab that are not used in the source fields in the master statement, and that are not used by the formulas.

The function makes it easier to create and maintain more complex Extraction Definitions.

To remove unused tables and fields:

1. Open an extraction definition in the **Extraction Definitions** window.
2. Select **Functions** > **Remove Unused Elements**.
3. Save.

2.5 Edit the data

After the general extraction definition is set up, you can manipulate the data selection in various ways. Use the following procedures to edit your data extractions:

- Specify the records to export by using segments
- Refine the data extraction by using conditions
- Add complex conditions to the extraction definition
- Add variables for executing the extraction definition
- Group and present data in a sorted order
- Apply predefined formulas
- Add conversion rules using export translations
- Apply predefined business logic
- Create sub-statements to the master: Inline views
- Create sub-statements to the master: Sub-queries
- Transform data by using SQL expressions

2.5.1 Specify the records to export by using segments

You can specify the records that you want to export from the selected tables and views by creating general segments.

Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

You can use general segments to improve the system performance, especially during complex queries, by restricting the number of records that are processed in the inline views.

Note

A segment in the **Batch Job** window overrides a segment in the **Extraction Setup** window, and a segment in the **Extraction Setup** window overrides a segment in the **Extraction Definitions** window.

1 First, create a general segment in the **General Segments** window.

The tables in the general segment must correspond with the tables in your extraction definition. See the ***Tools*** user manual for information on how to create a general segment.

2 In the **Extraction Definitions** window on the **Source** > **Segments** tab, apply the required general segments to the appropriate tables that are included in your extraction definition.

You can specify multiple segments per statement. It is possible to use the same segment for several tables, and the same table can be used with several segments. Segments that are applied to the same table are grouped using the AND operator.

3 If required, select the segments as variables on the **Source** > **Variables** tab. See Add variables for executing the extraction definition for more information.

4 Save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

2.5.2 Refine the data extraction by using conditions

You can refine the extract definition query by adding conditions. These search conditions are applied directly to the extraction definition, and apply to fields in the selected tables.

Note

You can also apply search conditions to extraction setups. Search conditions in the extraction definition and the extraction setup can be used at the same time. The conditions in the extraction setup further refine the data extraction. This allows an extraction definition to be used with multiple extraction setups, thus producing different data extractions.

1 In the **Extraction Definitions** window on the **Source** > **Conditions** tab, click **Edit**.

You are prompted to select a table amongst those that have been added to the extraction definition.

2 Select the table to which the condition should be applied.

The **Conditions** window appears.

3 In the **Field** drop down list, select the field from the table that defines the condition.

4 Select the appropriate operator and value to define the condition, and click **Add**.

You can apply conditions to more than one field and to more than one table.

5 Click **OK** and then save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

6 If required, select the conditions as variables on the **Source** > **Variables** tab. See Add variables for executing the extraction definition for more information.

To edit or delete existing conditions

You can change or remove existing conditions that are in the extraction definition.

1. On the **Source** > **Conditions** tab, click **Edit**.
2. Select the table to which the existing condition applies.

   To identify the required table, you can see the components of the existing conditions in the **Search conditions** pane.
3. In the **Conditions** window for the selected table, select the condition that you want to edit or delete.
   - To edit the condition, click **Edit**.

     The **Field**, **Operator** and **Value** fields can now be modified.
     - To add a condition to the existing condition, click **Add**.
     - To change the values in the existing condition, click **Replace**.
   - To delete the condition, click **Remove**.

     The search condition is removed from the extraction definition.
4. Click **OK** and then save the extraction definition. This will check for errors and validate the extraction definition.

2.5.3 Add complex conditions to the extraction definition

You can create advanced conditions with SQL statements. These conditions are useful for segmenting the data, and are defined on the **Source** > **Extra** tab.

The advanced condition is of two parts:

- The **Optimiser mode**, which defines the execution plan for the data query
- The **SQL condition**, which is the SQL statement. The specified conditions are appended to the WHERE clause of the current select statement.

1. In the **Optimiser mode** drop down list, select the mode to be supplied when issuing the data query to the database.

   The optimiser of the database system calculates the execution plan for the data query before the execution takes place. This makes the data retrieval as efficient as possible. The work of the optimiser is mainly based on statistical information about the tables, indices and other database objects that are part of the data query.

   In most cases you would like the database system to optimize for best throughput; that is, use the least amount of resources necessary to process all rows accessed by the data query. Automated processes benefit from such behaviour.

But the database system can also optimise a statement with the goal of best response time; that is, using the least amount of resources necessary to process the first row accessed by the data query. This gives you the first rows as soon as they are found in the database, disregarding the resource consumption to complete the query. Behaviour like this is useful while debugging or previewing a data query where, for example, you are only interested in seeing the first few records.

The execution plan produced by the optimiser can vary depending on the optimiser's goal. Optimising for best throughput is more likely to result in a full table scan rather than an index scan, or a sort-merge join rather than a nested loops join. Optimising for best response time, however, more likely results in an index scan or a nested loops join.

You can select from the following:

- **Oracle default** — The report will be executed according to the optimisation method defined for the database. In most cases this is the Cost based optimisation method.
- **All rows** — The report will be executed according to the Cost based optimisation method as recommended by SimCorp.
- **Rule** — The report will be executed according to the Rule based optimisation method.

In addition, you can select **Hint** to create an optimizer hint that alters the execution plan.

2. In the **SQL condition** pane, write the SQL statement. These statements can be simple or complex.

   You must use the SQL names from tables and fields that have been defined in the extraction definition. The conditions that you create here are appended to the WHERE clause of the current select statement.

   To see the resulting SQL query, open **Functions** > **Execution Plan**, as described in View preview data, SQL query and execution plan.

   To validate the SQL query, select **Functions** > **Verify Query**.

Note

Plain text conditions are not maintained during upgrades of the system.

Define variables in conditions

The conditions that you create on the **Source** > **Extra** tab can be included in the **Variables** tab together with the normal defined variables. This is done by adding a SQL comment after the condition, which must consist of these three parts:

- VARIABLE — fixed text to identify the comment as variable specification (not case sensitive)
- TYPE — defines the data type for the pop-up and validation (can be TEXT, NUMBER or DATE)
- Caption — any free text to specify the caption to be displayed in the dialog box

The comments are ignored if the value type and type specified in comments are not the same.

Note

The date format is: TO_DATE('20010101','YYYYMMDD').

2.5.4 Add variables for executing the extraction definition

In certain cases you might want segments or conditions to be variables when you are executing the extraction definition. This enables you to execute numerous data extractions by adjusting varying conditions. For example, you can execute data extractions for different dates.

You cannot create a variable. Instead, when you create a segment or condition, SimCorp Dimension defines it as a possible variable. (See Specify the records to export by using segments and Refine the data extraction by using conditions for information about creating segments and conditions).

- On the **Source** > **Variables** tab, select the **Is variable** check box to apply the appropriate segments and conditions as variables.

The text in the **Variable name** field is the text that appears in the prompt when executing the extraction definition. You can provide a more descriptive name for the variable. Otherwise, SimCorp Dimension uses the database caption of the field in the segment or condition.

You can define multiple variables, for example in different inline views, with the same **Variable name**. When execute the data extraction, you only have to specify the variable with the same name once. The variables with the same name must have the same data type (and the same default values).

When an extraction definition with a variable is executed, you will be prompted to define the parameters of the variable conditions.

The default values for the parameters are the values defined in the segment or condition. These values are used when executing the data extraction in a batch job.

2.5.5 Group and present data in a sorted order

If you are exporting your data to a flat file structure or an XML file, you can group and present the data in a particular order.

Note

Sorting is only relevant for flat file structures and XML files. Data is not necessarily sorted when extracted to a database table. See Export data to flat file structure and Export data to XML file structure.

1. On the **Source** > **Sorts** tab, in the **Field alias** column, select the fields that you want to organize. This will group the export data according to these fields.
2. In the **Sorting direction** column, select whether the data should be arranged in ascending order or descending order.
3. If you are sorting more than one field, assign the priority for each field. This groups and organizes the data according to the sort priority.
4. Save the extraction definition. This will check for errors and validate the extraction definition.

   You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

2.5.6 Apply predefined formulas

You can apply predefined formulas to the extraction definition.

Note

Formulas cannot be applied to Inline views or Sub-queries.

Prerequisites

Before you can use predefined formulas in your extraction definition, the following must be in place:

1 In the **Extraction Formula Fields** window, define the field names that are used in the formula.
   - A Enter the extraction formula field and extraction formula field name, and select the data type.
   - B Save the definition.

2 In the **Formulas** window, create the predefined formula.
   - A Enter a formula name and header, and set the **Usage** field to **Data Extractor**.
   - B Click **Edit Formula**.

     The **Edit Formula – Formulas** window appears.
   - C In the **Edit Formula – Formulas** window, create the formula by selecting amongst the fields that you defined in the **Extraction Formula Fields** window from the **Possible fields** pane and selecting amongst the **Functions and operators** pane.
   - D Click **OK** and then save the formula.

Set up formulas in extraction definitions

1 In the **Extraction Definitions** window, ensure that all required tables and fields needed for the formula are added to the **Source** > **Tables** and **Source** > **Fields** tabs.

2 On the **Formulas** tab, click **Add**.

The **Extraction formula name** dialogue box appears.

3 In the **Extraction formula name** dialogue box, enter an extraction formula name and then select the formula from the **Formula** drop down list.

4 Select the formula in the **Extraction definition formulas** pane, and then select a source field on the **Field Mappings** tab.

The list of selectable source fields is identical to the fields defined on the **Source** > **Fields** sub-tab.

Note

If required, you can assign a constant value to the formula field, instead of mapping it to a source field.

5 On the **Destination Fields** tab, add the formula in the **Formula** field and enter a user-defined SQL name.

6 Save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

2.5.7 Add conversion rules using export translations

You can add a translation to a data extraction on destination fields with the String data type. Export translations can be used for converting data to a defined format, such as all lower case or to all upper case. The translation must first be created in the **Export Translations** window.

Translations are executed after formulas.

Note

You can change the data type from Integer to String, and then add a translation. In the **Extraction Definitions** window on the **Destination Fields** tab, clear the **Use data dictionary** check box and change the data type from Integer to String.

1. On the **Destination Fields** tab, press F4 in the **Translation ID** field for the desired source field of data type String.

A list of predefined translations appears.

2. Select the required translation and save the extraction definition. This will check for errors and validate the extraction definition.

   You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

2.5.8 Apply predefined business logic

SimCorp provides a set of predefined functions in the form of plug-ins. Plug-ins are developed at the request of customers to add special business logic to extractions.

- To apply the plug-in, select the one from the drop down list on the **Plug-ins** tab.

The **Plug-in information** pane provides you with information about the type of plug-in, and the type of parameter required in the field.

The following describes the functionality of the various plug-ins.

2.5.8.1 Aux. status condition forwarder

The **Aux. status condition forwarder** plug-in can improve the runtime performance of data extractions that are started via the **Auxiliary Job Definitions** window or the **Order Auxiliary Job Definitions** window.

Data extractions that can benefit from this plug-in must match the following requirements:

- The extraction makes use of an inline view
- The inline view contains the table TRANSMAIN
- The inline view is joined with its parent statement using an outer join

Data extractions to be used in connection with auxiliary status jobs are generally formulated, and in most cases they extract data for all transactions in the system. If an auxiliary status job is started, it is because the status of a specific transaction in the system has been changed. If an extraction is started via an auxiliary status job, then the auxiliary status environment instructs the Data Extractor to run a specific extraction. The data extraction obtains filters from the auxiliary status environment, so that only data for the defined transactions are extracted.

The filters obtained from the auxiliary status environment are added to every occurrence of the TRANSMAIN in the master statement of the data extraction. All other instances of this table in any of the inline views are ignored.

This was done because multiple instances of the TRANSMAIN table with different semantics can occur in the same extraction. A practical example would be an extraction that shows transactions and all linked transactions at the same time. To achieve this, you would use two instances of the table in the same extraction.

Use this plug-in to manually apply the filters from the auxiliary status environment to other occurrences of the TRANSMAIN table inside the inline views. This can greatly improve the runtime performance because the inline view would then operate on a strongly reduced number of records.

Parameter 1: Name of the inline view containing an instance of the TRANSMAIN table to be filtered.

Parameter 2: n/a

2.5.8.2 Aux. status feedback

Auxiliary status jobs and order auxiliary status jobs can be used to trigger data extracts. This is usually done in order to extract data concerning transactions or orders. After completion of the extraction, the auxiliary job environment must know whether the extraction was successful or not. Depending on the setup, it might also be necessary to know which transactions or orders have been extracted.

There are many factors (for example, user defined search conditions and user defined links to related tables) that influence the result of an extraction. For example, if the auxiliary job triggers an extraction for a specific transaction or order, it is not guaranteed that the requested transaction or order is contained in the extract.

The **Aux. status feedback** plug-in enables you to generate a feedback to the auxiliary job with information about the extracted transactions. This feedback can be used in the auxiliary job to decide which auxiliary status number the transactions or orders can be raised to.

The extraction definition must contain a source field, which contains the internal key of the transaction or the order being extracted. The plug-in must be configured so that it looks in the correct field for this internal key, in order to generate the feedback.

Parameter 1: Name of the source field containing the internal key for identification of the transaction or the order.

Parameter 2: n/a

2.5.8.3 Child data vector

Use the **Child data vector** plug-in to compress the values from one column of a number of child records into one text field, which is then stored in the parent record as one value.

The compressed values are delimited with a '^' (hat).

For example, an output like this:

Security number	Security	Redemption date
1234	Bond A	01-03-2003
1234	Bond A	01-06-2003
1234	Bond A	01-09-2003
3456	Bond B	01-02-2003

Could be transformed into an output like this:

Security number	Security	Redemption date
1234	Bond A	01-03-2003^01-06-2003^01-09-2003
3456	Bond B	01-02-2003

Parameter 1: Configuration string

Parameter 2: n/a

The parameters are delimited with a ';' (semi colon) and consist of a keyword and a value delimited by a '='.

Use the following keywords to set up the function:

PARENT	Identifies the field in the parent record that holds the parent key data.
CHILD	Identifies the field in the parent record that receives the compressed child data.
DATA	Identifies the field in the child record that holds the data to be compressed.
KEYDATA	Identifies the field in the child record that holds the parent key data.
JOIN	Defines a join between two fields. Multiple joins can be defined to build a reference path.
SORT	List of fields to use for sorting. The sorting can only be done in ascending order.
DELIMIT	Token to be used as a delimiter (optional, default ‘^’). Use ‘\s’ for a semicolon, ‘\c’ for a comma and ‘\t’ for a tab.
QUOTE	Flag to indicate whether the child values have to be en-quoted (optional, default 0).

An example could look like this:

```
parent=MASTER_KEY;child=CHILD_VECTOR;join=dxddefs.dxddefik=dxdflds.dxddefik;join=dxddefs.dxddefik=dxdflds.dxddefik;keydata=dxdflds.dxddefik;data=dxdflds.dxdsqlname
```

Where the extract definition contains two fields, named MASTER_KEY and CHILD_VECTOR. The first field must be a source field containing the parent key data. The second field must be a destination field (varchar or clob), long enough to hold the compressed child data.

2.5.8.4 Data File Import

Use the **Data File Import** plug-in to import data from a file using external user schemas.

The plug-in accepts the following parameters (text):

- FILE — File name of the file to read from (you will be prompted during execution if the file name is not specified). Format of data file:
  - CSV format.
  - First two lines is a header and must start with “//”.
  - First line could contain anything after “//”.
  - Second line, after “//”, contains field headers separated by “;”.
  - Each record of the file must contain the same number of fields as specified in the second line of the header (otherwise, the extraction process is stopped with an error).
- ERR — Specifies if the extraction is stopped when there is an error (1), or if records with errors should be skipped (0).
- NOTNULL — List of the fields that must be checked for null values (a subset of the fields defined in the file header). Records with null values in these columns will be removed during processing.

2.5.8.5 Fetch limit

Use the **Fetch limit** plug-in to fetch data in data chunks. This plug-in interrupts the fetch loop if the limit is reached. When the fetch limit is passed, the records that exceed the fetch limit will be discarded.

This plug-in can be useful when testing large data extractions, where you would like to see only the first couple of records to verify the results.

Parameter 1: Number of records (for example 1000)

Parameter 2: n/a

2.5.8.6 Managed contract event information

Use the **Managed contract event information** plug-in to calculate event information for a managed contract. The events are compressed with values delimited with a '^' (hat) if more than one event is found for the managed contract.

Parameter 1: Configuration string

Parameter 2: n/a

Source fields(mandatory):

HOLKEY	Alias of the source field containing the internal key of the holdings

Optional input:

DATE	User input date to filter events
DELIMIT	Token to be used as a delimiter (default ‘^’). Use ‘\s’ for a semicolon, ‘\c’ for a comma and ‘\t’ for a tab.

Destination fields (at least one must be specified)

CDAT	Alias of the destination field for the result of future cash flow date
CCER	Alias of the destination field for the result of future cash flow certainty
CDIR	Alias of the destination field for the result of future cash flow direction
CVAL	Alias of the destination field for the result of future cash flow value
CCUR	Alias of the destination field for the result of future cash flow currency
CANN	Alias of the destination field for the result of future cash flow annotation
CDEL	Alias of the destination field for the result of future cash flow underlying for delivery
CUN	Alias of the destination field for the result of future cash flow underlying per contract
CFOR	Alias of the destination field for the result of future cash flow formula
DDAT	Alias of the destination field for the result of future decision date
DVAL	Alias of the destination field for the result of future decision value
FDAT	Alias of the destination field for the result of future fixing date
FVAL	Alias of the destination field for the result of future fixing value
PDAT	Alias of the destination field for the result of past event date
PTYP	Alias of the destination field for the result of past event type
PID	Alias of the destination field for the result of past event ID
PDES	Alias of the destination field for the result of past event description
PEFF	Alias of the destination field for the result of past event effect
PANN	Alias of the destination field for the result of past event annotation

2.5.8.7 Residual allocation

When using computer programs to perform floating point math, rounding errors can occur. This can pose a problem if such rounding errors become visible due to amplification or other factors. For example, if all figures should sum up to 100%, the achievable total will often be a figure such as 99.999%. The **Residual allocation** plug-in calculates the residual to 100% and allocates it to the biggest figure in the extract.

Parameter 1: The name of the destination field that contains rounding errors

Parameter 2: Specialised configurations of the plug-in

The parameters in the configuration string are delimited with ';' and consist of a keyword and a value delimited by a '='.

Use at least one of the following keywords to set up the function:

TOTAL	The expected total to which it should be summed up (default 100).
TOTALFLD	The SQL name of a destination field, which contains the values to be used instead of TOTAL.
THRESHOLD	Threshold for the plug-in to accept the residuals. The threshold is given as a percentage of the value the residual is allocated to (default 5).
GROUPING	A comma separated list of grouping fields. If given, the plug-in calculates residuals for each group separately.

2.5.8.8 Simple Calculator

Use the **Simple Calculator** plug-in to calculate values based on the source fields and store the result in a destination field. You must create this destination field before you can use this plug-in.

Parameter 1: SQL name of the destination field. This is the field that will receive the results of the calculation. It must be defined on the **Destination Fields** tab in the **SQL name** field, and where the **Source field** is set to (NONE).

Parameter 2: Formula. For example, (-1)'(-FIELDA/2)+FIELDB#1*2.5

FIELDA and FIELDB represent the SQL names of the source fields that are used in the calculation.

The following operators are allowed in the simple formulas:

+	Addition	3+7 = 10	(Priority 5)
-	Subtraction	5-2 = 3 or (-2*3) = -6	(Priority 5)
*	Multiplication	2*3 = 6	(Priority 4)
/	Division	15/2 = 7.5	(Priority 4)
‘	Power	4'2=16 or 9'(1/2) = 3	(Priority 3)
#	Substitution of NULL values	a#2 = 2	(Priority 2)
(	Opening bracket	3*(4-2) = 6	(Priority 1)
)	Closing bracket	3*(4-2) = 6	(Priority 1)

Source and destination fields must be numeric. Only source fields can be used as a variable in a formula. You cannot refer to the result of another formula in a formula, but you may use brackets. Division by 0 or other illegal operations result in NULL. If NULL (=empty) is included in a calculation, the result will also be NULL.

For more complex calculations use the **Formulas** tab. It includes additional type checking, extended control structures (for example, “if-then-else” statements), and other features for improved reusability.

2.5.8.9 String Generator

Use the **String Generator** plug-in to generate a string, which is then inserted into the specified column for each extracted record. The string can contain tokens such as user name, time stamps, etc.

Parameter 1: Name of the destination field. It must be defined on the **Destination Fields** tab in the **SQL name** field, and where the **Source field** is set to (NONE).

Parameter 2: Configuration string.

The content of the generated string is defined via a configuration string consisting of the following elements:

b	Space
n	Record number
u	User name
t	Time stamp
d	Date stamp

Note

Date stamps and time stamps are set right before the records are written to the destination media. This means that if the data extraction takes a long time, you possibly get various date stamps and time stamps.

2.5.8.10 Time Weighted Index (TWI)

Use the **Time Weighted Index (TWI)** plug-in to calculate aggregations of the time weighted index (TWI) on a group of holdings.

The calculation can be performed in two modes: on a daily basis or accumulated since a specific date. The aggregations are made per date. The plug-in inserts a new row into the data extraction which contains the calculated results for each date. Each inserted record also contains the date information and can be recognized by the record type set in the field DXDRECTYPE in the extract table.

In order to use the plug-in, you must first set up a performance calculation in the **Performance Calculation** window. This performance calculation is used as a basis for the TWI calculations.

The extract definition that uses this plug-in must include at least two source fields: one for delivering the internal key of the performance calculation, and the other one for delivering the date series. Typically, this information comes from the PERFORMIK field in the PERFORM table for the internal key of performance calculation, and from the TODATE field in the PERFORMCALC table for the date information.

Note

Apply your preferred sorting. Note that the top level sorting must be made on the field with the date information. If the data is not read in chronological order from the database, the calculation might produce unreliable results. The sorting is especially important when calculating the accumulated TWI because the start date for the accumulation is set to the TODATE of the first record that was processed.

Note

Use the search conditions on the PERFORM table to make sure that only one performance calculation is selected. If more than one performance calculation is selected, the plug-in might produce unreliable results.

The details for the calculation must be configured with a configuration string which is supplied to the plug-in as a parameter.

Parameter 1: Configuration string

Parameter 2: n/a

The configuration string can contain the following keys:

Source fields (mandatory):

PERFFLD	Alias of the source field containing the internal key of the performance calculation
DATEFLD	Alias of the source field containing the date

Destination fields (at least one must be specified):

QCFLD	Alias of the destination field for the result for the daily TWI in QC
PCFLD	Alias of the destination field for the result for the daily TWI in PC
RCFLD	Alias of the destination field for the result for the daily TWI in RC
ACCQCFLD	Alias of the destination field for the result for the accumulated TWI in QC
ACCPCFLD	Alias of the destination field for the result for the accumulated TWI in PC
ACCRCFLD	Alias of the destination field for the result for the accumulated TWI in RC

Calculation options (optional):

EXCOSTS	Exclude costs (1 or 0, default 0)

EXTAXES	Exclude taxes (1 or 0, default 0)
EXCOUPONS	Exclude coupons (1 or 0, default 0)
EXDIVIDENDS	Exclude dividends (1 or 0, default 0)
INFUNDCOSTS	Include funding costs (1 or 0, default 0)
SHOWPOPUP	Prompt user interactively for the calculation options (1 or 0, default 1)

An example of a configuration string could be:

PERFFLD=PERFORM_PERFORMIK;DATEFLD=PERFORMCALC_TODATE;ACCRCFLD=TWI_RC_ACC;RCFLD=TWI_RC;EXCOSTS=1;EXTAXES=0;EXCOUPONS=1;SHOWPOPUP=0

2.5.8.11 Value at Risk (VaR)

Use the **Value at Risk (VaR)** plug-in to calculate aggregations of the value at risk (VaR) on a group of holdings.

The calculation can be performed in two modes: normal or incremental. The aggregations can be specified by defining grouping levels. The plug-in inserts a new row into the data extract, which contains the calculated results for each group in each grouping level. Each inserted record also contains the information of the group fields and can be recognised by the record type set in the DXDRECTYPE field in the extract table. If incremental value at risk is calculated, a whole set of detail records is saved together with the aggregated record for each grouping level.

In order to use the plug-in, you must first set up a performance calculation that defines a value at risk calculation in the **Performance Calculation** window. This performance calculation is used as a basis for the VaR calculations.

The extract definition that uses this plug-in must include at least three source fields: one for delivering the internal key of the performance calculation, one for delivering the date series, and one for delivering the internal keys of the holdings. Typically, this information comes from the PERFORMIK field in the PERFORM table for the internal key of performance calculation, from the TODATE field in the PERFORMCALC table for the date information, and from the HOLKEYIK field in the PERFORMCALC table for the internal keys of the holdings.

Note

Apply your preferred sorting. Note that the top level sorting must be made on the field with the date information. If the data is not read in chronological order from the database, the calculation might produce unreliable results.

Note

Use the search conditions on the PERFORM table to make sure that only one performance calculation is selected. If more than one performance calculation is selected, the plug-in might produce unreliable results.

All the details for the calculation must be configured with a configuration string that is supplied to the plug-in as a parameter.

Parameter 1: Configuration string

Parameter 2: n/a

The configuration string can contain the following keys:

Source fields (mandatory):

PERFFLD	Alias of the source field containing the internal key of the performance calculation
DATEFLD	Alias of the source field containing the date
HOLKEYFLD	Alias of the source field containing the internal key of the holdings
GRPFLDS	List of source field aliases to specify the grouping levels

Destination fields (at least one must be specified):

VARQCFLD	Alias of the destination field for the result for the VaR in QC
VARPCFLD	Alias of the destination field for the result for the VaR in PC
VARRCFLD	Alias of the destination field for the result for the VaR in RC
INCRVARQCFLD	Alias of the destination field for the result for the incremental VaR in QC
INCRVARPCFLD	Alias of the destination field for the result for the incremental VaR in PC
INCRVARRCFLD	Alias of the destination field for the result for the incremental VaR in RC
DETAILFLD	Alias of the destination field marking the inserted records as detail or aggregation

An example of a configuration string would be:

PERFFLD=PDEF_PERFORMIK;GRPFLDS=PRES_TODATE,PG_PORGRP,P_POR,S_INSTYPE,QC_CUR;VARQCFLD=VAR_QC;VARRCFLD=VAR_RC;INCRVARRCFLD=VAR_RC_INCR;DATEFLD=PRES_TODATE;HOLKEYFLD=HK_HOLKEYIK;DETAILFD=ISDETAIL

The grouping fields define which aggregations must be calculated. Five grouping levels are in the example:

PRES_TODATE,PG_PORGRP,P_POR,S_INSTYPE,QC_CUR

The plug-in groups the data by all of these fields and makes an aggregation for each group that is found. In a second iteration the lowest grouping level QC_CUR is dismissed. The plug-in starts grouping with the remaining four group fields:

PRES_TODATE,PG_PORGRP,P_POR,S_INSTYPE

New aggregations are calculated for each group. Afterwards, the lowest grouping level is dismissed and the calculations start again. This is done repeatedly until all aggregations for all grouping levels have been calculated.

When you decide to use the extracted data, you will have to identify the aggregations of each grouping level. To do this, use the DXDRECTYPE field in the data extract. All records that were read from the database contain the number 1. All aggregated records contain a number above 101, where 101 represents the highest grouping level, 102 represents the second highest level, 103 the third highest level, etc.

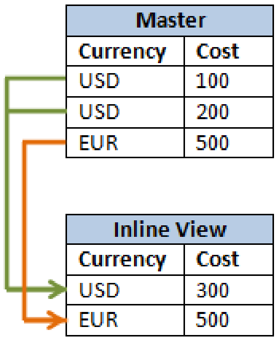
When calculating incremental value at risk, a new set of detail records is saved together with each aggregation. This can make it hard to interpret the output of the data extract. To simplify matters, you can optionally add an additional destination field and use the **DETAILFLD** parameter on it. The plug-in will then set a Boolean flag into the generated records, which indicates whether the record is a detail record or an aggregated record.

2.5.9 Create sub-statements to the master: Inline views

An inline view aggregates data before handling it.

For example, in an extraction definition where the MASTER statement extracts costs associated with various currency, you can create an inline view that aggregates the data.

As the example shows, the two costs denominated in USD are aggregated in the inline view. With an inline view, it is possible to retrieve new types of information, such as determining the weight of each cost within its respective value. Thus, the USD100 costs out of the total USD300 can now be calculated as 33 percent of the USD costs. This calculation is not possible without aggregating the data with an inline view.



Master	
Currency	Cost
USD	100
USD	200
EUR	500

Inline View	
Currency	Cost
USD	300
EUR	500

Use inline views for the following:

- Grouping of sub-totals and grand-totals
- Count number (for example, count number of transactions per Portfolio group)
- Sum (for example, sum payment amount per Security ID)

- Maximum/minimum values (for example, maximum dates for interest adjustments)

To create an inline view:

1 In the **Extraction Definitions** window, in the **Statement hierarchy** pane, click **New**.

  You are prompted to define the statement.

  A Enter a name, and in the **Type** field select **Inline view**.

  B In the **Parent** field, select the statement to which this is the child.

  You can create a sub-statement to the MASTER statement or to another sub-statement.

  C Select the **Distinct** check box to obtain a single record as result, and click **OK**.

2 Add the required tables and fields to the inline view sub-statement.

  When selecting the fields for the inline view sub-statement, make sure that you select at minimum a field for aggregated value and a field to join the sub-statement with its parent.

Note

When you are configuring sub-statements, the only relevant tab is the **Source** tab. Even though the **Destination Fields**, **Formulas**, **Plug-ins** and **Settings** tabs are enabled, they work only for the MASTER statement.

  A In the **Statement hierarchy** pane, select the new sub-statement.

  B Select the tables or views from which to extract data by selecting an available field in the **Table** column on the **Source** > **Tables** tab and pressing F4.

  A list of all available tables and views appears. Select the required tables and views from the list and click **OK**.

  C Select the required fields from those tables that have been added to the sub-statement. There are two ways to add fields:

  - Automatic — Use the **Data Dictionary Browser**
  - Manual — Use the **Source** > **Fields** tab

  D Apply the required aggregation functions (**Aggr. Function**) to appropriate fields.

3 If required, limit the inline view by adding **Segments** or **Conditions** (see Specify the records to export by using segments and Refine the data extraction by using conditions).

4 Add the fields from the inline view sub-statement to the MASTER statement.

  A In the **Statement hierarchy** pane, select the MASTER statement.

  B On the **Source** > **Fields** tab, in the **Table alias** field, select the inline view sub-statement, and in the **Field** field, select the accompanying fields from the inline view sub-statement.

5 Link the MASTER statement with the inline view sub-statement.

A On the **Source** > **Joins** tab, select the fields that are common in both the MASTER statement and in the inline view sub-statement.

B If required, make a left outer join between the MASTER and the sub-statement.

To make a left outer join, select the **Outer join** check box. The result of a left outer join for table A and B always contains all records of the **From field** (left table (A)), even if the join-condition does not find any matching record in the **To field** (right table (B)).

6 Add the inline view sub-statement to the **Destination Fields** tab by selecting it in the **Source field** field.

7 Save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**.

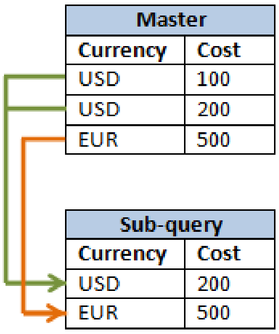
You can also see the inline view sub-statement in the FROM clause of the SQL query. Select **Functions** > **Execution Plan**. See View preview data, SQL query and execution plan for more information.

2.5.10 Create sub-statements to the master: Sub-queries

A sub-query is a condition that runs on the MASTER statement.

For example, in an extraction definition where the MASTER statement extracts costs associated with various currency, you can create a sub-query that extracts the data according to a certain condition, such as only extracting the highest costs in each currency.

As the example shows, the sub-query returns only the highest costs in each currency; in this case it returns USD200 and EUR500.



Master	
Currency	Cost
USD	100
USD	200
EUR	500

Sub-query	
Currency	Cost
USD	200
EUR	500

For information on how to avoid drain on system performance, refer to the SimCorp Dimension ***System Performance Reference Manual***.

To create a sub-query:

1 In the **Extraction Definitions** window in the **Statement hierarchy** pane, click **New**.

You are prompted to define the statement.

A Enter a name, and in the **Type** field select **Sub-query**.

B In the **Parent** field, select the statement to which this is the child.

You can create a sub-statement to the MASTER statement or to another sub-statement.

C Select the **Distinct** check box to obtain a single record as result (recommended), and click **OK**.

2 Add the required tables and fields to the sub-query sub-statement.

When specifying the sub-query sub-statement, make sure that you select only one field on the **Fields** sub-tab.

You can add conditions to the sub-query, but only one field should be added on the **Fields** sub-tab for joining.

Note

When you are configuring sub-statements, the only relevant tab is the **Source** tab. Even though the **Destination Fields**, **Formulas**, **Plug-ins** and **Settings** tabs are enabled, they work only for the MASTER statement.

A In the **Statement hierarchy** pane, select the new sub-statement.

B Select the tables or views from which to extract data by selecting an available field in the **Table** column on the **Source** > **Tables** tab and pressing F4.

A list of all available tables and views appears. Select the required tables and views from the list and click **OK**.

C Select the required field from those tables that have been added to the sub-statement. There are two ways to add the field:

- Automatic — Use the **Data Dictionary Browser**
- Manual — Use the **Source** > **Fields** tab

D Apply the required aggregation function (**Aggr. Function**) to the field.

3 In the **Segments** or **Conditions** tabs, add specific segments or conditions to the sub-query (see Specify the records to export by using segments and Refine the data extraction by using conditions

4 Link the MASTER statement with the sub-query sub-statement.

A In the **Statement hierarchy** pane, select the MASTER statement.

B On the **Source** > **Joins** tab, select the field in the MASTER statement that is common with the sub-query sub-statement.

5 Save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**.

You can also see the sub-query sub-statement in the WHERE clause of the SQL query. Select **Functions** > **Execution Plan**. See View preview data, SQL query and execution plan for more information.

2.5.11 Transform data by using SQL expressions

You can specify SQL expressions to use Oracle® functions for data transformation. The data extractor runs the functions on the database layer. For simple transformations, use the functionality as an alternative to SimCorp Dimension formulas to run a fast data insert (Direct Data Copy), which is not possible with formulas.

You can use the **SQL Expression Builder** to more easily create and edit valid SQL expressions in SimCorp Dimension. The **SQL Expression Builder** is available from the following tabs in **Extraction Definitions**.

- **Source** > **Fields** — Right-click the **SQL expression** field and select **Edit SQL expression**.

  The **SQL expression** field is available if you do not specify the **Table alias** and **Field** fields. This means that you can specify a source field (in the **Table alias** and **Field** fields), or an SQL expression (in the **SQL Expression** field).
- **Source** > **Extra** — Right the **SQL condition** field and select **Edit SQL expression**.
- **Data Quality** — Right-click the **Rule conditions** field and select **Edit SQL expression**.

Open the **SQL Expression Builder** to view the available SQL fields, functions and expressions that you can use when creating an SQL expression. For Data Quality rules, the list of SQL fields includes all destination field aliases.

Note

When you build an **SQL Expression** on a field(s) from a **Report View**, make sure to select the check box **Full view** for the table (s) in the **Extract Definitions** window on the **Source** – **Tables** tab. Otherwise the field(s) will not be shown, because the reduced list of fields will not include the field(s).

However, the used field(s) will also be included if added to the list of fields on the **Source** – **Fields** tab.

The SQL expressions are validated against an Oracle check. For more information, see Oracle reference documentation on SQL expressions.

To transform data by using SQL expressions:

1. Open the **SQL Expression Builder** in the **Extraction Definitions** window.
2. In the **SQL Expression Builder**, double-click a SQL field, function or expression to insert it in the **SQL text** field at the cursor position.

   You can also type the SQL expression directly in the **SQL text** field.
3. To check the SQL expression syntax, click **Check**. The SQL expressions are also checked when you save the extraction definition.
4. Click **OK**.
5. For SQL expressions on the **Source** > **Fields** tab:
   A SimCorp Dimension automatically populates the **Field alias** field for the SQL expression. You can change the field alias.

B You can specify the **Data type**, **Field length**, **Decimals** and **Aggregation function** fields for the fields that are defined by an SQL expression.

C Optional: Enter a description of the SQL expression in the **Description** field.

6 Save the extraction definition. This will check for errors and validate the extraction definition.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

2.5.12 Validate extract data by using Data Quality Rules

You can validate the extract data by using Data Quality Rules specified in the **Extraction Definitions** window.

Related topics

- Data Quality Rules explained below
- Create Data Quality Rules on the next page
- View errors and warnings from Data Quality Rules on page 55

2.5.12.1 Data Quality Rules explained

You can control and validate the data to be extracted by using a set of data quality rules specified in the **Extraction Definitions** window.

You can validate the data to be extracted against user-defined business rules, and then browse the error log afterwards.

For example, use the data quality rules when you are extracting data for a downstream application such as a data warehouse, and you want to validate the data as part of the extraction process.

A data quality rule is a SQL condition that defines the data in a given data set. The data quality rules describe the properties of the data in order for it to be valid.

You specify the data quality rules in the **Extraction Definitions** window > **Data Quality** tab. The data quality rules are applied to the extracted data before it is loaded to a target destination. This means that data quality rule conditions apply to the destination fields.

The data extract is executed in a number of steps:

1. Extract data.
2. Execute the data quality rules. Exceptions to the rules are written to the DXDRULESLOG table.
3. Load (when you are exporting to a Data Warehouse schema).

For details, seeCreate Data Quality Rules on the next pageandView errors and warnings from Data Quality Rules on page 55.

2.5.12.2 Create Data Quality Rules

To create data quality rules in the **Extraction Definitions** window:

1 Open the window and select a setup, or create a new setup.

2 On the **Data Quality** tab, specify data quality rules according to the following:

A In the **Data quality rules** section, specify the data quality rule by using the following fields:

- **Rule order** — Data quality rule execution order.
- **Rule name** — ID of the data quality rule.
- **Rule severity** — Specifies if the data quality rule is handled as an **Error** or **Warning**.
- **Error category** — Error category of the data quality rule when it fails. To specify an error category, open the **Rejection Categories** window or right-click the field and select **Edit Table**.
- **Disabled** — Select the check box to disable the data quality rule. Disabled data quality rules are not used for the data validation.
- **Description** — Optionally describe the data quality rule.

B In the **Rule conditions** section, specify the condition for the data quality rule.

Specify a SQL expression that uses SQL names from the **Destination Fields** tab.

Specify the rule conditions for a given data set. The following condition types are supported:

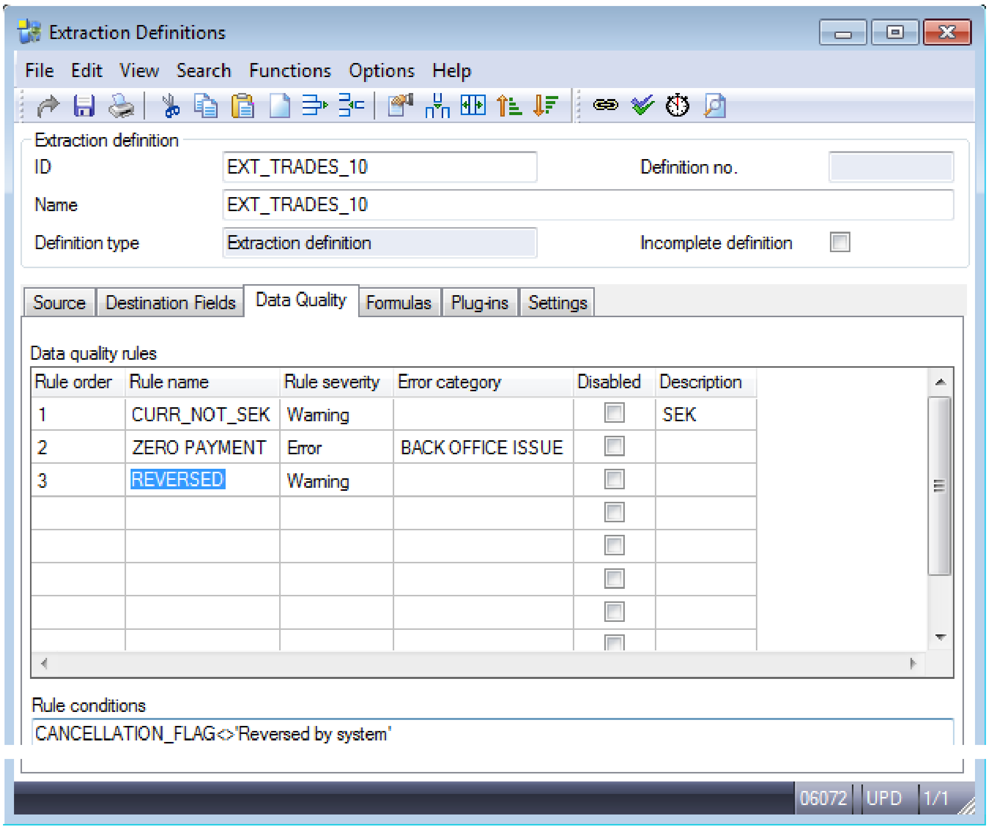
- <field>{ ‘=’ ‘>’ ‘<’ ‘<>’ ‘<=’ ‘>=’ ‘Like’ ‘Not like’ } <value>
- <field>{ ‘Between’ ‘Not between’ ‘In’ ‘Not in’ } <value>[‘;’< value >]
- <field>{ ‘Is Null’ ‘Not null’ }

Note

You can use CASE WHEN SQL Expressions to specify a condition similar to <field>{<operations>}<field>. For example, specify CASE WHEN <field><operation><field> then 1 ELSE 0 END on the **Source** tab.

3 Save.

For example:



Extraction definition

ID	EXT_TRADES_10	Definition no.	
Name	EXT_TRADES_10		
Definition type	Extraction definition	Incomplete definition	□

Data quality rules

Rule order	Rule name	Rule severity	Error category	Disabled	Description
1	CURR_NOT_SEK	Warning		□	SEK
2	ZERO PAYMENT	Error	BACK OFFICE ISSUE	□	
3	REVERSED	Warning		□	
				□	
				□	
				□	
				□	

Rule conditions

CANCELLATION_FLAG<>'Reversed by system'

2.5.12.3 View errors and warnings from Data Quality Rules

You can view information about the failed records in the popup window shown after the data extraction is complete. You can view the number of error and warning rules that have failed and the number of records that are affected.

You can also view the failed records in the following windows:

- **Database User Schema Contents** — Right-click an object and select **View Extracts**.
- **Extract Tables** — Select **Functions** > **View Extracts**.
- **Extraction Setups** — Select **Functions** > **View Extracts**.
- **Extracts Viewer** — View the data quality rules in the **Rules** and **Rule condition** sections.

  The tree view contains an **All records** root node, and the failed rules are shown as sub-nodes.

2.6 Combine multiple queries using Union

You can create complex data extracts in the **Extraction Definitions** window by combining inline views in the MASTER statement.

See Create sub-statements to the master: Inline views for more information.

You can use Union in extraction definitions to merge data from different extracts into the same extract table.

You can, for example, create multiple source definitions in the **Extraction Definitions** window and then combine these to one destination definition by using the **Union** and **Union All** operators.

- Extract data from two different tables and then place the data in the same extract table.
- Extract data to different extract tables, and then create a third extraction definition that selects data from the extract tables and places the data in another extract table.

You can, for example, use the functionality to export the Extraction Setups and Extraction Definitions that are specified, to an extract table in SimCorp Dimension.

Another example is that data is available in different tables in the SimCorp Dimension database depending on the type of security (bonds, equities etc). You can use the union functionality to extract data from the different tables and place the data in the same extract table.

Note

Using the **Union** and **Union All** operators may affect the system performance.

The difference between the **Union** and **Union All** operators:

- Union — Remove duplicate records (records included in both views/extracts).
- Union All — Include duplicate records (records included in both views/extracts).

Note

Only records that are duplicated in both views are removed. Records that are duplicated in one view is not removed.

To create a Union of two inline views:

1. In the **Extraction Definitions** window, add an inline view to the MASTER statement.
   - Add the source definition in the **Source** tab > **Tables** sub-tab and in the **Source** tab > **Fields** sub-tab.
2. In the **Extraction Definitions** window, add a second inline view to the MASTER statement.
   - Add the source definition to the second inline view.

Note

The second inline view can be based on other tables, joins etc.

3 Create a Union of the results from the two inline views.

To create a Union of the results from the two inline views, the settings in the **Source** > **Fields** sub-tab for the two inline views must be compatible.

This means that the same number of fields must be included in both inline views, and that the **Source** > **Fields** > **Data type** (for example, Integer, String, etc) for each field must be the same for both inline views.

You map the fields in the inline views in **Source** > **Fields** > **Field order**.

Note

You can add different tables and fields to the two inline views, but the data type for field order 1 in inline view 1 must be identical to the data type for field order 1 in inline view 2.

To use the **Destination Fields** tab to change the data type for an extracted field, the Union must be made on the data extraction level and not on the inline view level. That is, two extraction definitions must be created, and the **Destination Fields** > **Data type** of each field must be identical to create a Union.

A Select the MASTER statement and then click **Edit**.

B In the **Statement** popup window, in the **Operator** field, select **Union** or **Union All**. This creates a Union of the defined inline views.

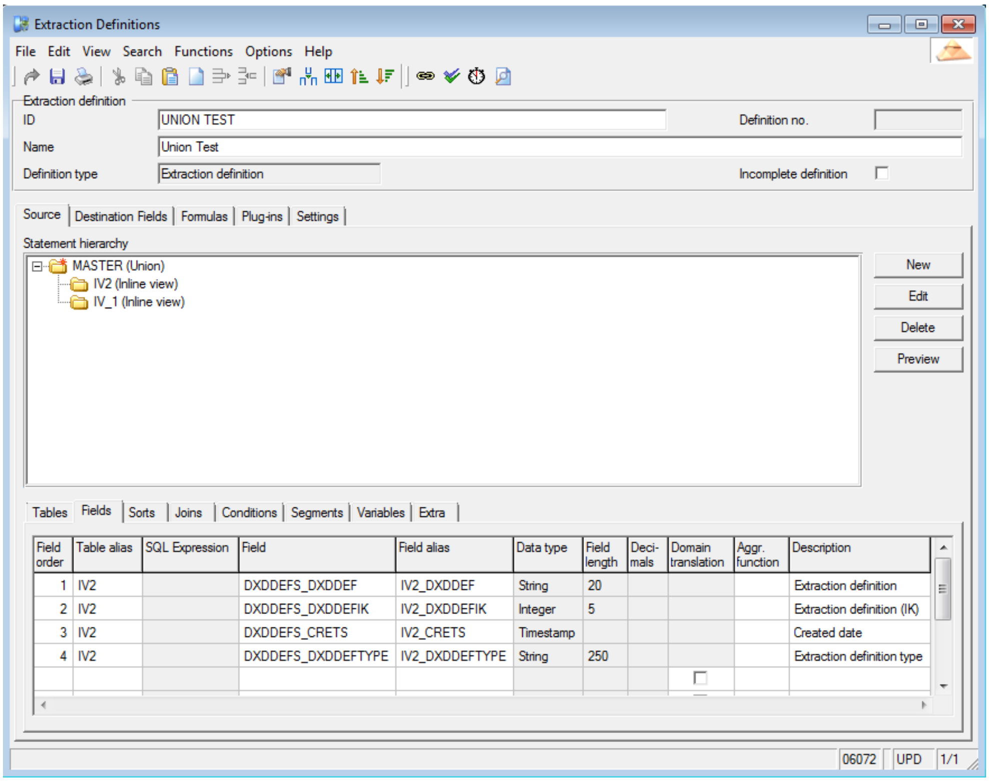
You can now select fields from the first inline view added to the extraction definition. Add the relevant fields and set the description to match the Union result of the two inline views.

The fields selected in the **Source** > **Fields** tab are available in the **Destination Fields** tab. The result will include data as a Union of the two separate inline views.

Note

If you define a Union for inline views that do not match, an error is reported when the extraction definition is validated. For example, if the data type of the source fields differ, or if the inline views include a different number of fields.

For example:


Extraction Definitions
File Edit View Search Functions Options Help
Source | Destination Fields | Formulas | Plug-ins | Settings
New
Edit
Delete
Preview
Tables | Fields | Sorts | Joins | Conditions | Segments | Variables | Extra

Extraction definition

ID: UNION TEST

Definition no.:

Name: Union Test

Definition type: Extraction definition

Incomplete definition □

Statement hierarchy

- MASTER (Union)
  - IV2 (Inline view)
  - IV_1 (Inline view)

Field order	Table alias	SQL Expression	Field	Field alias	Data type	Field length	Deci-mals	Domain translation	Aggr. function	Description
1	IV2		DXDDEFS_DXDDEF	IV2_DXDDEF	String	20				Extraction definition
2	IV2		DXDDEFS_DXDDEFIK	IV2_DXDDEFIK	Integer	5				Extraction definition (IK)
3	IV2		DXDDEFS_CRETS	IV2_CRETS	Timestamp					Created date
4	IV2		DXDDEFS_DXDDEFTYPE	IV2_DXDDEFTYPE	String	250				Extraction definition type
								□		

06072 | UPD | 1/1

2.7 Adjust general settings for data extraction

The **Settings** tab in the **Extraction Definition** window defines how data should be stored in the database for the extraction definition.

- In the **Storage** section, select one of the following:
  - **Overwrite** – The data extraction writes over an existing data extraction
  - **Append** – The data extraction appends to the end of an existing data extraction
- In the **Option** section, select one of the following to specify whether the generated query statement should use bind variables or literals:
  - Select **Use bind variables** when the statement is run often or the generated query is small. This provides the best performance.
  - Select **Use literals** when the query is run rarely, takes a long time, and/or data in the predicate column are unevenly distributed.
- In the **Authorisation** section, view the level of authorisation that is applied to the tables in the extraction definition. See Authorisation in Extraction Definitions for more information.

2.8 Authorisation in Extraction Definitions

You can configure the level of authorisation that is applied to the tables included in the data extractions.

You can configure the default level of authorisation (for all extraction definitions) and then configure the level of authorisation for a specific extraction definition, if you have been granted update access in the **Tasks and Commands** window.

To configure the default level of authorisation:

- Open **Miscellaneous Options** > **Data Extractor** and specify the level of authorisation. The **Partial** authorisation level is the default option.

To configure the level of authorisation for a specific extraction definition:

- Open **Extraction Definitions** and select **Functions** > **Set Authorisation**. Specify the level of authorisation in the popup window. The selected authorisation level is shown on the **Settings** tab.

The following levels of authorisation are available.

Authorisation level	Description
Full	Only the supported tables to which authorisation is applied can be used: FUNDS, CALCFIGURES, CALCFIGUREDETAILS, CALCFIGURESUB, CALCPFCCOPIES, GLHOLKEYS, CALCFIGSUBDATA, PFCHOLDINGS, PFCKEYRATIOS, FUNDEVENTSSUB, FUNDEVGLHOLDINGS, CALCEVENTS The list of supported tables also includes all tables that are affected by the General Authorisation and User Specific Authorisation functionality.
Partial	Access to all tables. Authorisation is applied to the supported tables only.
No authorisation	No authorisation is applied.
Default	The default as specified in the **Miscellaneous Options** window.

Note

When you import an extraction definition with **Default** as authorisation level, the default option is read from the setting in **Miscellaneous Options** > **Data Extractor** in the target SimCorp Dimension installation.

Difference between the authorisation modes

The authorisation modes are applied to the tables that are used in extraction definitions (not to windows and functions which are controlled in the **Tasks and Commands** window).

The different modes specify if and how the authorisation conditions are applied to the tables in an extraction definition (for the tables that can be used).

Authorisation mode	Authorisation conditions applied	Tables allowed
Full	For all supported tables	Only from a supported list
Partial	For all supported tables	All
No	None	All

This means that it is not possible to use an unsupported table in an extraction definition with full as the authorisation mode. All the supported tables have an authorisation condition specified.

The partial authorisation mode applies the authorisation conditions to the supported tables, but all the other tables can be used without any authorisation conditions.

The no authorisation mode makes it possible to use any table and no authorisation conditions are applied. This authorisation mode is useful in batch jobs and for customers without the need for authorisation. The authorisation mode also improves the system performance of the data extraction.

2.9 Data extracts from standby database explained

You can run Data Extractor queries against a standby database, instead of against the production database, to mitigate system load on the production database.

Active Data Guard allows you to have read-only access to a standby database, and the changes from the production database are applied continuously. For more information about Oracle Active Data Guard, see Oracle product documentation.

This makes it possible to have a source located on a standby database and a destination located on the production database.

To execute Data Extractor queries against a standby database, you specify rules in the **Configure Reports and Data Extracts Execution To Standby Database** window. The rules defines for which data extracts and time interval for when the extracts are executed against the standby database.

Use the window to:

- Specify the order in which the rules are sorted.
- Specify the Extraction Definition or Extraction Setup that is included in a rule.
- Specify the time interval for when the rule is applied.
- Specify the standby database timeout for an Extraction Definition or Extraction Setup that is included in a rule.

For details, see:

- Enable standby database in Data Extractor below
- Specify rules for executing data extracts on standby database below

2.9.1 Enable standby database in Data Extractor

To enable Active Data Guard for data extractions:

1. Open the **System Environment Configuration** window.
2. On the **Standby DB** tab, select the **Use standby database for data extraction** check box.
3. In the **Standby DB configuration** section, specify either:
   - **Database name** — enter a TNS name for the standby database.
   - **Connection string** — enter a connection string for the standby database.
4. Click **OK**.

To specify the how the data extracts are executed on a standby database, see Specify rules for executing data extracts on standby database below.

2.9.2 Specify rules for executing data extracts on standby database

To specify rules for executing data extracts on a standby database:

1. Open the **Configure Reports and Data Extracts Execution To Standby Database** window.
2. Specify the sort order for the rule in the **Sort order** field.
3. Select **Extract** in the **Type** field.
4. Specify either an Extraction Definition or an Extraction Setup in the **Extraction definition** or **Extraction setup** fields.
5. Optional: Specify the **User** or **User group** for the rule.
6. Specify the time interval for when to execute the data extract against the standby database in the **From (hh:mm)** and **To (hh:mm)** fields.
7. Specify the timeout in seconds for the standby database in the **Standby DB timeout (sec)** field. The timeout specifies the number of seconds that can pass before the standby database is synchronised with the production database. Otherwise, the data extracts are executed against the production database.

   The default timeout is 0, if no rules are defined.
8. Select the **Always use primary DB** check box to always execute the data extract on the production database.
9. Save.

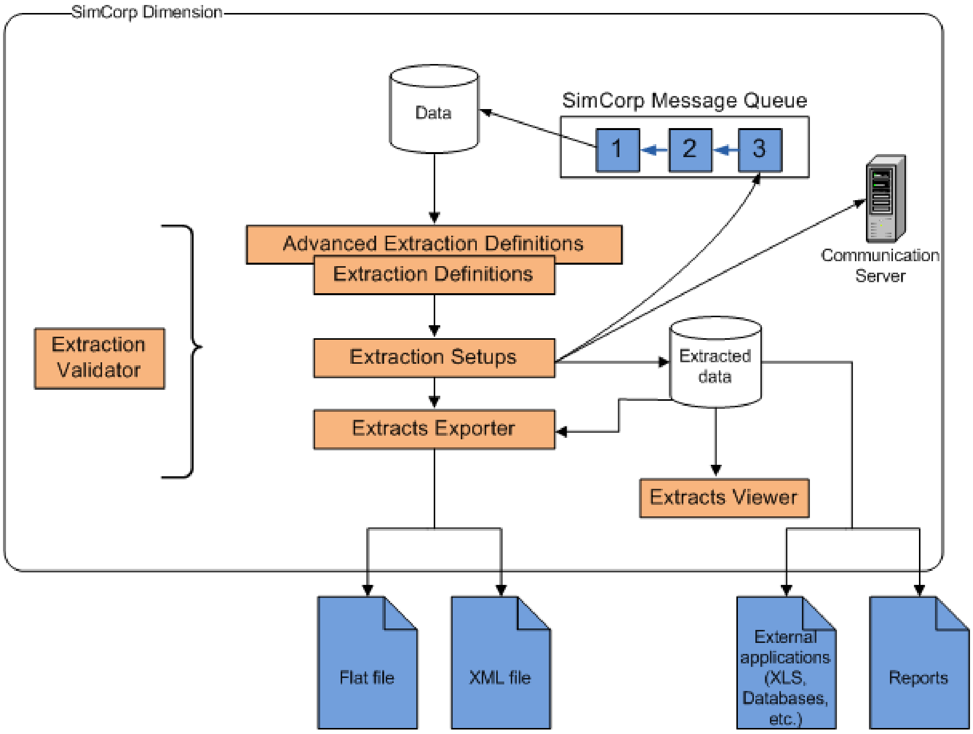
Select **Functions** > **Show Exceptions Log** to view the logs that are related to the currently selected rule.

3 Processing the data extracts

In order to make a data extraction from SimCorp Dimension, you must first define the data source and the data destination. This is done in the **Extraction Definitions** window, and described in Defining and editing data extractions.

Secondly, you must define where the data is physically stored in the Oracle® database, which is done in the **Extraction Setups** window. The main functionality of the extraction setup is to connect the extraction definitions to:

- The **Extracts Exporter** window for flat file and XML file output
- The SimCorp Message Queue for internal updates of data
- The Communication Server for controlled use in multiple systems
- **Database User Schemas** for use in external applications and reports.



After you have defined the data to extract, and edited it as required, as described in Defining and editing data extractions, you can extract the data to the following formats:

- Flat file – See Export data to flat file structure
- XML file – See Export data to XML file structure
- Message queue – See Extract data to the message queue
- Communication Server – See Extract data to the communication server
- External application – See Extract data to an external application

- Report – See Extract data to a report
- Data warehouse schema – See Extract data to a data warehouse schema

3.1 Authorisation structure with extraction setups

There are three ways to control access to data extracts:

- Normal SimCorp Dimension authorisation when extracting data

  You are only allowed to extract the information that your user profile is allowed to view. For information about normal authorisation, refer to the ***System Access*** user manual.
- Authorisation on extracted data from within SimCorp Dimension

  You define access rights based on the settings in the **Extraction Setup** window, where you set read and write authorisation, allowing you to view and/or overwrite extracted data.
- Authorisation when using external user schemas

  The authorisation is defined by the Oracle authorisation setup.

3.2 Export data to flat file structure

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract for use in a spreadsheet or in another system. To do this, you can export the data to a flat file structure.

Before you can export the data to a flat file, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, follow the workflow to export the data to a flat file.

Process	See
Define the naming conventions for the external file names	Prerequisite: Create reference file
Create an extraction setup	Create extraction setup
Create extracts exporter definition	Create extracts exporter
Select plug-ins for the extracts exporter definition	Format data extractions

3.2.1 Prerequisite: Create reference file

To export data to a flat file, you use the **Extracts Exporter Definition** window. The **Extracts Exporter Definition** window requires a reference file, where you define the reference to external files and define naming conventions for the external file names. You can create the reference file before you start the process of exporting the data.

Generate export file names based on extracts exporter definition ID

You can generate export files in the **Extracts Exporter Definitions** window that use the ID of the extracts exporter definition as the file name.

- Open **DATA** > **REFERENCE DATA** > **Miscellaneous Codes** > **Reference Files** and select the **SimCorp Dimension config** component type. Then enter `EXPORTERID` as the component text.

You use the reference file in the **Extracts Exporter Definitions** window. The name of the export files is the ID of the extracts exporter definition.

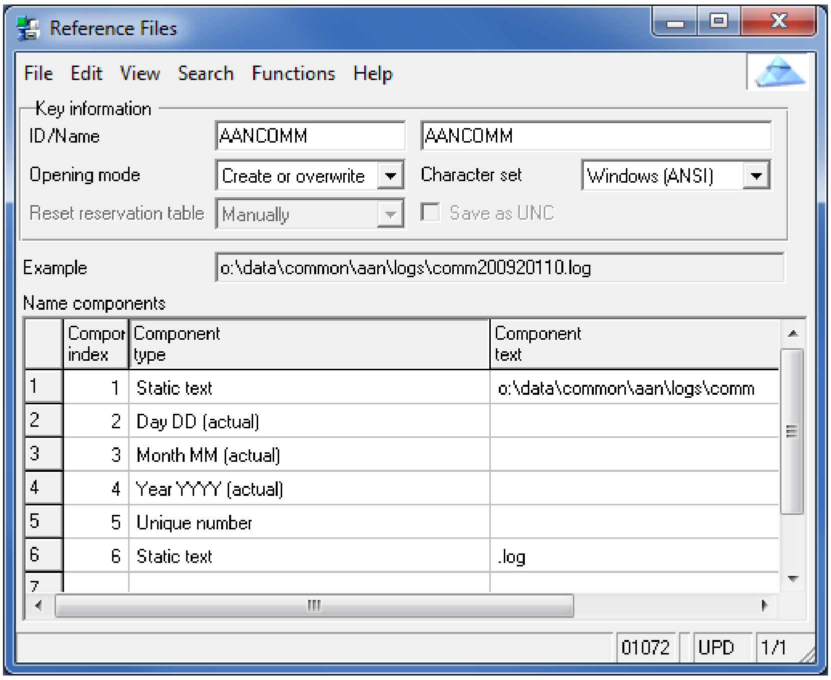
To create a reference file definition

1. Open the **DATA** > **REFERENCE DATA** > **Miscellaneous Codes** > **Reference Files** window, and enter an ID and name.
2. In the **Opening mode** field, select how the external file should be accessed.
   - **Append** — This option means that the new file is appended to an already existing file. If the file does not already exist, it is created.
   - **Create or overwrite** — This option means that a new file will be created. If it already exists, the content of the file will be overwritten with the new file.
   - **Create, exclusively** — This option means that a check will be performed to ensure that the file does not already exist. If it exists, you will receive a warning and the writing to the file is terminated.

     You must then select an option from the **Reset reservation table** field. This field is used in conjunction with the **Create, exclusively** opening mode and the **Unique number** or **Unique alpha** component type.

     The **Reset reservation table** field defines how often the unique number or unique alpha component of the filename is to be reset by the system. You can select: **Daily**, **Monthly**, **Yearly** or **Manually**. Manual resetting of the components is performed by selecting **Functions** > **Reset Reservation Table**.
   - **Read only** — This option is used when the file is used for read only purposes.
   - **Read only/Scan for files** — This option is used when you want to ensure that each file is only read once. By default, the **Save as UNC** check box is selected. This converts the file name to a UNC before it is saved.
3. Select the character set for the external file.
4. In the **Component type** field, select components of the file name structure that you are creating.
5. Save.

Example



Key information

ID/Name AANCOMM AANCOMM

Opening mode Create or overwrite

Character set Windows (ANSI)

Reset reservation table Manually

☐ Save as UNC

Example o:\data\common\aan\logs\comm200920110.log

Name components

	Compor index	Component type	Component text
1	1	Static text	o:\data\common\aan\logs\comm
2	2	Day DD (actual)	
3	3	Month MM (actual)	
4	4	Year YYYY (actual)	
5	5	Unique number	
6	6	Static text	.log
7			

01072 UPD 1/1

In this example, **Static text** provides the path where the extract will be saved. This is followed by file naming convention of **Day DD**, **Month MM**, and **Year YYYY**. Then the naming convention is followed by a **Unique number** to avoid a file being overwritten. This example then concludes the file convention with the file type (.log).

3.2.2 Create extraction setup

Use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the various destinations, such as user schemas, message queues, or communication servers.

The extraction setup can also be used on extracts exporter definitions. When you export data to a flat file, you connect the extraction definition with the extracts exporter definition.

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition. See Defining and editing data extractions, which describes how to set up and configure an extraction definition.
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

Note

A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.

4 To add conditions to the extraction setup:

Note

A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

A Open **Functions** > **Condition Definitions**.

B Select the table for which you want to define the conditions and click **OK**.

C Define the conditions for that table and click **OK**.

5 To export data to a flat file, in the **Destination type** field, select **None**.

6 Save the extraction setup. This will check for errors and validate the extraction setup.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

3.2.3 Create extracts exporter

Use the **Extracts Exporter Definitions** window to define exports for flat files and XML files. This window defines the structure of the export, attaches a reference file, and can also apply predefined plug-ins to format the extraction.

3.2.3.1 Prerequisite: Create field formats

Before you create an extracts exporter definition, define the formatting for each destination field of the export file. This is done in the **Field Formats** window. This is used to convert SimCorp Dimension internal data to the format required for the output file.

1. Open **CONNECTIVITY** > **INTERFACES** > **Field Formats** window.

   Alternatively, in the **Extracts Exporter Definitions** window, you can right-click the **Field format ID** field and select **Edit Table**.

1 Enter an ID and name, and select the data type for the export field.

Data type can be: **Date/time stamp**, **Numeric** or **Text**.

2. In the **General parameters** section, select width restrictions for the export field.
3. Depending on the data type, set the required parameters for the **Text Details** tab, the **Date/Time Stamp Details** tab, or the **Numeric Details** tab.

Note

In the **Examples** section, you can test and verify if the formatting is defined as required.

4. Save.

3.2.3.2 Create extracts exporter definition

To create an extracts exporter definition:

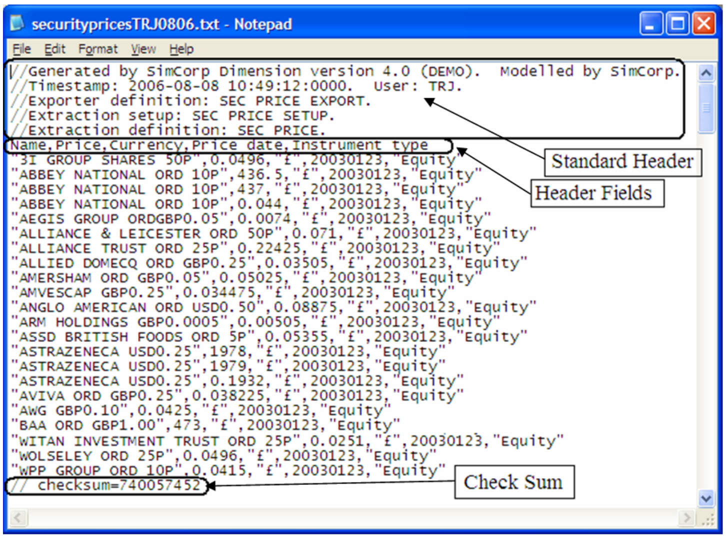
1. In the **Extracts Exporter Definitions** window, enter an ID and name.
2. Select an extraction setup in the **Extraction setup** field.See Create extraction setup, which describes how to create an extraction setup.
3. In the **Exporter type** field, select **New extraction**. This exporter type is for extraction setups of destination type **Message queue**, **Communication Server**, or **None**.

   The extracts exporter will execute the extract setup (data is selected from the SimCorp Dimension tables). Instead of writing the data to an extract table, the data is exported to a file according to the settings in the **Extracts Exporter Definitions** window.
4. In the **Export file** field, select the reference file that defines the naming conventions for this export.
5. In the **Separation type** field, select how the field should be separated in the flat file.

   You can select for example: **Tab separated**, **Semicolon separated**, **Fixed format**, **Comma separated**, **CSV – Semicolon**, **CSV – Quoted**, or **CSV – Comma**.
6. Select amongst the following:
   - **Standard header** — Select if you want a standard header to be added to the output file.
   - **Include check sum** — Select if you want a check sum comment in the output file. The check sum is a computed value that is dependent on the content of the exported file. Before importing, this value can then be verified, thereby providing some confidence that the information has not been tampered with.
   - **Include header fields** — Select if you want standard headers. The header fields is a line containing the name of the fields in the same

order as they are presented in the records. If selected, the **Header fields** field becomes available for edit.

- **Always create file** — Select if you want a file to be created, whether or not there are errors.



securitypricesTRJ0806.txt - Notepad
File Edit Format View Help
Standard Header
Header Fields
Check Sum

```
//Generated by SimCorp Dimension version 4.0 (DEMO).  Modelled by SimCorp.
//Timestamp: 2006-08-08 10:49:12:0000.  User: TRJ.
//Exporter definition: SEC PRICE EXPORT.
//Extraction setup: SEC PRICE SETUP.
//Extraction definition: SEC PRICE.
Name,Price,Currency,Price date,Instrument type
"3I GROUP SHARES 50P",0.0496,"£",20030123,"Equity"
"ABBEY NATIONAL ORD 10P",436.5,"£",20030123,"Equity"
"ABBEY NATIONAL ORD 10P",437,"£",20030123,"Equity"
"ABBEY NATIONAL ORD 10P",0.044,"£",20030123,"Equity"
"AEGIS GROUP ORDGBP0.05",0.0074,"£",20030123,"Equity"
"ALLIANCE & LEICESTER ORD 50P",0.071,"£",20030123,"Equity"
"ALLIANCE TRUST ORD 25P",0.22425,"£",20030123,"Equity"
"ALLIED DOMECQ ORD GBP0.25",0.03505,"£",20030123,"Equity"
"AMERSHAM ORD GBP0.05",0.05025,"£",20030123,"Equity"
"AMVESCAP GBP0.25",0.034475,"£",20030123,"Equity"
"ANGLO AMERICAN ORD USD0.50",0.08875,"£",20030123,"Equity"
"ARM HOLDINGS GBP0.0005",0.00505,"£",20030123,"Equity"
"ASSD BRITISH FOODS ORD 5P",0.05355,"£",20030123,"Equity"
"ASTRAZENECA USD0.25",1978,"£",20030123,"Equity"
"ASTRAZENECA USD0.25",1979,"£",20030123,"Equity"
"ASTRAZENECA USD0.25",0.1932,"£",20030123,"Equity"
"AVIVA ORD GBP0.25",0.038225,"£",20030123,"Equity"
"AWG GBP0.10",0.0425,"£",20030123,"Equity"
"BAA ORD GBP1.00",473,"£",20030123,"Equity"
"WITAN INVESTMENT TRUST ORD 25P",0.0251,"£",20030123,"Equity"
"WOLSELEY ORD 25P",0.0496,"£",20030123,"Equity"
"WPP GROUP ORD 10P",0.0415,"£",20030123,"Equity"
// checksum=740057452
```

7 In the **Rules** section, select the data to be exported:

A In the **Destination field** field, select the fields to be exported.

B In the **Position in data** field, enter the position of the field in the exported file.

If required, you can add blank columns to the export file by entering a position number without a destination field.

C If required, in the **Constant value** field for string or fixed value fields, add a constant value.

D In the **Field format ID** field, select the appropriate field formats for each destination field. (See Prerequisite: Create field formats to see how to create a field format).

E If required, enter a text string in the **Header fields** field. This extends the grid for the fields with a new column as header field. This field is added prior to the extracted data.

8 Save the extracts export definition.

9 You can now do the following:

- Preview the data extraction — Select **Functions** > **Preview Data**.
- Extract the currently defined data extraction — Select **Functions** > **Execute**.
- Export the current configuration of this window — Select **Functions** > **Export Definition**.

  This is used to exchange definitions between installations for example from SimCorp to customers. The file is saved in the extension ".xxd" as extracts export definitions.

- Add additional formatting to the current data extraction — Select **Functions** > **Plug-ins**.

  Plug-ins are described in Format data extractions.

3.2.4 Format data extractions

SimCorp provides a set of predefined functions in the form of plug-ins. Plug-ins are developed at the request of customers to add special business logic to extractions.

To apply the plug-in, in the **Extracts Exporter Definitions** window, select **Functions** > **Plug-ins**.

Select the required plug-in from the **Plug-in** field, and enter the required configuration in the parameter fields.

The **Information** pane provides you with information about the type of plug-in, and the type of parameter required in the field.

For a list of plug-ins, open the **Plug-in** field help in the **Extracts Exporter Definitions > Plug-ins** sub-window, follow the **Technical Information** link and expand the **Values in DXDFuncs** link. (Or see the online help version of the present topic which links there directly.)

3.3 Export data to XML file structure

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract for use in a spreadsheet or in another system. To do this, you can export the data to an XML file.

Before you can export the data to an XML file, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, follow the workflow to export the data to an XML file structure.

Process	See
Define the naming conventions for the external file names	Prerequisite: Create reference file
Create an extraction setup	Create extraction setup
Create extracts exporter definition	Create extracts exporter definition
Select plug-ins for the extracts exporter definition	Format data extractions

3.3.1 Prerequisite: Create reference file

To export data to an XML file, you use the **Extracts Exporter Definition** window. The **Extracts Exporter Definition** window requires a reference file, where you define the reference to external files and define naming conventions for the external file names. You can create the reference file before you start the process of exporting the data.

Generate export file names based on extracts exporter definition ID

You can generate export files in the **Extracts Exporter Definitions** window that use the ID of the extracts exporter definition as the file name.

- Open **DATA** > **REFERENCE DATA** > **Miscellaneous Codes** > **Reference Files** and select the **SimCorp Dimension config** component type. Then enter `EXPORTERID` as the component text.

You use the reference file in the **Extracts Exporter Definitions** window. The name of the export files is the ID of the extracts exporter definition.

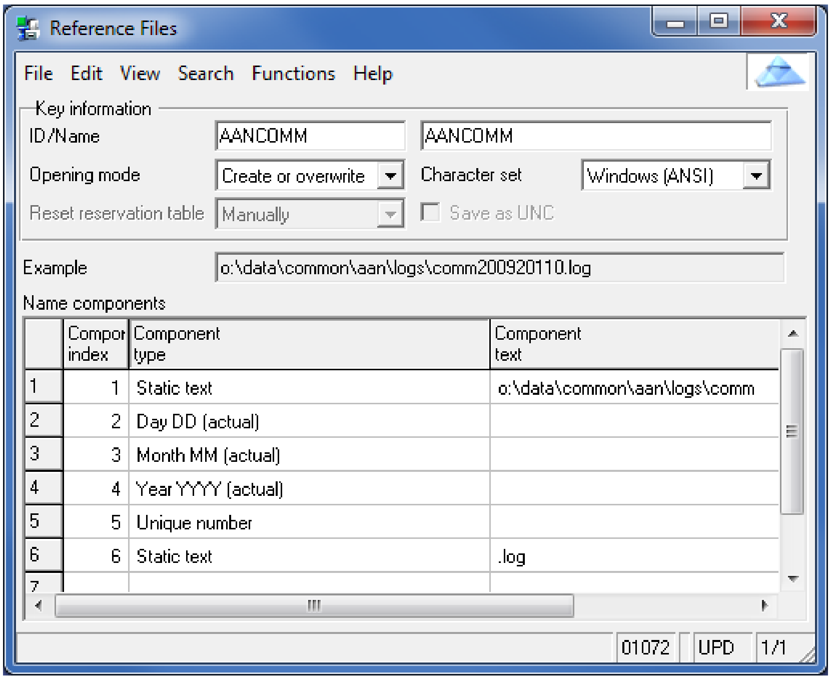
To create a reference file definition

1. Open the **DATA** > **REFERENCE DATA** > **Miscellaneous Codes** > **Reference Files** window, and enter an ID and name.
2. In the **Opening mode** field, select how the external file should be accessed.
   - **Append** — This option means that the new file is appended to an already existing file. If the file does not already exist, it is created.
   - **Create or overwrite** — This option means that a new file will be created. If it already exists, the content of the file will be overwritten with the new file.
   - **Create, exclusively** — This option means that a check will be performed to ensure that the file does not already exist. If it exists, you will receive a warning and the writing to the file is terminated.

     You must then select an option from the **Reset reservation table** field. This field is used in conjunction with the **Create, exclusively** opening mode and the **Unique number** or **Unique alpha** component type.

     The **Reset reservation table** field defines how often the unique number or unique alpha component of the filename is to be reset by the system. You can select: **Daily**, **Monthly**, **Yearly** or **Manually**. Manual resetting of the components is performed by selecting **Functions** > **Reset Reservation Table**.
   - **Read only** — This option is used when the file is used for read only purposes.
   - **Read only/Scan for files** — This option is used when you want to ensure that each file is only read once. By default, the **Save as UNC** check box is selected. This converts the file name to a UNC before it is saved.
3. Select the character set for the external file.
4. In the **Component type** field, select components of the file name structure that you are creating.
5. Save.

Example



Key information

ID/Name	AANCOMM	AANCOMM	
Opening mode	Create or overwrite	Character set	Windows (ANSI)
Reset reservation table	Manually	□ Save as UNC	

Example o:\data\common\aan\logs\comm200920110.log

Name components

	Compor index	Component type	Component text
1	1	Static text	o:\data\common\aan\logs\comm
2	2	Day DD (actual)	
3	3	Month MM (actual)	
4	4	Year YYYY (actual)	
5	5	Unique number	
6	6	Static text	.log
7			

01072 UPD 1/1

In this example, **Static text** provides the path where the extract will be saved. This is followed by file naming convention of **Day DD**, **Month MM**, and **Year YYYY**. Then the naming convention is followed by a **Unique number** to avoid a file being overwritten. This example then concludes the file convention with the file type (.log).

3.3.2 Create extraction setup

Use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the various destinations, such as user schemas, message queues, or communication servers.

The extraction setup can also be used on extracts exporter definitions. When you export data to an XML file structure, you connect the extraction definition with the extracts exporter definition.

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition. See Defining and editing data extractions, which describes how to set up and configure an extraction definition.
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

Note

A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.

---

4 To add conditions to the extraction setup:

Note

A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

A Open **Functions** > **Condition Definitions**.

B Select the table for which you want to define the conditions and click **OK**.

C Define the conditions for that table and click **OK**.

5 To export data to a flat file, in the **Destination type** field, select **None**.

6 Save the extraction setup. This will check for errors and validate the extraction setup.

You can see a preview of your data extraction from **Functions** > **Preview Data**. See View preview data, SQL query and execution plan for more information.

3.3.3 Create extracts exporter definition

Use the **Extracts Exporter Definitions** window to define exports for flat files and XML files. This window defines the structure of the export, attaches a reference file, and can also apply predefined plug-ins to format the extraction.

3.3.3.1 Prerequisite: Create field formats

Before you create an extracts exporter definition, define the formatting for each destination field of the export file. This is done in the **Field Formats** window. This is used to convert SimCorp Dimension internal data to the format required for the output file.

1 Open **CONNECTIVITY** > **INTERFACES** > **Field Formats** window.

Alternatively, in the **Extracts Exporter Definitions** window, you can right-click the **Field format ID** field and select **Edit Table**.

2 Enter an ID and name, and select the data type for the export field.

Data type can be: **Date/time stamp**, **Numeric** or **Text**.

3 In the **General parameters** section, select width restrictions for the export field.

4 Depending on the data type, set the required parameters for the **Text Details** tab, the **Date/Time Stamp Details** tab, or the **Numeric Details** tab.

Note

In the **Examples** section, you can test and verify if the formatting is defined as required.

5 Save.

3.3.3.2 Create extracts exporter definition

To create an extracts exporter definition:

1 In the **Extracts Exporter Definitions** window, enter an ID and name.
2 Select an extraction setup in the **Extraction setup** field. See Create extraction setup, which describes how to create an extraction setup.
3 In the **Exporter type** field, select **New extraction**. This exporter type is for extraction setups of destination type **Message queue**, **Communication Server**, or **None**.

The extracts exporter will execute the extract setup (data is selected from the SimCorp Dimension tables). Instead of writing the data to an extract table, the data is exported to a file according to the settings in the **Extracts Exporter Definitions** window.

4 In the **Export file** field, select the reference file that defines the naming conventions for this export. See Prerequisite: Create reference file, which describes how to create a reference file.
5 In the **Separation type** field, select **XML**.
6 Select amongst the following:
   - **Standard header** — Select if you want a standard header to be added to the output file.
   - **Include check sum** — Select if you want a check sum comment in the output file. The check sum is a computed value that is dependent on the content of the exported file. Before importing, this value can then be verified, thereby providing some confidence that the information has not been tampered with.
   - **Always create file** — Select if you want a file to be created, whether or not there are errors.

7 In the **Rules** section, select the data to be exported:
   A In the **Destination field** field, select the fields to be exported.
   B In the **Position in data** field, enter the position of the field in the exported file.

   If required, you can add blank columns to the export file by entering a position number without a destination field.

C. In the **XML tags** field, edit the required XML tags. This field contains the XML tag names for each field when creating XML files. The XML tags are defaulted to the SQL names.

D. If required, in the **Constant value** field for string or fixed value fields, add a constant value.

E. In the **Field format ID** field, select the appropriate field formats for each destination field. (See section Create extracts exporter definition to see how to create a field format).

8. Save the extracts export definition.
9. You can now do the following:
   - Preview the data extraction — Select **Functions** > **Preview Data**.
   - Extract the currently defined data extraction — Select **Functions** > **Execute**.
   - Export the current configuration of this window — Select **Functions** > **Export Definition**.

     This is used to exchange definitions between installations for example from SimCorp to customers. The file is saved in the extension ".xxd" as extracts export definitions.
   - Add additional formatting to the current data extraction — Select **Functions** > **Plug-ins**.

     Plug-ins are described in Format data extractions.

3.3.4 Format data extractions

SimCorp provides a set of predefined functions in the form of plug-ins. Plug-ins are developed at the request of customers to add special business logic to extractions.

To apply the plug-in, in the **Extracts Exporter Definitions** window, select **Functions** > **Plug-ins**.

Select the required plug-in from the **Plug-in** field, and enter the required configuration in the parameter fields.

The **Information** pane provides you with information about the type of plug-in, and the type of parameter required in the field.

For a list of plug-ins, open the **Plug-in** field help in the **Extracts Exporter Definitions > Plug-ins** sub-window, follow the **Technical Information** link and expand the **Values in DXDFuncs** link. (Or see the online help version of the present topic which links there directly.)

3.4 Extract data to the message queue

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract for use back into the SimCorp Dimension system. This allows you to automatically update multiple records. To do this, you extract the data to the message queue.

Before you can extract the data to the message queue, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the message queues.

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition.
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used

Note

A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.

4. To add conditions to the extraction setup:

Note

A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

   A. Open **Functions** > **Condition Definitions**.
   B. Select the table for which you want to define the conditions and click **OK**.
   C. Define the conditions for that table and click **OK**.

5. To export data to the message queue, in the **Destination type** field, select **Message queue**.
6. On the **Message Queue** tab:
   A. Select the message queue to which you will export the data.
   B. In the **Insertion mode** field, select one of the following:
      - **Configure Data to Filter** — Then select a filter or a data format setup in the **Configure Data to Filter** window.
      - **Insert via Data Format Setup** — Then select the data format setup in the **Data Format Setup** field. This insertion mode processes the **Import to Message Queue** function for the data format setup. It generates SC formatted records when processed. The data format setup is applied to each record of the data before it is sent to the message queue.
      - **Transfer to Message Queue** — Then select the data format setup in the **Data Format Setup** field. This insertion mode inserts the external format for the selected data format setup. The data is not validated and all records are inserted with 'Queued' message status. Each record is inserted with base filter and the data format setup specified in front. The data format setup is not applied to the data before it is inserted to the message queue.
7. Save the extraction setup. This will check for errors and validate the extraction setup.
8. You can now do the following:
   - Preview the data extraction — Select **Functions** > **Preview Data**.
   - Extract the currently defined data extraction — Select **Functions** > **Execute**.
   - Export the current configuration of this window — Select **Functions** > **Export Setup**.

     This is used to exchange setups between installations for example from SimCorp to customers. The file is saved in the extension ".dxs" as extraction setup files.

3.5 Extract data to the communication server

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract to the communication server. For information about setting up and using the communication server, see the ***Communication Server*** user manual.

Before you can extract the data to the communication server, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the communication server.

1 In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2 In the **Source** section, select an extraction definition.
3 If required, select a segment that is specific to this extraction setup.

If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

Note

A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.

---

4 To add conditions to the extraction setup:

Note

A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

A Open **Functions** > **Condition Definitions**.
B Select the table for which you want to define the conditions and click **OK**.
C Define the conditions for that table and click **OK**.

5 To export data to the message queue, in the **Destination type** field, select **Communication Server**.
6 In the **Communication Server** tab:

A Select the configurable message type for the Communication Server. In the **Configurable Message Types** window, you can define the distribution type as either **Data request**, **Extract data – Point-2-Point**, or **Extract data – Subscribe**.
B In the **Message label** field, specify the message label to be used by a handler in the Communication Server solution.
C In the **Format** field, select either **XML-data** or **XML-text** for the format of the extracted data to be sent to the Communication Server

7 Save the extraction setup. This will check for errors and validate the extraction setup.
8 You can now do the following:

- Preview the data extraction — Select **Functions** > **Preview Data**.

  - Extract the currently defined data extraction — Select **Functions** > **Execute**.
  - Export the current configuration of this window — Select **Functions** > **Export Setup**.

    This is used to exchange setups between installations for example from SimCorp to customers. The file is saved in the extension ".dxs" as extraction setup files.

3.6 Extract data to an external application

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract to an external application.

Before you can export the data to an external application, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, follow the workflow to export the data to an external application.

Process	See
Create database user schemas. The extracted data is stored in separate database user schemas in the SimCorp Dimension database.	Prerequisite: Create database user schema
Create the extraction setup, which links the extraction definition to the database user schema.	Create extraction setup

3.6.1 Prerequisite: Create database user schema

A database user schema is a dedicated space for data extractions or Communication Server in the Oracle® database. These must be defined in advance. You must have at least one database user schema per SimCorp Dimension installation.

The following types of database user schema are available:

- **Communication server user schema** — These are used with the Communication Server. See the ***Communication Server*** user manual for more information.
- **External extract user schema** — These are used with the **Extraction Setup** window. They are created and modified outside of SimCorp Dimension, but can be populated from inside SimCorp Dimension. They must be registered in SimCorp Dimension.
- **Internal extract user schema** — These are used with the **Extraction Setup** window. They are created and modified from inside SimCorp Dimension, and they are maintained by SimCorp Dimension.
- **Data Warehouse schema** — These are used with the **Extraction Setup** window and makes it possible to update, insert and delete records in an externally modelled database schema. See Extract data to a data warehouse schema for more information.

3.6.1.1 Create database user schema

Database user schemas are created in the **MAINTENANCE** > **SYSTEM MAINTENANCE** > **Database Administration** > **Database User Schemas Administration** window.

For security reasons, authorisation rights to the **Database User Schemas Administration** window are typically restricted to the Oracle® database administrator. For more information on user rights, refer to the ***System Access*** user manual.

1. In the **Database User Schemas Administration** window, enter an ID and schema name.
2. In the **Schema type** field, select whether the database user schema is in internal schema or an external schema.

   If you are registering an external user schema, it must first exist in the database.

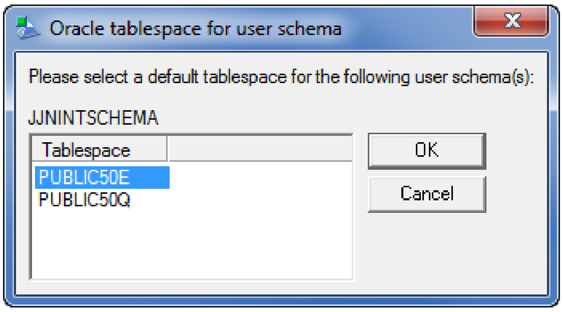
   Note

   It is recommended that if normal users are granted authorisation rights to create database user schemas, then they only create **Internal extract user schemas**.

   ---

3. If required, select the **Limit** check box and enter a maximum size for database usage.
4. Save.

   You are prompted to select a tablespace for the user schema.



Oracle tablespace for user schema

Please select a default tablespace for the following user schema(s):

JJNINTSCHEMA

Tablespace

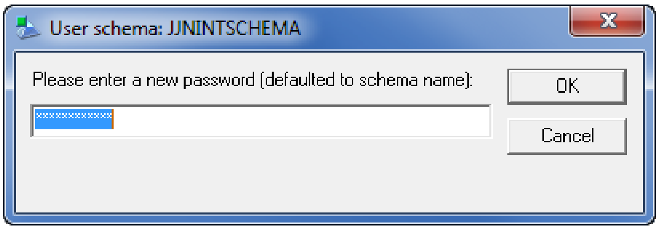
PUBLIC50E

PUBLIC50Q

For technical information on user schemas and tablespaces, refer to the SimCorp Dimension ***System Administrators*** manual.

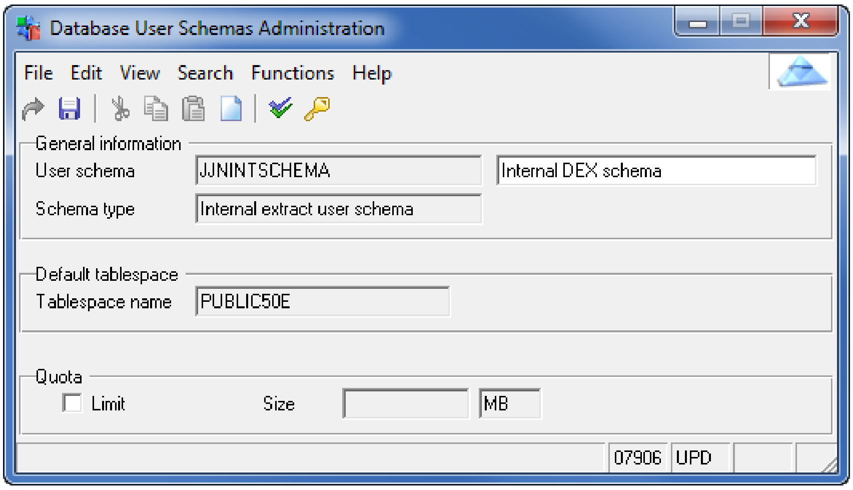
5. Select the appropriate tablespace and click **OK**.

   You are prompted to enter a password for the user schema. By default, it is the user schema name.


User schema: JJNINTSCHEMA
Please enter a new password (defaulted to schema name):
****
OK
Cancel

6. Click **OK**.

   The database user schema is created.


Database User Schemas Administration
File Edit View Search Functions Help

General information

User schema: JJNINTSCHEMA — Internal DEX schema

Schema type: Internal extract user schema

Default tablespace

Tablespace name: PUBLIC50E

Quota

☐ Limit   Size   MB

07906 UPD

Verify the user schema

You can check the integrity of an internal extract user schema by selecting **Functions** > **Verify Schema**. This is often useful during upgrade.

If an error is found, you will be asked if the problems should be rectified.

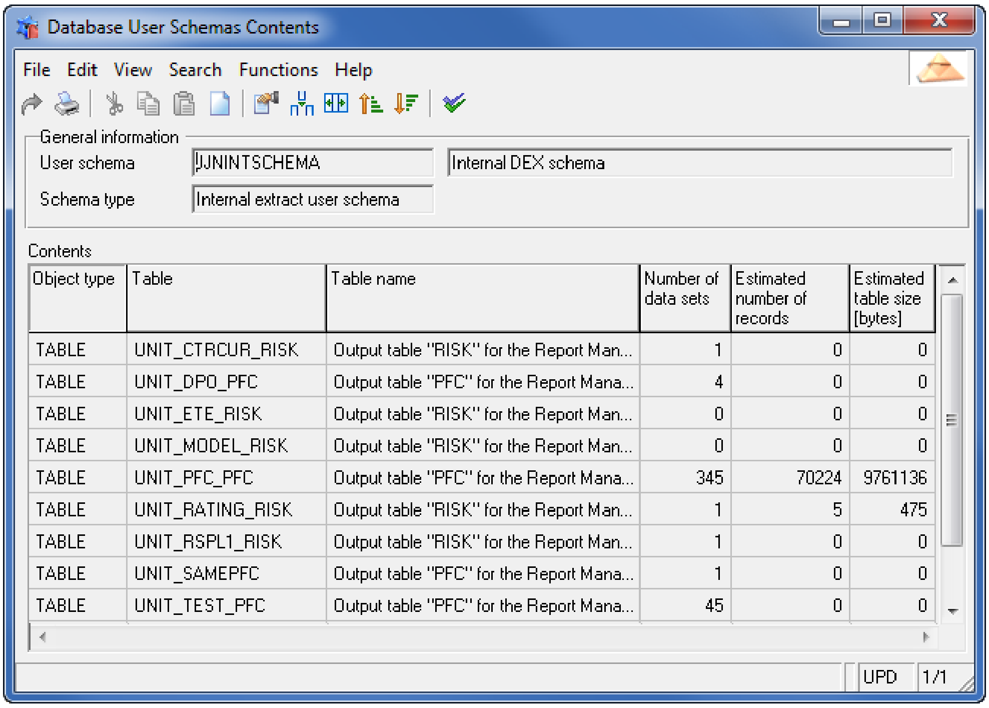
Recreate the user schema

If an internal user schema has been dropped by an external application, you can recreate it by selecting **Functions** > **Recreate Schema**.

This action deletes all objects in the user schema and resets the password.

3.6.1.2 Maintain database user schema

Database user schemas are maintained in the **MAINTENANCE** > **SYSTEM MAINTENANCE** > **Database Administration** > **Database User Schemas Contents** window. This window is used for viewing the contents of a database user schema.



General information

User schema: JJNINTSCHEMA — Internal DEX schema

Schema type: Internal extract user schema

Contents

Object type	Table	Table name	Number of data sets	Estimated number of records	Estimated table size [bytes]
TABLE	UNIT_CTRCUR_RISK	Output table "RISK" for the Report Man...	1	0	0
TABLE	UNIT_DPO_PFC	Output table "PFC" for the Report Mana...	4	0	0
TABLE	UNIT_ETE_RISK	Output table "RISK" for the Report Man...	0	0	0
TABLE	UNIT_MODEL_RISK	Output table "RISK" for the Report Man...	0	0	0
TABLE	UNIT_PFC_PFC	Output table "PFC" for the Report Mana...	345	70224	9761136
TABLE	UNIT_RATING_RISK	Output table "RISK" for the Report Man...	1	5	475
TABLE	UNIT_RSPL1_RISK	Output table "RISK" for the Report Man...	1	0	0
TABLE	UNIT_SAMEPFC	Output table "PFC" for the Report Mana...	1	0	0
TABLE	UNIT_TEST_PFC	Output table "PFC" for the Report Mana...	45	0	0

The **Database User Schemas Contents** window shows the information for the tables or other objects that are defined in the selected user schema. The grid shows the following information for each table:

- **Number of data sets** — This field shows the number of different data extractions in that particular table. If the table is set to overwrite for each extract, this will always be ‘1’. If the table is set to append, then the value will increase.
- **Estimated number of records** — This field shows the estimated number of records in the extract table. The number is estimated based on statistical information retrieved from the database server.
- **Estimated table size [bytes]** — This field shows the estimated physical size of the extract table. The size is estimated based on statistical information retrieved from the database server.

In order to ensure that the most recent and correct information is shown, update the table statistics. Right-click the grid and select **Update Table Statistics**.

3.6.2 Create extraction setup

Use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the various destinations, such as user schemas, message queues, communication servers, or extracts exporter definitions. When you export data to an external application, you connect the extraction definition with the database user schemas.

Note

Alternatively, you could export to the Communication Server, which then sends the data to an external application. See Extract data to the communication server for more information.

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition.

   (See Defining and editing data extractions, which describes how to set up and configure an extraction definition).
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

   ### Note

   A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.
4. To add conditions to the extraction setup:

   ### Note

   A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

   A. Open **Functions** > **Condition Definitions**.

   B. Select the table for which you want to define the conditions and click **OK**.

   C. Define the conditions for that table and click **OK**.
5. To export data to an external application, in the **Destination type** field, select **Extract table**.

6. In the **Extract Table** tab:
   1. Select the database user schema. (See Prerequisite: Create database user schema to see how to create a database user schema).

      You also have the option to use the destination table specified in the **Extraction Definitions** or **Advanced Extraction Definitions** windows as the data storage. Select the **Use table from Extraction Definition** option button to automatically add the user schema and table name to the extraction setup (for extraction definitions with a destination table specified).
   2. In the **Table name** field, edit the name of the extract table to be created or used.

      It is possible to use the same table in multiple extraction setups, provided that the extraction definition is the same.
   3. In the **Storage option** field, select how the data is stored in the extract table.
      - If you select **Append**, an extract to the user schema extract table is made, where the results are appended to the existing results.
      - If you select **Default**, the storage setting selected in the **Extraction Definitions** window on the **Settings** tab will be used.
      - If you select **Overwrite**, an extract is made to the selected database user schema extract table. That is, all existing data extractions previously made with this extraction setup are deleted, and a new extract is made in the extract table.
      - If you select **Overwrite all**, an extract is made to the selected database user schema extract table, and all existing data extractions in the same database user schema extract table are overwritten.
      - See Extract data to a data warehouse schema for more information about the **Update**, **Update and Insert**, and **Delete** storage options.

D. Define the authorisation settings for the data extraction. These apply to all data extractions executed with the extraction setup. In the **Default read scope** field and the **Default write scope** field, select the read and write privileges for the data extraction (see Authorisation structure with extraction setups on page 63 for more information about security for data extractions):

   - If you select **Private**, only the user who creates the data extraction is allowed to read and/or write to the data extraction.
   - If you select **Public**, every user is allowed to read and/or write to the data extraction.
   - If you select **Group**, access is limited to a selected group. This group is selected from the available choices in **Default auth. Profile** field. Note that the authorisation profiles are created by the system administrator. See the ***System Access*** user manual for information about authorisation profiles.

7. Save the extraction setup. This will check for errors and validate the extraction setup.
8. Select **Functions** > **Check Extract Table**. This performs an update check on the defined extract table, if it exists. If an extract table does not already exist, this creates one.

   If there are irregularities, the system points it out and suggests solutions.
9. If you are processing the data extraction via calculation servers, start the server and then select **Options** > **Enable Server Execution**.

Note

This is a user specific option. This means that the selection is saved and will be reused each time the **Extraction Setups** window is opened, regardless of the selected extraction setup.

If you do not select **Enable Server Execution**, the execution runs locally. This enables the execution to run faster in cases where there are no available servers. However, if it is a large extraction to be executed, it can take a considerable amount of time. A local execution may result in the local machine being occupied for other tasks during that period.

10. You can now do the following:
    - Preview the data extraction — Select **Functions** > **Preview Data**.
    - Extract the currently defined data extraction — Select **Functions** > **Execute**.

      This extracts the data to the defined table in the selected user schema. If a table has not yet been created, the **Execute** function also triggers the creation of it
    - Export the current configuration of this window — Select **Functions** > **Export Setup**.

      This is used to exchange setups between installations for example from SimCorp to customers. The file is saved in the extension ".dxs" as extraction setup files.

After you execute the data extraction, you can view it by selecting **Functions** > **View Extracts**. See Viewing extract table contents and structure for more information about viewing data extractions.

3.7 Extract data to a report

When an extraction definition has been saved, the data that you want to extract can be processed in different ways. One way is to export the extract to a report. Extract tables automatically become available to the report interface if they are placed in an internal user schema.

Note

Alternatively, you can export the data to an external user schema, and then connect to this schema directly from the Crystal reports interface.

See the ***Reporting*** user manual for more information about creating reports.

Before you can export the data to a report, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, follow the workflow to export the data to a report.

Process	See
Create database user schemas. The extracted data is stored in separate database user schemas in the SimCorp Dimension database.	Prerequisite: Create database user schema
Create the extraction setup, which links the extraction definition to the database user schema.	Create extraction setup

Process	See
Create the reports in Crystal Reports (or in another 3rd party reporting tool).	Create reports based on extraction tables
Execute the data extraction and the report.	Execute reports based on extraction tables

3.7.1 Prerequisite: Create database user schema

A database user schema is a dedicated space for data extractions or Communication Server in the Oracle® database. These must be defined in advance. You must have at least one database user schema per SimCorp Dimension installation.

The following types of database user schema are available:

- **Communication server user schema** — These are used with the Communication Server. See the ***Communication Server User Manual*** for more information.
- **External extract user schema** — These are used with the **Extraction Setup** window. They are created and modified outside of SimCorp Dimension, but can be populated from inside SimCorp Dimension. They must be registered in SimCorp Dimension.
- **Internal extract user schema** — These are used with the **Extraction Setup** window. They are created and modified from inside SimCorp Dimension, and they are maintained by SimCorp Dimension.
- **Data Warehouse schema** — These are used with the **Extraction Setup** window and makes it possible to update, insert and delete records in an externally modelled database schema. See Extract data to a data warehouse schema for more information.

3.7.1.1 Create database user schema

Database user schemas are created in the **MAINTENANCE** > **SYSTEM MAINTENANCE** > **Database Administration** > **Database User Schemas Administration** window.

For security reasons, authorisation rights to the **Database User Schemas Administration** window are typically restricted to the Oracle® database administrator. For more information on user rights, refer to the ***System Access*** user manual.

1. In the **Database User Schemas Administration** window, enter an ID and schema name.
2. In the **Schema type** field, select whether the database user schema is in internal schema or an external schema.

   If you are registering an external user schema, it must first exist in the database.

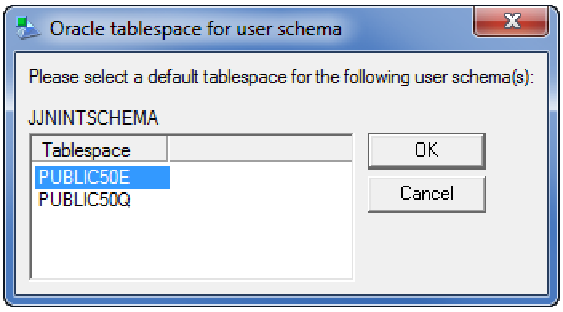
Note

It is recommended that if normal users are granted authorisation rights to create database user schemas, then they only create **Internal extract user schemas**.

3 If required, select the **Limit** check box and enter a maximum size for database usage.

4 Save.

You are prompted to select a tablespace for the user schema.



Oracle tablespace for user schema

Please select a default tablespace for the following user schema(s):

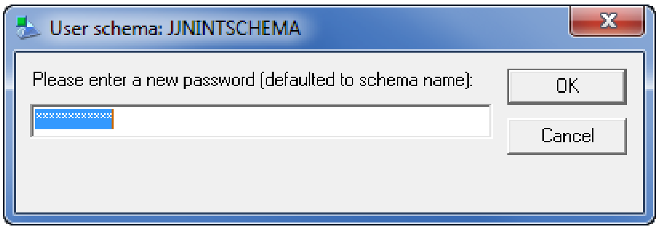
JJNINTSCHEMA

Tablespace
PUBLIC50E
PUBLIC50Q

For technical information on user schemas and tablespaces, refer to the SimCorp Dimension ***System Administrators Manual***.

5 Select the appropriate tablespace and click **OK**.

You are prompted to enter a password for the user schema.



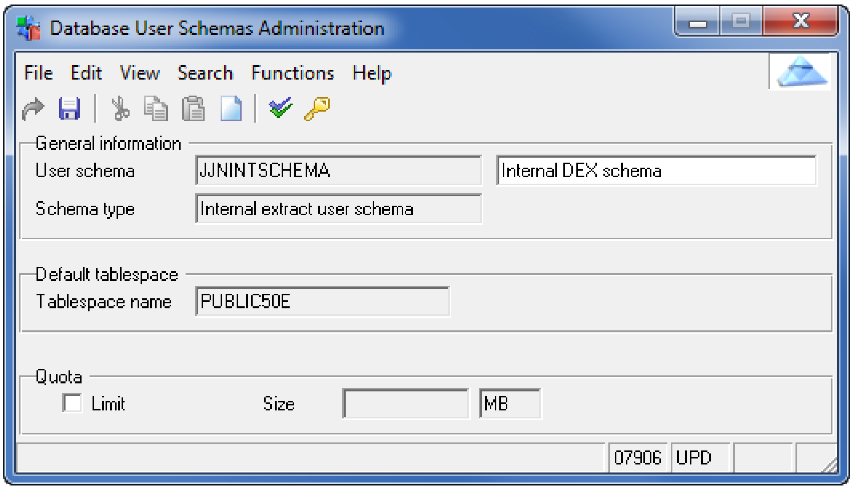
User schema: JJNINTSCHEMA

Please enter a new password (defaulted to schema name):

****

6 Click **OK**.

The database user schema is created.


Database User Schemas Administration
File Edit View Search Functions Help

General information
User schema: JJNINTSCHEMA — Internal DEX schema
Schema type: Internal extract user schema

Default tablespace
Tablespace name: PUBLIC50E

Quota
□ Limit Size: MB

07906 UPD

Verify the user schema

You can check the integrity of an internal extract user schema by selecting **Functions** > **Verify Schema**. This is often useful during upgrade.

If an error is found, you will be asked if the problems should be rectified.

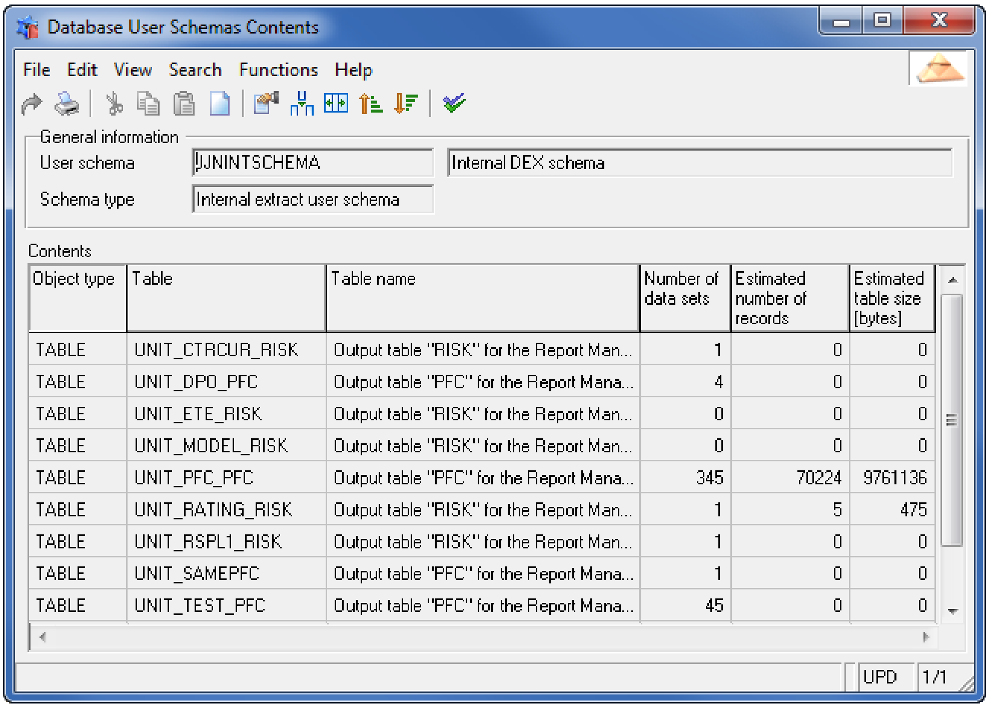
Recreate the user schema

If an internal user schema has been dropped by an external application, you can recreate it by selecting **Functions** > **Recreate Schema**.

This action deletes all objects in the user schema and resets the password.

3.7.1.2 Maintain database user schema

Database user schemas are maintained in the **MAINTENANCE** > **SYSTEM MAINTENENACE** > **Database Administration** > **Database User Schemas Contents** window. This window is used for viewing the contents of a database user schema.



General information

User schema: JJNINTSCHEMA — Internal DEX schema

Schema type: Internal extract user schema

Contents

Object type	Table	Table name	Number of data sets	Estimated number of records	Estimated table size [bytes]
TABLE	UNIT_CTRCUR_RISK	Output table "RISK" for the Report Man...	1	0	0
TABLE	UNIT_DPO_PFC	Output table "PFC" for the Report Mana...	4	0	0
TABLE	UNIT_ETE_RISK	Output table "RISK" for the Report Man...	0	0	0
TABLE	UNIT_MODEL_RISK	Output table "RISK" for the Report Man...	0	0	0
TABLE	UNIT_PFC_PFC	Output table "PFC" for the Report Mana...	345	70224	9761136
TABLE	UNIT_RATING_RISK	Output table "RISK" for the Report Man...	1	5	475
TABLE	UNIT_RSPL1_RISK	Output table "RISK" for the Report Man...	1	0	0
TABLE	UNIT_SAMEPFC	Output table "PFC" for the Report Mana...	1	0	0
TABLE	UNIT_TEST_PFC	Output table "PFC" for the Report Mana...	45	0	0

The **Database User Schemas Contents** window shows the information for the tables or other objects that are defined in the selected user schema. The grid shows the following information for each table:

- **Number of data sets** — This field shows the number of different data extractions in that particular table. If the table is set to overwrite for each extract, this will always be '1'. If the table is set to append, then the value will increase.
- **Estimated number of records** — This field shows the estimated number of records in the extract table. The number is estimated based on statistical information retrieved from the database server.
- **Estimated table size [bytes]** — This field shows the estimated physical size of the extract table. The size is estimated based on statistical information retrieved from the database server.

In order to ensure that the most recent and correct information is shown, update the table statistics. Right-click the grid and select **Update Table Statistics**.

3.7.2 Create extraction setup

Use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. This window connects the extraction definitions with the various destinations, such as user schemas, message queues, communication servers, or extracts exporter definitions. When you export data to a report, you connect the extraction definition with the database user schemas.

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition.

   (See Defining and editing data extractions, which describes how to set up and configure an extraction definition).
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retrieve data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

   Note

   A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definition** window.
4. To add conditions to the extraction setup:

   Note

   A condition in the **Extraction Setup** window enhances or overrides condition defined in the **Extraction Definition** window.

   A. Open **Functions** > **Condition Definitions**.

   B. Select the table for which you want to define the conditions and click **OK**.

   C. Define the conditions for that table and click **OK**.
5. To export data to an external application, in the **Destination type** field, select **Extract table**.

6. In the **Extract Table** tab:

   A. Select the database user schema. (See Prerequisite: Create database user schema to see how to create a database user schema).

      You also have the option to use the destination table specified in the **Extraction Definitions** or **Advanced Extraction Definitions** windows as the data storage. Select the **Use table from Extraction Definition** option button to automatically add the user schema and table name to the extraction setup (for extraction definitions with a destination table specified).

   B. In the **Table name** field, edit the name of the extract table to be created or used.

      It is possible to use the same table in multiple extraction setups, provided that the extraction definition is the same.

   C. In the **Storage option** field, select how the data is stored in the extract table.

      - If you select **Append**, an extract to the user schema extract table is made, where the results are appended to the existing results.
      - If you select **Default**, the storage setting selected in the **Extraction Definitions** window on the **Settings** tab will be used.
      - If you select **Overwrite**, an extract is made to the selected database user schema extract table, and all existing data extractions previously made with this extraction setup are deleted.
      - If you select **Overwrite all**, an extract is made to the selected database user schema extract table, and all existing data extractions in the same database user schema extract table are overwritten.
      - See Extract data to a data warehouse schema for more information about the **Update**, **Update and Insert**, and **Delete** storage options.

D. Define the authorisation settings for the data extraction. These apply to all data extractions executed with the extraction setup. In the **Default read scope** field and the **Default write scope** field, select the read and write privileges for the data extraction (see Authorisation structure with extraction setups on page 63 for more information about security for data extractions):

   - If you select **Private**, only the user who creates the data extraction is allowed to read and/or write to the data extraction.
   - If you select **Public**, every user is allowed to read and/or write to the data extraction.
   - If you select **Group**, access is limited to a selected group. This group is selected from the available choices in **Default auth. Profile** field. Note that the authorisation profiles are created by the system administrator. See the ***System Access*** user manual for information about authorisation profiles.

7. Save the extraction setup. This will check for errors and validate the extraction setup.
8. Select **Functions** > **Check Extract Table**. This performs an update check on the defined extract table, if it exists. If an extract table does not already exist, this creates one.

   If there are irregularities, the system points it out and suggests solutions.
9. If you are processing the data extraction via calculation servers, start the server and then select **Options** > **Enable Server Execution**.

   Note

   This is a user specific option. This means that the selection is saved and will be reused each time the **Extraction Setups** window is opened, regardless of the selected extraction setup.

   If you do not select **Enable Server Execution**, the execution runs locally. This enables the execution to run faster in cases where there are no available servers. However, if it is a large extraction to be executed, it can take a considerable amount of time. A local execution may result in the local machine being occupied for other tasks during that period.

10. You can now do the following:
    - Preview the data extraction — Select **Functions** > **Preview Data**.
    - Extract the currently defined data extraction — Select **Functions** > **Execute**.

      This extracts the data to the defined table in the selected user schema. If a table has not yet been created, the **Execute** function also triggers the creation of it.
    - Export the current configuration of this window — Select **Functions** > **Export Setup**.

      This is used to exchange setups between installations for example from SimCorp to customers. The file is saved in the extension ".dxs" as extraction setup files.

After you execute the data extraction, you can view it by selecting **Functions** > **View Extracts**. See Viewing extract table contents and structure for more information about viewing data extractions.

3.7.3 Create reports based on extraction tables

The next step is to create reports in Crystal Reports (or in another 3rd party reporting tool) which are based on the extraction tables and install the report in SimCorp Dimension.

3.7.4 Execute reports based on extraction tables

To execute reports that are based on extraction tables, the first step is to execute the data extraction in the **Extraction Setups** window. The next step is to execute the report in the **Report Properties** window.

If the data extraction contains data from calculations in SimCorp Dimension, remember to execute these calculations before the data extraction is executed.

It is recommended to include the data extraction and the report execution as batch jobs in a batch job group to make sure that the report is based on the latest data. You can also include other batch jobs if the result of these batch jobs needs to be included in the report (for example calculations). It is also possible to create online jobs (in the **Online Jobs** window) to execute the data extraction and report execution.

For example, include portfolio calculations in the batch job group if these calculations need to be included in a holdings report based on extraction tables. Include the following batch jobs in the batch job group:

1. Execute the portfolio calculation
2. Execute the extraction setup
3. Execute the report in report properties

3.8 Extract data to a data warehouse schema

You can update, insert and delete records in data warehouse schemas in the Data Extractor. The following additional operations are available:

- Update — Update the target records (using the MERGE statement). The matched records are updated.
- Update and Insert — Update and insert target records (using the MERGE statement). The matched records are updated and then non-matched records are inserted.
- Delete — Delete target records. The matched records are deleted.

You can, for example, use the functionality to update and insert records in a data warehouse schema when running incremental loads of transactions to a data warehouse or reporting database.

Before you can export data to a data warehouse schema, you must first create an extraction definition (see Defining and editing data extractions).

After you create an extraction definition, follow the workflow to export the data to a data warehouse schema.

Process	See
Create a data warehouse schema. The extracted data is stored in separate database user schemas in the SimCorp Dimension database.	Prerequisite: Create database user schema
Create the extraction setup, which links the extraction definition to the data warehouse schema.	Create extraction setup

3.8.1 Prerequisite: Create database user schema

The following additional operations are available in data warehouse schemas:

- Update — Update the target records (using the MERGE statement). The matched records are updated.
- Update and Insert — Update and insert target records (using the MERGE statement). The matched records are updated and the non-matched records are inserted.
- Delete — Delete target records. The matched records are deleted.

You can, for example, use the functionality to update and insert records in a data warehouse schema when running incremental loads of transactions to a data warehouse or reporting database.

The data warehouse schema is a separate database user schema. Primary keys for the tables are supported in a read-only mode for data warehouse schemas.

To create a data warehouse schema in SimCorp Dimension:

1. Create the data warehouse schema in the database by using Oracle or other 3rd party tools. Create the tables in the data warehouse schema.

You must maintain primary keys and constraints in the tables directly on the database. You cannot edit the tables in data warehouse schemas from SimCorp Dimension.

2. Register the data warehouse schema in SimCorp Dimension. Both the data warehouse schema and the tables must be registered.
   - Register the data warehouse schema in the **Database User Schemas Administration** window.
   - Register the tables in the **Database User Schemas Contents** window.

     All tables are shown in the **Contents** section. The **Status** field shows if the table is **Ok**, **Not registered** or **Does not exist**. You can register the non-registered tables by selecting **Functions** > **Register All Tables**, or right-click a table and select **Register Table**.
3. Use the tables in the data warehouse schema.
   - SELECT data from the tables in the **Extraction Definitions** window.
   - INSERT data in the tables.
   - UPDATE data in the tables. Both Update, and Update and Insert, are supported.
   - DELETE data from the tables.

3.8.2 Data Warehouse: Creating field names in tables and views

When you are creating a **Data Warehouse schema** for use in the Data Extractor, the field names in tables and views must be specified in uppercase. For example:

```
CREATE OR REPLACE FORCE VIEW "AYF_STAGE"."SRV_VIEW"
(
"POR", "TRANSEX", "SECSHORT"
)
AS
SELECT PF.POR,
TR.TRANSEX,
SEC.SECSHORT
FROM TTRANSMAIN@KBK_DBLINK1.WORLD TR,
PORTFOLIOS@KBK_DBLINK1.WORLD PF,
SECURITIES@KBK_DBLINK1.WORLD SEC
WHERE TR.PORIK=PF.PORIK
AND TR.SECIK=SEC.SECIK
AND PF.PORIK=3099;
```

You must create the tables and views as Oracle objects in the Data Warehouse schemas, by using SQL expressions or third party tools (for example, SQL*Plus and Oracle SQL Developer).

When the objects have been created, you must register them in the **Database User Schemas Contents** window to use them in SimCorp Dimension.

3.8.3 Create extraction setup

Use the **Extraction Setups** window to control the execution of the extraction definitions and the storage of the extracts. The window connects the extraction definitions with the various destinations, such as user schemas, message queues, communication servers, or extracts exporter definitions. When you export data to a database user schema, you connect the extraction definition with the database user schemas.

To create an extraction setup for data warehouse schemas:

1. In the **Extraction Setups** window, enter an ID and name for the extraction setup.
2. In the **Source** section, select an extraction definition. See Defining and editing data extractions
3. If required, select a segment that is specific to this extraction setup.

   If you are specifying your extraction setup by using segments, you must first create a general segment. Segments are a convenient shortcut to retreive data from the database. Each segment is given an identification, and instead of repeating the search conditions each time you want to retrieve something from the database, the segment identification is used.

Note

A segment in the **Extraction Setup** window enhances or overrides a segment defined in the **Extraction Definitions** window.

4. To add conditions to the extraction setup:

Note

A condition in the **Extraction Setup** window enhances or overrides conditions defined in the **Extraction Definitions** window.

   A. Open **Functions** > **Condition Definitions**.
   B. Select the table for which you want to define the conditions and click **OK**.
   C. Define the conditions for that table and click **OK**.

5. To export data to a data warehouse schema, in the **Destination type** field, select **Extract table**.
6. On the **Extract Table** tab:

A. Select the database user schema (see Prerequisite: Create database user schema for more information).

   You also have the option to use the destination table specified in the **Extraction Definitions** or **Advanced Extraction Definitions** as the data storage. Select the **Use table from Extraction Definition** option button to automatically add the user schema and table name to the extraction setup (for extraction definitions with a destination table specified).

B. In the **Table name** field, edit the name of the extract table to be created or used. It is possible to use the same table in multiple extraction setups, provided that the extraction definition is the same.

C. In the **Storage option** field, select how the data is stored in the extract table.

Storage option	Description
Delete	Delete the target records. The matched records are deleted.
Update	Update the target records (using the MERGE statement). The matched records are updated.
Update and Insert	Update and insert target records (using the MERGE statement). The matched records are updated and the non-matched records are inserted.

7. Save the extraction setup. This will check for errors and validate the extraction setup.
8. Select **Functions** > **Check Extract Table**. This performs an update check on the defined extract table, if it exists. If an extract table does not already exist, this creates one.

   If there are irregularities, the system points it out and suggests solutions.
9. If you are processing the data extraction via calculation servers, start the server and then select **Options** > **Enable Server Execution**.

Note

This is a user specific option. This means that the selection is saved and will be reused each time the **Extraction Setups** window is opened, regardless of the selected extraction setup.

If you do not select **Enable Server Execution**, the execution runs locally. This enables the execution to run faster in cases where there are no available servers. However, if it is a large extraction to be executed, it can take a considerable amount of time. A local execution may result in the local machine being occupied for other tasks during that period.

10 You can now do the following:

- Preview the data extraction — Select **Functions** > **Preview Data**.
- Extract the currently defined data extraction — Select **Functions** > **Execute**.

  This extracts the data to the defined table in the selected user schema. If a table has not yet been created, the **Execute** function also triggers the creation of it.
- Export the current configuration of this window — Select **Functions** > **Export Setup**.

  This is used to exchange setups between installations for example from SimCorp to customers. The file is saved with the extension ".dxs" as extraction setup files.

After you execute the data extraction, you can view it by selecting **Functions** > **View Extracts**. See Viewing extract table contents and structure for more information.

4 Viewing extract table contents and structure

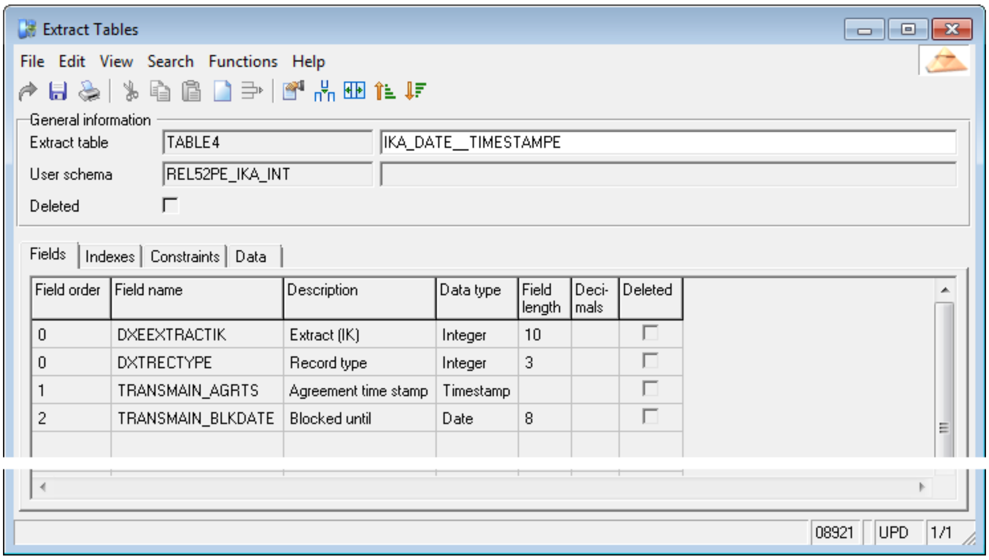
This section shows how to see the content and structure of your data extracts:

- View the structure of the extract table
- View the data extract

4.1 View the structure of the extract table

After you execute the data extraction, you can analyse the structure of the extract table. This allows you to verify that the correct fields are selected for the extract table, and the attributes such as data type and field length are correct.

Use the **Extract Tables** window to analyse the fields, indexes, constraints and data in the extract table created from internal extract user schemas.



Extract table: TABLE4 IKA_DATE__TIMESTAMPE

User schema: REL52PE_IKA_INT

Deleted: ☐

Field order	Field name	Description	Data type	Field length	Deci-mals	Deleted
0	DXEEXTRACTIK	Extract (IK)	Integer	10		☐
0	DXTRECTYPE	Record type	Integer	3		☐
1	TRANSMAIN_AGRTS	Agreement time stamp	Timestamp			☐
2	TRANSMAIN_BLKDATE	Blocked until	Date	8		☐

This window enables you to do the following:

Function	Description
Check the layout of the extract tables	In the **Fields** tab, you can see the structure of the extract table, including field order, field name, description, data type, field length, decimals, and whether or not the field has been deleted.
Validate the extract table, and ensure that it is up to date	Select **Functions** > **Check Extract Table**

Function	Description
Rename extract tables	Select **Functions** > **Rename Table** You are prompted to write a new name for the extract table. When you rename an extract table, you are prompted if the related extract table specified in the **Extraction Setups** window also should be updated with the new name. The relation between the extraction setup and the extract table is broken if you do not update the related extract table.
View the extract table	Select **Functions** > **View Extracts** This opens the data extraction in the **Extracts Viewer** window.
View the objects in which the extract table is used	Select **File** > **Used In** This shows you which extraction setups use the extract table.
Drop extract tables	Select **Functions** > **Drop Table** This deletes the extract table.
Recreate the extract tables marked as deleted	After an extract table is dropped: Select **Functions** > **Recreate Extract Table** This recreates the dropped extract table.
Delete extract tables	After an extract table is dropped: Select **File** > **Delete** This deletes the registration of the extract table, and it cannot then be recreated.
Delete the registration of an extract table/field which is marked as deleted	If you remove a destination field in an extraction definition and in the **Extraction Setups** window select **Functions** > **Check Extract Table**, the field will be deleted from the extract table and marked as deleted in the **Extract Tables** window. To delete the registration of the field in the extract table, right-click the field in the **Extract Tables** window and select **Delete Row**.
View the indexes of extract tables	The **Indexes** tab shows information about the indexes in the selected extract table. The mandatory indexes cannot be created, updated or deleted.
Create, edit and delete user-defined indexes	On the **Indexes** tab, you can create, modify and delete customized indexes for extract tables in internal user schemas. In the **Position** field, specify the field position for compound indexes (default value for simple indexes is 1).

Function	Description
View table constraints and indexes	On the **Constraints** tab, you can view all constraints and indexes that are available. The indexes and constraints for tables in data warehouse schemas are not registered.
View data in data warehouse schemas	On the **Data** tab, you can view the data from a table in data warehouse schemas.

4.2 Manage extract tables in Data Warehouse schemas

You can use the **Extract Tables** window to create and manage extract tables in Data Warehouse schemas.

For details, see:

- Extract tables in Data Warehouse schemas explained below
- Create and edit extract tables in Data Warehouse schemas on the next page

4.2.1 Extract tables in Data Warehouse schemas explained

You can use the **Extract Tables** window in SimCorp Dimension to view and manage the extract tables related to the Data Extractor for **Internal extract user schemas** and **Data Warehouse schemas**.

Note

You must register the **Data Warehouse schema** in SimCorp Dimension before you can create and manage extract tables.

For extract tables in a **Data Warehouse schema**, you can use the **Extract Table** window to:

- Create and maintain extract tables directly without involving a Database Administrator (for example).
- Design a table layout on the **Fields** tab. Use the tab to:
  - Specify fields in the extract table. You can select fields from the data dictionary browser.
  - Specify a surrogate keyto substitute the primary key (in the**Sequence** field).
- Import a table layout from an extraction definition defined in the **Extraction Definitions** window.
- Specify constraints for extract tables on the **Constraints** tab. The following constraints types are available:
  - Check constraint
  - Foreign key

  - Primary key
- OntheDatatab, view the extracted data for extract tables.
  - Unique

For more details, see Create and edit extract tables in Data Warehouse schemas below.

4.2.2 Create and edit extract tables in Data Warehouse schemas

Create and manage extract table in the **Extract Tables** window:

- Select **File** > **New** to create a new extract table in a Data Warehouse schema.
- Select **File** > **Save** to save the changes to an extract table in a Data Warehouse schema. The extract table is validated.
- Select **File** > **Copy** to copy the extract table including all table information.

Design an extract table layout on the **Fields** tab in the **Extract Tables** window. You can specify the following on the **Fields** tab:

- **Field order**
- **Field name** (must be a valid SQL name)
- **Description**
- **Data type**
- **Field length**
- **Decimals**
- **Sequence**
  - Specify a surrogate key to an extract table with a running number that is assigned at runtime (based on Oracle SEQUENCE). Insert the surrogate key as a field on the **Fields** tab.
  - Surrogate key is a substitution for the primary key and is a unique identifier for each row that can be used as a primary key to the table.
  - Sequence field is available for fields with a NUMERIC data type.
- Deleted (indicates if the current field is not available in the extract table)

Select fields for a data dictionary browser (not available for extract tables):

- Select **Functions** > **Select field via data dictionary browser**.

Import table layout from extraction definition:

- Select **Functions** > **Load Layout From Extraction Definition**. You cannot load a table layout from extraction definitions with the **Incomplete definition** check box selected.

Create physical table in the database:

- Select **Functions** > **Recreate Extract Table**.

Edit existing tables in a Data Warehouse schema using an external tool:

- If the table exist in the database, but not registered in SimCorp Dimension: Open the **Database User Schemas Contents** window to register the table.
- If the table is registered but does not exist in the database, the table is marked as **Deleted**. Select **Functions** > **Recreate Extract Table**.

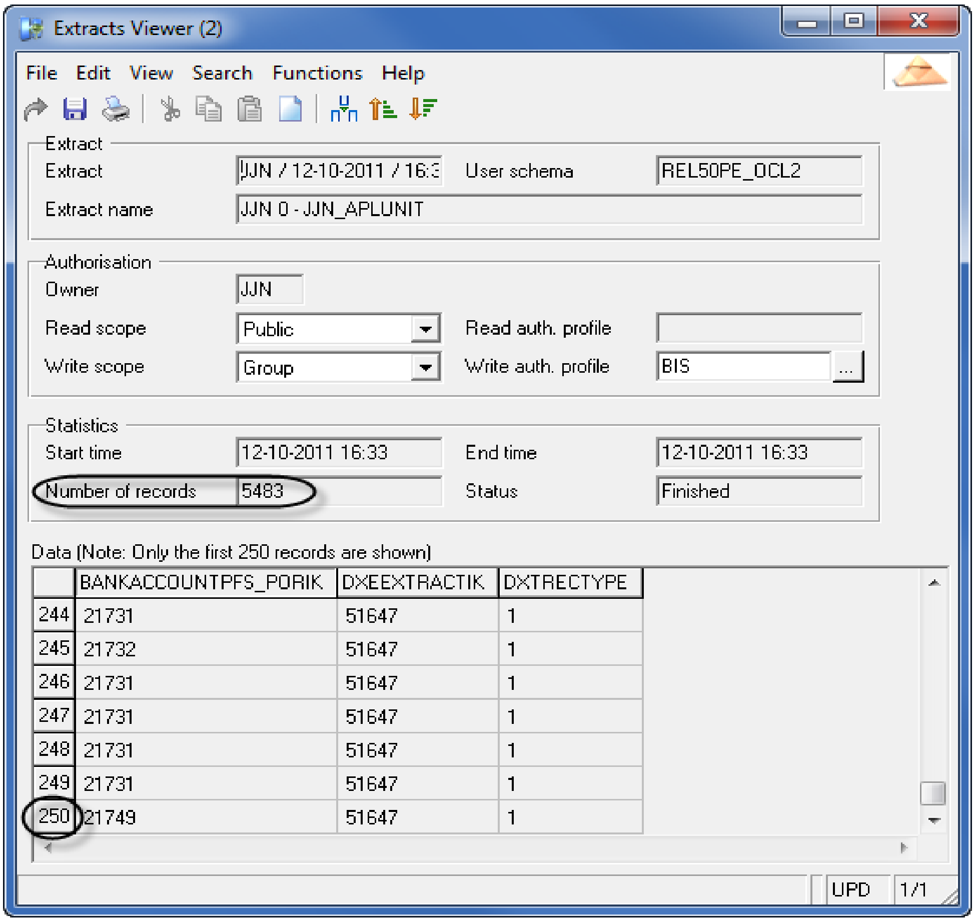
Specify constraints for the extract table on the **Constraints** tab. The following types of constraints are available:

- Check constraint
- Foreign key
- Primary key
- Unique

4.3 View the data extract

After you execute the data extraction, you can view it in the **Extracts Viewer** window.

You can access the **Extracts Viewer** window from the SimCorp Dimension portal, or from the **Extraction Setups** window **Functions** > **View Extracts**.



Extract

Extract: JJN / 12-10-2011 / 16:3

User schema: REL50PE_OCL2

Extract name: JJN 0 - JJN_APLUNIT

Authorisation

Owner: JJN

Read scope: Public

Read auth. profile:

Write scope: Group

Write auth. profile: BIS

Statistics

Start time: 12-10-2011 16:33

End time: 12-10-2011 16:33

Number of records: 5483

Status: Finished

Data (Note: Only the first 250 records are shown)

	BANKACCOUNTPFS_PORIK	DXEEXTRACTIK	DXTRECTYPE
244	21731	51647	1
245	21732	51647	1
246	21731	51647	1
247	21731	51647	1
248	21731	51647	1
249	21731	51647	1
250	21749	51647	1

UPD 1/1

View the data

Only the first 250 records appear in the window.

The actual number of records is shown in the **Statistics** section.

To view all of the records, select **Functions** > **Show All**.

Note

External user schemas are not supported by the viewer.

In the **Data** section, you see the data extraction as defined. There are two additional columns:

- DXEXTRACTIK — An internal key which is unique for every extract
- DXTRECTYPE — A numerical representation of the type of extract which will normally be '1'

Authorisation rights

You must have authorisation to view the data extraction. If you own the data extraction, you can edit the **Read scope** and **Write scope** for the data extraction. These fields are identical to the fields in the **Extraction Setups** window, but in the **Extracts Viewer** they only apply to the specific data extract, whereas in the **Extraction Setups** window they apply to all data extractions executed with that particular extraction setup.

Delete the data extraction

Delete the data extraction by selecting **File** > **Delete**. See Delete data extractions manually for more information.

5 Monitoring validity of data extractor objects

Use the **Extraction Validator** window to check all extraction objects in an installation for their validity.

The **Extraction Validator** window validates the following object types:

- Database User Schemas Administration
- Extract Tables
- Extraction Definitions
- Advanced Extraction Definitions
- Extraction Setups
- Extracts Exporter Definitions
- Extracts Viewer

This section contains the following topics:

- When to validate
- Validate data extractor objects manually
- Validate data extractor objects with batch jobs

5.1 When to validate

It is important to validate your extractor objects, particularly when you are migrating or upgrading your system.

More so, it is important to monitor validation if you work in a dynamic installation with many dependencies and where extraction objects are often changed.

Use the **Extraction Validator** window to find errors in the setups and broken links between extraction objects without running the extraction.

Scenario

You have prepared twenty extractions, which are based on plug-ins and formulas, and you have applied segment conditions and other search conditions. You then save the output into an extract table in different external user schemas. The extraction becomes invalid if an external application has dropped a column from the table you prepared, or if the password to the external user schema has been changed, or if somebody has added a new parameter to a formula without adjusting the mapping of destination fields to that formula, etc.

It can be a tedious task to check these dependencies manually. The **Extraction Validator** window does this with only a few mouse clicks.

5.2 Validate data extractor objects manually

The **Extraction Validator** has the following search criteria:

- Validation state

- Object
- Object name
- Object date
- Status
- Validated date
- Object type

Validate

To query for the required objects and test their validity, do the following.

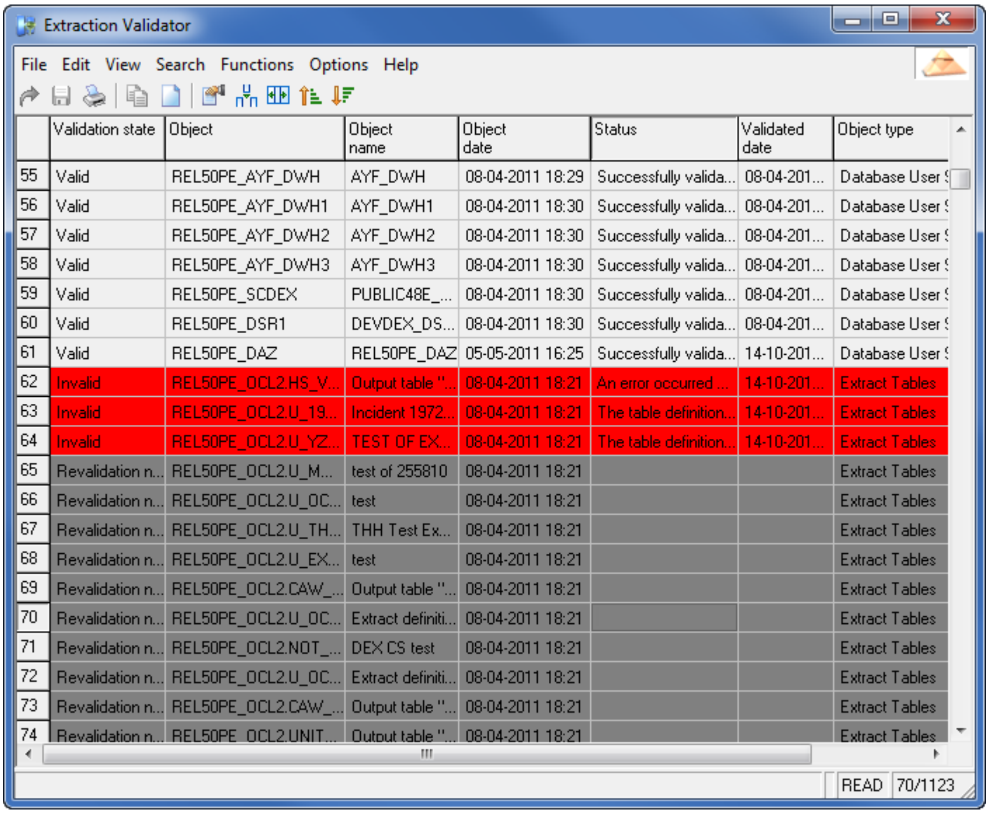
1. Search for objects in the **Extraction Validator** window, by entering the required search criteria in a field and then pressing ENTER.

   You can use wildcards (%) in your queries.

   The relevant objects then appear in the window.

2. Run a check on the extraction objects that are retrieved. There are two approaches:
   - All extraction objects at the same time — Select **Functions** > **Validation On All Records**.
   - Limit the check to a specific selection — Select the rows of the objects that you want to validate, right-click and select **Validation On Selection**.

The result of the validation appears in the window.



	Validation state	Object	Object name	Object date	Status	Validated date	Object type
55	Valid	REL50PE_AYF_DWH	AYF_DWH	08-04-2011 18:29	Successfully valida...	08-04-201...	Database User
56	Valid	REL50PE_AYF_DWH1	AYF_DWH1	08-04-2011 18:30	Successfully valida...	08-04-201...	Database User
57	Valid	REL50PE_AYF_DWH2	AYF_DWH2	08-04-2011 18:30	Successfully valida...	08-04-201...	Database User
58	Valid	REL50PE_AYF_DWH3	AYF_DWH3	08-04-2011 18:30	Successfully valida...	08-04-201...	Database User
59	Valid	REL50PE_SCDEX	PUBLIC48E_...	08-04-2011 18:30	Successfully valida...	08-04-201...	Database User
60	Valid	REL50PE_DSR1	DEVDEX_DS...	08-04-2011 18:30	Successfully valida...	08-04-201...	Database User
61	Valid	REL50PE_DAZ	REL50PE_DAZ	05-05-2011 16:25	Successfully valida...	14-10-201...	Database User
62	Invalid	REL50PE_OCL2.HS_V...	Output table "...	08-04-2011 18:21	An error occurred ...	14-10-201...	Extract Tables
63	Invalid	REL50PE_OCL2.U_19...	Incident 1972...	08-04-2011 18:21	The table definition...	14-10-201...	Extract Tables
64	Invalid	REL50PE_OCL2.U_YZ...	TEST OF EX...	08-04-2011 18:21	The table definition...	14-10-201...	Extract Tables
65	Revalidation n...	REL50PE_OCL2.U_M...	test of 255810	08-04-2011 18:21			Extract Tables
66	Revalidation n...	REL50PE_OCL2.U_OC...	test	08-04-2011 18:21			Extract Tables
67	Revalidation n...	REL50PE_OCL2.U_TH...	THH Test Ex...	08-04-2011 18:21			Extract Tables
68	Revalidation n...	REL50PE_OCL2.U_EX...	test	08-04-2011 18:21			Extract Tables
69	Revalidation n...	REL50PE_OCL2.CAW_...	Output table "...	08-04-2011 18:21			Extract Tables
70	Revalidation n...	REL50PE_OCL2.U_OC...	Extract definiti...	08-04-2011 18:21			Extract Tables
71	Revalidation n...	REL50PE_OCL2.NOT_...	DEX CS test	08-04-2011 18:21			Extract Tables
72	Revalidation n...	REL50PE_OCL2.U_OC...	Extract definiti...	08-04-2011 18:21			Extract Tables
73	Revalidation n...	REL50PE_OCL2.CAW_...	Output table "...	08-04-2011 18:21			Extract Tables
74	Revalidation n...	REL50PE_OCL2.UNIT...	Output table "...	08-04-2011 18:21			Extract Tables

Autocorrect

When you validate, you can automatically correct inconsistencies between the source and target field in extraction definitions and extraction setups (including the extract tables), when the underlying objects have been changed in the SimCorp Dimension database.

You can, for example, use the functionality in connection with upgrades to automatically correct the extraction definitions and extraction setups.

Select amongst the following in the **Options** menu:

- **Autocorrect Extraction Definitions** — Automatically correct inconsistencies between source and target fields in extraction definitions.
- **Autocorrect Extraction Setups** — Automatically correct inconsistencies between source and target fields in extraction setups.
- **Interactive validation** — Suppress popup during validation. If you select this option, then you must response to popup windows during validation. If you deselect this option, then pop-ups are suppressed and validation goes to silent mode.

5.3 Validate data extractor objects with batch jobs

To validate data extractor objects with batch jobs, do the following.

1. Create a batch job, and in the **Window/Function** field select **Extraction Validator – Validate Data Extractor Objects**.
2. In the **Object** field, select one of the following:
   - Database User Schemas
   - Extract Tables
   - Extraction Definitions
   - Extraction Setups
   - Extracts Exporter Definitions
   - Extracts Viewer
3. Select whether or not the validation should automatically correct inconsistencies between source and target fields.
4. Execute the batch job in a batch job group.

6 Cleaning up data extractions and data extractor objects

The data extractor produces two types of data that can accumulate and obstruct your system:

- Redundant data extractions
- Redundant data extractor objects

This chapter describes how to clean up both types of data.

- When to clean up data extractions and data extractor objects
- Delete data extractions manually
- Define redundancy of data extractions
- Delete data extractions with Database User Schema Contents
- Delete data extractor objects with Cleanup Data Extractor
- Create batch jobs for cleanup

6.1 When to clean up data extractions and data extractor objects

When you are cleaning up your system for redundant data related to data extractions, there are two types of data that can be redundant: the data extractions themselves, and the data extractor objects that are created to extract the data.

Clean up your data extractions according to your business model. You can clean up your data extractions in the following manner:

- Delete data extractions manually

or

- Define redundancy of data extractions, followed by one of the following:
    - Delete data extractions with Database User Schema Contents
    - Delete data extractor objects with Cleanup Data Extractor
    - Create batch jobs for cleanup

Clean up data extractor objects to reduce clutter in your environment. The process of cleaning up redundant data extractor objects is described in Delete data extractor objects with Cleanup Data Extractor.

Note

These methods are only relevant for internal user schemas. External user schemas cannot be cleaned up using SimCorp Dimension. This should be done using Oracle or 3rd party applications such as Toad.

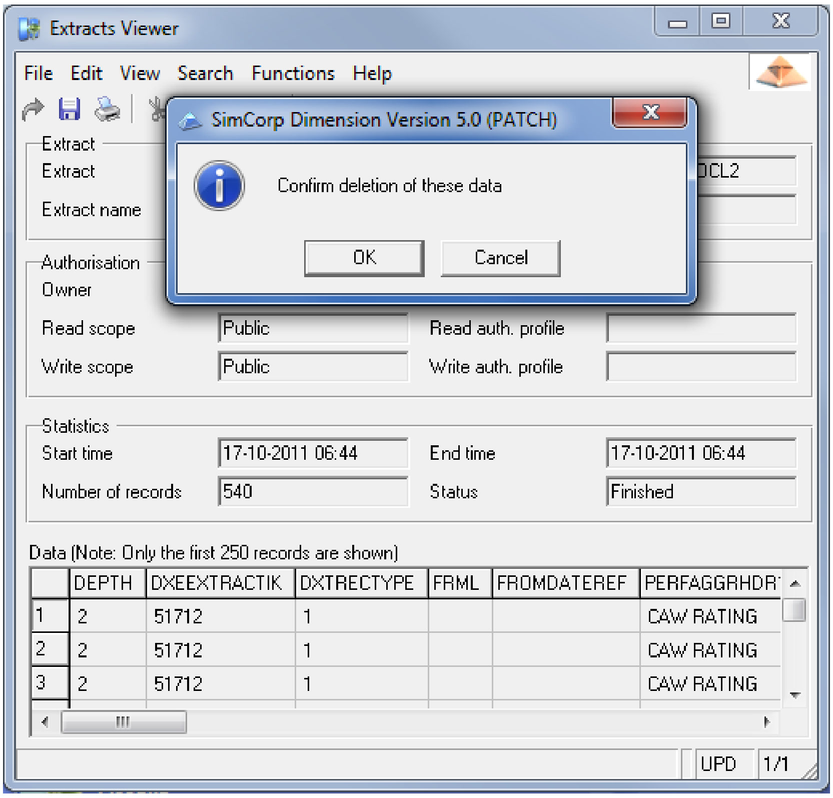
6.2 Delete data extractions manually

Use the **Extracts Viewer** to select data extractions that are no longer required, and then select **File** > **Delete**.

This method is useful when you want to pinpoint an exact data extraction to be deleted.

You must have **Write scope** authorisation for the data extraction in order to delete it.This is in contrast to the other cleanup methods, where only data extractions that have been enabled for automatic cleanup can be deleted.

For example:


Extracts Viewer
File Edit View Search Functions Help
SimCorp Dimension Version 5.0 (PATCH)
OK
Cancel
UPD
1/1

Confirm deletion of these data

Extract

Extract DCL2

Extract name

Authorisation

Owner

Read scope Public Read auth. profile

Write scope Public Write auth. profile

Statistics

Start time 17-10-2011 06:44 End time 17-10-2011 06:44

Number of records 540 Status Finished

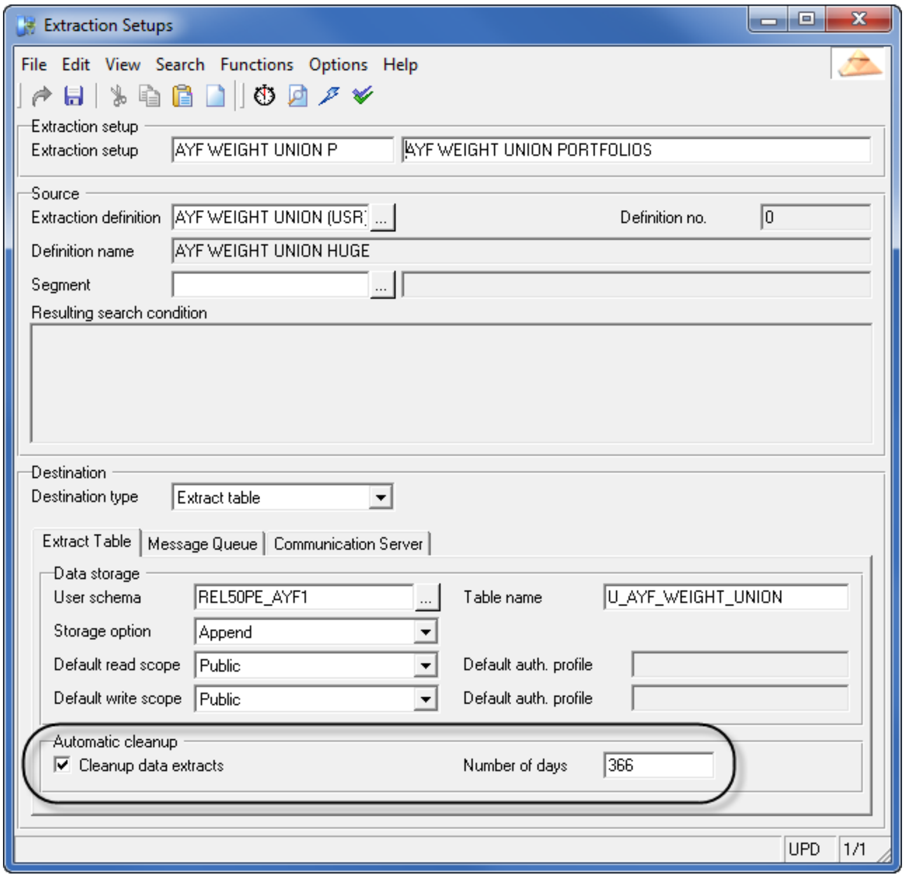
Data (Note: Only the first 250 records are shown)

	DEPTH	DXEEXTRACTIK	DXTRECTYPE	FRML	FROMDATEREF	PERFAGGRHDR
1	2	51712	1			CAW RATING
2	2	51712	1			CAW RATING
3	2	51712	1			CAW RATING

6.3 Define redundancy of data extractions

Use the **Extraction Setup** window to define how many days a data extraction can stay in the system before being labelled as outdated. When the data extraction is executed, it is made and marked for cleanup according to the number of days defined.

For example:



Extraction setup

Extraction setup	AYF WEIGHT UNION P	AYF WEIGHT UNION PORTFOLIOS

Source

Extraction definition	AYF WEIGHT UNION (USR)	Definition no.	0
Definition name	AYF WEIGHT UNION HUGE		
Segment			

Resulting search condition

Destination

Destination type: Extract table

Extract Table | Message Queue | Communication Server

Data storage

User schema	REL50PE_AYF1	Table name	U_AYF_WEIGHT_UNION
Storage option	Append		
Default read scope	Public	Default auth. profile	
Default write scope	Public	Default auth. profile	

Automatic cleanup

☑ Cleanup data extracts Number of days 366

When the number of days has passed, the data extraction is internally labelled as outdated and ready for cleanup using one of the following methods:

- Delete data extractions with Database User Schema Contents
- Delete data extractor objects with Cleanup Data Extractor
- Create batch jobs for cleanup

6.4 Delete data extractions with Database User Schema Contents

When you create an extraction setup, you can define how many days a data extraction can stay in the system before being labelled as outdated (see Define redundancy of data extractions).

When this time period has passed, the data extraction is internally labelled as outdated and ready for cleanup.

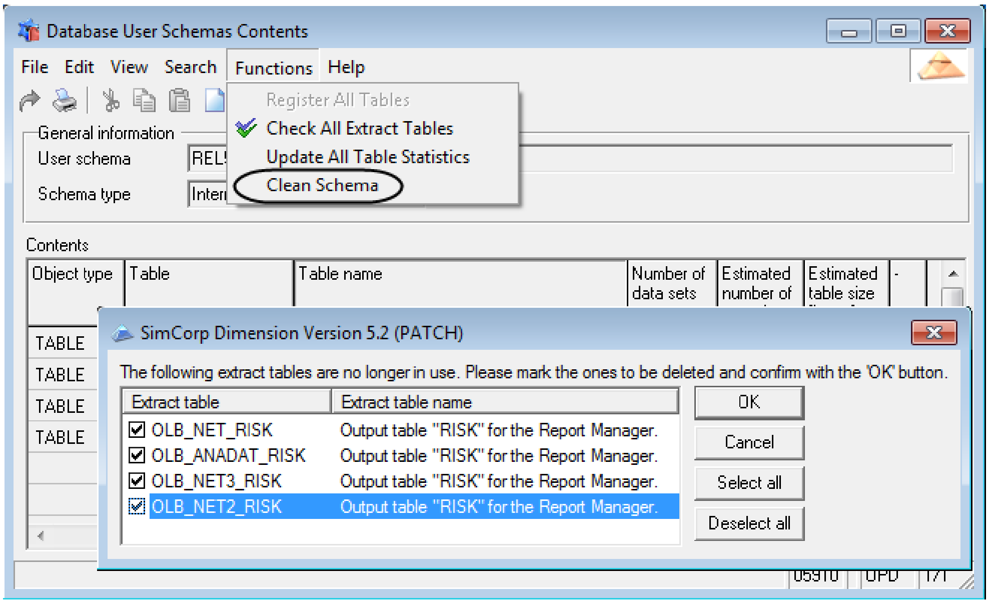
Use the **Database User Schemas Contents** window to select all outdated data extractions in a particular user schema and delete them.

In the **Database User Schemas Contents** window, select the user schema for which you want to delete the outdated data extractions, and select **Functions** > **Clean Schema**.

A list of outdated extract tables in the user schema for which you have Write scope authorisation appears.

Select the extract tables to delete, and click **OK**.

For example:


Database User Schemas Contents
File Edit View Search Functions Help
Register All Tables
Check All Extract Tables
Update All Table Statistics
Clean Schema
General information
User schema REL!
Schema type Inter
Contents
Object type Table Table name Number of data sets Estimated number of Estimated table size
TABLE
TABLE
TABLE
TABLE
SimCorp Dimension Version 5.2 (PATCH)
OK
Cancel
Select all
Deselect all

The following extract tables are no longer in use. Please mark the ones to be deleted and confirm with the 'OK' button.

Extract table	Extract table name
☑ OLB_NET_RISK	Output table "RISK" for the Report Manager.
☑ OLB_ANADAT_RISK	Output table "RISK" for the Report Manager.
☑ OLB_NET3_RISK	Output table "RISK" for the Report Manager.
☑ OLB_NET2_RISK	Output table "RISK" for the Report Manager.

Cleaning up data extractions can also be done through the **Cleanup Data Extractor** window, however in that window the data extractions are deleted for an extraction setup, as opposed to a user schema (see Delete data extractor objects with Cleanup Data Extractor).

6.5 Delete data extractor objects with Cleanup Data Extractor

Use the **Cleanup Data Extractor** window to delete the data extractor object types and/or data extractions that are marked for cleanup from your system.

6.5.1 When to use the Cleanup Data Extractor

The **Cleanup Data Extractor** window is used to delete unreferenced data extractor objects from the system. Unreferenced objects are objects that have no functional relations within the system. The tool searches for unreferenced objects according to the selected application window.

In order to avoid that the cleanup creates unreferenced objects, it is recommended that the cleanup is done in the following order:

1. Extracts Exporter Definitions
2. Extraction Setups
3. Extraction Definitions

Note

Do not delete any objects with dynamic references. The **Cleanup Data Extractor** does not identify whether objects are used in, for example the Communication Server. Nor does it identify extraction setups with references to extract tables through extraction definitions.

Object type	Search criteria
Extraction definitions	**Unreferenced objects** — Extraction definitions that are not referenced in an extraction setup.
Extraction setups	**Unreferenced objects** — Extraction setups that are not referenced in an extracts exporter definition, batch job, online job, auxiliary job, extract viewer, or database user schemas administration. **Extracts marked for cleanup** — All expired data extractions according to definition in the **Number of days** field in the **Extraction Setups** window.
Extracts exporter definitions	**Unreferenced objects** — Extracts exporter definitions that are not referenced in a batch job, online job, or extract viewer.
Extracts viewer	• **Extract registration – Cancelled by user** — Extract registration with status Cancelled by user. • **Extract registration – Currently running** — Extract registration with status Currently running. • **Extract registration – Deleted** — Extract registration with status Deleted. • **Extract registration – Failed** — Extract registration with status Failed. • **Extract registration – Finished** — Extract registration with status Finished. Only data extractions that are not extracted on extract tables are shown in the grid, for example extract registrations created via extraction exporter definition or communication server. Extract registrations associated to extract tables cannot be cleaned alone. They must be cleaned together with data content in the extract viewer. Data extractions that were started less than seven days prior are not shown in the grid and therefore cannot be cleaned. **Extract registration – (none)** — Extract registration with status (none).

Note

The **Cleanup Data Extractor** window deletes all information about the data extractions from database. Data extractions shown in the grid are those with public **Write scope** authorisation or the current user has **Write scope** authorisation.

6.5.2 How to use the Cleanup Data Extractor

Use the **Cleanup Data Extractor** window to delete several data extractor objects at once.

1. In the **Cleanup Data Extractor** window, select an **Object Type** and **Search criteria**.
2. If required, filter the data extractor objects by specifying a **From date** and **To date**.
3. Click **Search** to display all the objects that match the search criteria.
4. Select the desired objects from the grid individually, or click **Select All**.
5. Click **Cleanup**. The selected objects are deleted.

6.6 Create batch jobs for cleanup

You can set up cleanup jobs to run automatically at specific times to clean up extract registrations or data extractions that are marked for cleanup. This is done by creating batch jobs.

1. In the **Batch Jobs** window, enter an ID and name.
2. In the **Window/function** field, select one of the following:
   A. **Cleanup Data Extractor – Extract Registration** — to have an automatic cleanup of extract registration.
      I. Select the extract status for the extract registration.
      II. If required, filter the extract registrations by specifying a from date and to date.

   B. **Cleanup Data Extractor – Extracts Marked for Cleanup** — to have an automatic cleanup of expired extracts.
      I. Select the extraction setup that defines the data extraction.
3. Save the batch job, and add it to a batch job group to execute.

For more information about creating batch jobs, see the ***Batch Jobs*** user manual.

Having properly set up the batch job, the outdated data extractions will be deleted every time the batch job is run.

This cleanup process is useful when large amounts of data are extracted on a regular basis, because it can be set up once and for all, leaving a minimal amount of manual maintenance.

Index

ABOUT SIMCORP