

Database mapping and definition files

This appendix includes the following topics:

- [About database mapping and definition files](#)
- [About the db.xml file](#)
- [About the def.xml file](#)
- [About the tree.xml file](#)

About database mapping and definition files

An understanding of database mapping and definition files is essential if you want to add internal proprietary information such as, user name and telephone number, to the reports generated by Report Manager.



Entering invalid strings into these files can cause the customized reporting mechanism to fail. Any modifications made to the xml files falls under the sole responsibility of the customer. Before modifying these files verify that you are experienced with the SQL language, familiar with XML, and that you have fully read and understood this section.

The database mapping and definitions for customized reports are stored in the following files:

- **db.xml.** Describes the database mapping.
- **def.xml.** Defines dependency conditions.
- **tree.xml.** Defines the hierarchical structure of the Data Tree.

These files are found located in: `<precise_root>/products/foresight/etc/reports/udr`

About the db.xml file

The db.xml file defines all Database Tables and Join conditions between tables, which are used by the customized reporting mechanism.

The db.xml file uses the following tags to define the database mapping:

- [About the Table tag](#)
- [About the Join tag](#)
- [About the Related tag](#)

About the Table tag

The Table tag defines the table and its attributes, as described in the following table. The Table tag is a top level element in the db.xml file.

Example:

```
<table name="PS_INEN_ENVIRONMENT" type="lookup"/>
<table name="PW_SPTS_TRAN_STAT_$sumlvl" type="statistics" summary_levels="T,H,D,W,M" >
</table>
```

The following table describes the table tag attributes.

Table 1 Table tag attributes

Attribute	Description
name	Specifies the table name as defined in the database schema. For a statistics table type, you can use the \$sumlvl variable. In this case, you must set also the summary_levels attribute. When executing the customized report, the \$sumlvl variable uses the value that you set in the summary_levels attribute.
type	Defines the table type, as follows: • Lookup. A lookup table maps between entity identifiers and entity names. • Statistics. A statistics table that includes statistics performance information and is summarized into various time intervals (such as, hourly, daily, and weekly). • History. History table that includes statistics performance information, but is not summarized into various time intervals.
summary_levels	This attribute is only relevant for a statistics table type. It lists the available summary levels for this table. These values include: • T. Time slice • H. Hourly • D. Daily • W. Weekly • M. Monthly

time_field	Defines the timestamp column name. Use this attribute only if the column name is different than the standard xxxx_timestamp column name, where xxxx indicates the name of the table (abbreviated).
noteable_name	A flag that enables or disables Join conditions. Set this flag to true for tables that contain Join conditions written in Java code. In this case, replace the table-name with the table name whose Join conditions you want to disable. For example, noPS_INSM_SERVER_MACHINE="true", will disable the Join conditions in table PS_INSM_SERVER_MACHINE.
alias	Defines an alias name for a table that is used frequently. For example, you can use the same lookup table again for different purposes.

About the Join tag

The Join tag is a second level element in the db.xml file, and is part of the table definition. The Join tag defines Join conditions between the current table and other specified tables. The Join tag defines a one-way connection from the current table to another table, but not in the opposite direction. Usually, lookup tables do not include a Join definition. Instead, they will be referenced inside a Join definition of Statistics Tables. You should not define a Join between two Statistics Tables, unless it is a One-to-one Join. The Join predicates are included in the tag contents.

Example:

```
<table name="PS_INCE_INSTANCE" type="lookup">
  <join toTable="PS_INEN_ENVIRONMENT" otherTables="PS_INAP_APP_TIER">
    INEN_ID = INAP_INEN_ID AND INCE_INAP_ID = INAP_ID
  </join>
</table>
```

The following table describes the Join tag attributes.

Table 2 Join tag attributes

Attribute	Description
toTable	Specifies the table that is joined with the current table.
otherTables	Specifies the name of a connection table. This attribute is used if the Join is performed through another table.
normalization_field	Defines a Join condition for a Lookup Table, whose name ends with _N. You can use the normalization_field attribute to specify the name of the normalization column. In this case, the select will include a 'group by' with the normalization_field column and will be wrapped by another select, which joins with the Lookup Table specified in the toTable attribute. The reasons for applying this method are: • Performance. The Join with the Lookup Table is done only in the wrapping select and not in the inner select which retrieves more rows. • Database limitations. In Oracle it is impossible to apply any option on a column of a long data type, or to use it in the 'group by'.

About the Related tag

The Related tag is a second level element in the db.xml file, and is part of the table definition. This tag is only relevant for lookup tables. It defines the statistics table used for retrieving entity identifiers that depend on a specified condition (see the depend attribute).

A table can include more than one Related tag. Report Manager uses the first related table whose dependency condition is met. It is possible to define the last line without a depend attribute, so that if no previous dependency was met, this table is always selected. If you want an error message to be displayed when no dependency conditions are met, do not specify a related table with no dependency.

The Performance Management Database includes Snapshot Tables (for details about these tables, see the Performance Management Database documentation). To achieve optimal performance when using Snapshot Tables, it is recommended to define the smallest Snapshot Tables first, followed by the larger ones.

For example:

```
<table name="PW_SPLD_LOCALE_DIM" type="lookup">
  <related table="PW_SPTS_TRAN_STAT_LS_$sumlvl" depend="loc srv snapshot"/>
  <related table="PW_SPTS_TRAN_STAT_AOLS_$sumlvl" depend="org app loc srv snapshot"/>
</table>
```

The following table describes the Related tag attributes.

Table 3 Related tag attributes

Attribute	Description
table	Defines the name of the Statistics Table.
depend	Specifies the dependency condition identifier. This identifier is defined in the def.xml file. If the dependency condition is met, then the table specified in the table attribute is used to retrieve the list of entity identifiers.

About the def.xml file

The def.xml file is a configuration file that defines dependency conditions using a single tag called dependency. These definitions are used by the depend attribute of the db.xml file.

A special subset of these definitions are dependency conditions for Snapshot Tables. A Snapshot Table is used if all the selected data fields in the query exist in this snapshot.

The dependency tag contains a single attribute, called name. The value of this attribute is a character string that identifies the dependency. This character string must be unique in the def.xml file.

The dependency tag format contains dependency conditions, as follows:

```
<![CDATA[unary-operator{field-id,...}] logical-operator ...>
```

where:

- The field-id specified in the brackets is a numeric identifier of a data field in the Data Tree as defined in the tree.xml file. You can specify one or more data field identifiers inside the brackets according to the specified unary-operator.
- The unary-operator defines the relation between the data field identifiers specified in brackets and the SELECT list in the query. The value can be one of the following:
 - **a.** All. All the data field identifiers must exist in the SELECT list.
 - **na.** Not all. All of the data field identifiers must not exist in the SELECT list.
 - **o.** One. At least one data field identifier must exist in the SELECT list.
 - **no.** Not one. At least one data field identifier must not exist in the SELECT list.
- The logical-operator between dependency conditions can be one of the following:
 - **|.** OR
 - **&.** AND

Example:

```
<dependency name="form appl user snapshot and user"><![CDATA[na{2003,4} &
a{2002}]]></dependency>
<dependency name="oa user"><![CDATA[a{2002}]]></dependency>
```

About the tree.xml file

The tree.xml file defines the hierarchical structure of the Data Tree. Each data field that is a leaf node, includes a unique numeric identifier. The data field identifiers are used in the def.xml file. For this reason you should not modify the existing data field identifiers, but you can add additional data field identifiers to the Data Tree. If you add new data field identifiers to the Data Tree, you must also add new data fields to the def.xml file.

The tree.xml file uses the following tags:

- Group
- Object
- Objref
- Alt

About the Group tag

This tag defines a group that is a non-leaf node. A group can contain data fields (leaf nodes) and other groups. Example:

```
<group name="General" desc="0">
<group isApptier="true" tech="SP" name="SP" desc="17" />
</group>
<group isApptier="true" tech="ST" installCode="SQ-SQCOL" name="ST" desc="1076" />
```

The following table describes the Group tag attributes.

Table 4 Group tag attributes

Attribute	Description
name	Defines a string representing the group name. Report Manager translates this string to a text defined in the udrTxtResource.xml resource file, and then displays the translated text in the Data Tree.
desc	Defines a unique numeric identifier that represents the description of the group. Report Manager translates this number into text defined in the udrTxtResource.xml resource file. Before adding a new identifier to the file, you must verify that this number is not already in use. (The numeric identifier at the end of the file list is not necessarily the highest one.)
isApptier	Defines whether the group is an Tier group or not.
tech	Defines a string that represents the technology of the Tier group. Report Manager translates this string into text defined in the udrTxtResource.xml resource file.

installCode	Defines a string that represents the installation code. Its format is a combination of the product code and init code. For example, IS-FP, where IS is the product code and FP is the init code.
-------------	--

Report Manager validates the installation code using the Infrastructure Installation Table named PS_INLL_INSTALLATION, to determine whether or not to display this group in the specific Precise application. (The Infrastructure Installation Table contains information regarding the application's system installations.)

About the Object tag

This tag defines a data field that is displayed as a leaf node in the Data Tree. It can be a column from a table or a formula.

Example:

```
<object id="1" type="id" data_type="string" name="Environment" desc="1">
<object id="1000" type="instance_id" name="SPSystem" desc="20">
<object id="21" type="time" data_type="time" xls_format="m/d/yy h:mm" name="Timestamp" desc="7">
```

The following table describes the Object tag attributes.

Table 5 Object tag attributes

Attribute	Description
id	Defines a numeric identifier of the data field.
type	Defines a string that represents the type of object. The value can be one of the following: id. Identifier data field time. Time data field instance_id. Instance data field cnt. Counter data field (in this case, also defined as an Aggregate function)
data_type	Describes the type of each data field. Default values include: - id and instance_id - string - cnt - double - time - time It is possible to override the default type of the data field type by defining the data_type attribute. Possible values include: string, time, int, and float.
xls_format	The format of each type of data field has a default value used by Microsoft Excel. By default, Microsoft Excel uses no special format for strings. For numeric data types, Microsoft Excel uses the floating decimal point format. For date/time data types, Microsoft Excel uses the numeric format. You can override the default format by defining the xls_format attribute. For example, you can override the format of the date/time data type with one of the following formats: m/d/yy, d-mmm-yy, d-mmm, mmm-yy, h:mm AM/PM, h:mm:ss AM/PM, h:mm, h:mm:ss, and m/d/yy h:mm.
name	Defines a string that represents the data field name. Report Manager translates this string into a text defined in the udrTxtResource.xml resource file, and then displays the translated text in the Data Tree.
desc	Defines a unique numeric identifier that represents the description of the data field. Report Manager translates this number into a text defined in the udrTxtResource.xml resource file. Before adding a new identifier to the file, you must verify that this number is not already in use. (The numeric identifier at the end of the file list is not necessarily the highest one.)
installApp	Defines a string that represents a Tier code. A specific data field may depend on a specific Tier, so if the Tier is not installed in the current application, the data field will not be displayed in the Data Tree. The Tier installation is verified against the Infrastructure Tier Table named PS_INAP_APP_TIER.

installCode	Defines a string that represents the installation code. Its format is a combination of the product code and init code. For example, IS-FP, where IS is the product code and FP is the init code. Report Manager validates the installation code using the Infrastructure Installation Table named PS_INLL_INSTALLATION, to determine whether or not to display this group in the specific Precise application. (The Infrastructure Installation Table contains information regarding the application's system installations.)
includeQ	Defines a new query used if the object returns no data (in this case, the data field is not displayed). For example: <pre><object id="4026" type="id" name="QSiebelView" desc="446" includeQ="SELECT INLI_INCE_ID FROM PS_INLI_LITE_INSTANCE, PS_FSEC_ENV_CONTENTS WHERE INLI_INTE_CODE = 'SQ' AND INLI_INCE_ID = FSEC_INSTANCE_ID AND INLI_DELETED = 'F' AND FSEC_ENV_ID = ?"></pre>

About the Objref tag

This tag defines a reference to another Object that is already defined in the Data Tree definition. Use the Objref tag if the data field you want to add to the Data Tree has the same attributes as an existing object (instead of repeating the entire definition of the object).

For example, the Instance ID object (data field) may appear under various groups of a Tier.

The following example creates a reference to an object 3 (Server) definition:

```
<objref id="3" />
```

About the Alt tag

The meaning of a data field can depend on its context, that is, on the other fields that are selected with it. The Alt tag defines an alternative selection of expression for its data field, based on the other fields selected. When the data field is selected, Report Manager scans the alternatives in the same order as they were listed in the file, and chooses the first alternative whose dependency is met.

Example:

```
<object id="5050" type="cnt" name="ORInOracleTimeSum" desc="825">
  <alt select="SUM(ORSS_IN_ORACLE_TIME_SUM)" table="PW_ORSS_STATEMENTS_STATS_$sumlvl" depend="or stmtnt no sess" />
  <alt select="SUM(ORSA_IN_ORACLE_TIME_SUM)" table="PW_ORSA_STMT_APPL_STATS_$sumlvl" depend="or stmtnt and sess" />
  <altselect="SUM(ORAS_IN_ORACLE_TIME_SUM)" table="PW_ORAS_APPLICATION_STATS_$sumlvl" />
</object>
```

In this example, if the user selects the In Oracle time data field:

- If the or stmtnt no sess dependency is met (defined in the def.xml file), then Report Manager selects the SUM(ORSS_IN_ORACLE_TIME_SUM) expression from the PW_ORSS_STATEMENTS_STATS_\$sumlvl Table.
- If the first dependency is not met, and the or stmtnt and sess dependency is met (defined in the def.xml file), then Report Manager selects the SUM(ORSA_IN_ORACLE_TIME_SUM) expression from the PW_ORSA_STMT_APPL_STATS_\$sumlvl Table.
- If the second dependency is not met, Report Manager selects the SUM(ORAS_IN_ORACLE_TIME_SUM) expression from the PW_ORAS_APPLICATION_STATS_\$sumlvl Table.

The following table describes the Alt tag attributes.

Table 6 Alt tag attributes

Attribute	Description
select	Defines a table column name or an expression. For a counter (type="cnt"), an expression with an aggregated function is required. The expression can include the \$timeFld string, which Report Manager replaces with the timestamp column of the table used in the query. In this case, the table attribute must be empty.
table	Specifies the name of the table as defined in the Performance Management Database.
depend	Specifies the name of the dependency condition as defined in the def.xml file. If the specified condition is met, Report Manager uses the alternative SELECT.
otherTables	Defines a name of a connection table. If you need to join one table to other tables for this data field only, you can specify its conditions in the alt tag and it is therefore not necessary to define this Join in the db.xml file.