Cross Database Comparison
Setup Guide
TABLE OF CONTENT

1 INTRODUCTION TO CROSS DATABASE COMPARISON (CDC) .......................................................... 5
1.1 Implementation considerations ........................................................................................................... 6
1.1.1 Prerequisite ..................................................................................................................................... 6
1.1.2 Integration ....................................................................................................................................... 7
1.1.3 Features ......................................................................................................................................... 7
1.1.3.1 New features with Solution Manager 7.1 SP5 ................................................................................. 7
1.1.3.2 New features with Solution Manager 7.1 SP10 .............................................................................. 7
1.1.3.3 New features with Solution Manager 7.1 SP12 .............................................................................. 7

2 PREREQUISITES FOR CROSS DATABASE COMPARISON ................................................................. 9
2.1 Correct versions of the software components ...................................................................................... 9
2.2 Activation of services .......................................................................................................................... 9
2.3 Authorization for users ...................................................................................................................... 9
2.3.1 CDC authorization objects on Solution Manager .............................................................................. 9
2.3.2 CDC authorization objects on Managed System .......................................................................... 11
2.4 Connections to the data source system ............................................................................................. 11
2.5 Whitelists ....................................................................................................................................... 12
2.5.1 Whitelist to access the application server .................................................................................... 12
2.5.1.1 Procedure ....................................................................................................................................... 12
2.5.2 Whitelist to access the local PC ........................................................................................................... 12
2.5.2.1 Procedure ....................................................................................................................................... 12

3 SETTING UP CROSS DATABASE COMPARISON ................................................................................. 14
3.1 Starting Cross Database Comparison ................................................................................................ 14
3.2 Defining Comparison Objects ........................................................................................................... 15
3.2.1 Creating Comparison Objects ...................................................................................................... 15
3.2.1.1 Procedure ....................................................................................................................................... 15
3.2.2 Maintain comparison tables and fields ........................................................................................... 19
3.2.2.1 Procedure ....................................................................................................................................... 19
3.2.3 Maintain Foreign Key ..................................................................................................................... 21
3.2.3.1 Procedure ....................................................................................................................................... 21
3.2.4 Maintain Filters ............................................................................................................................... 22
3.2.4.1 Procedure ....................................................................................................................................... 22
3.2.5 Maintain Context Fields (Display fields) .......................................................................................... 23
3.2.6 Assign comparison table fields ....................................................................................................... 24
3.2.6.1 Procedure ....................................................................................................................................... 24
3.2.7 Generating Extractors for a Comparison Object .............................................................................. 26
3.2.7.1 Procedure ....................................................................................................................................... 26
3.2.8 Generation of Comparison Object from SAP Landscape Replication Server (SLT) repository .... 27
3.2.8.1 Procedure ....................................................................................................................................... 27
3.2.9 Change Documents for Comparison Objects .................................................................................. 27
3.2.9.1 Procedure ....................................................................................................................................... 27
3.2.10 Versioning of Comparison Object Data Models ........................................................................... 28
3.2.10.1 Procedure .................................................................................................................................... 28

3.3 Defining Comparison Instances ......................................................................................................... 28
3.3.1.1 Procedure to Create Comparison Instances ................................................................. 29
3.3.2 Change Documents for Comparison Instances ............................................................... 31
3.3.2.1 Procedure .................................................................................................................. 31
3.4 Defining Comparison Groups .............................................................................................. 31
3.4.1.1 Procedure to maintain Comparison Group ..................................................................... 31
3.5 Executing and Evaluating Comparison Runs ........................................................................ 33
3.5.1 Scheduling Comparison Runs ........................................................................................... 34
3.5.1.1 Procedure to schedule a job ...................................................................................... 34
3.5.2 Stopping Comparison Runs .............................................................................................. 35
3.5.2.1 Procedure to stop a Comparison Run ......................................................................... 35
3.5.3 Display Comparison Results ............................................................................................. 35
3.5.3.1 Procedure ................................................................................................................ 36
3.5.4 Deleting Comparison Results .......................................................................................... 38
3.5.4.1 Procedure ................................................................................................................ 38
4 FILTERING IN CROSS DATABASE COMPARISON ................................................................. 39
4.1.1 Example for filter type 1 .................................................................................................. 39
4.1.2 Example for filter type 2 ............................................................................................... 40
4.1.3 Example for filter type 3 ............................................................................................... 40
4.1.4 Example for filter type 4 ............................................................................................... 41
4.2 Relative Date Filtering (filter type 3 and 4) ........................................................................ 41
4.2.1.1 Example .................................................................................................................. 42
4.2.1.2 Example .................................................................................................................. 42
5 DATA CONVERSION IN CROSS DATABASE COMPARISON .................................................. 44
5.1 Conversion of comparison key fields .................................................................................... 44
5.1.1 Conversion of integer comparison key fields .................................................................... 44
5.1.1.1 Example ................................................................................................................ 44
5.1.1.2 Restriction .............................................................................................................. 45
5.1.2 Conversion of decimal floating point number comparison key fields ......................... 45
5.1.2.1 Example ................................................................................................................ 45
5.1.2.2 Restriction .............................................................................................................. 45
5.1.3 Conversion of UUID/GUID comparison key fields ......................................................... 46
5.2 Conversion of comparison fields (no keys) ........................................................................ 46
5.2.1.1 Example 1 .............................................................................................................. 46
5.2.1.2 Example 2 .............................................................................................................. 46
6 CROSS DATABASE COMPARISON FOR XML FILES ......................................................... 47
6.1.1.1 Example .................................................................................................................. 47
7 CONSIDERATIONS FOR CDC SYSTEM LANDSCAPE ....................................................... 49
7.1 Overview on System and Transport Landscape ................................................................... 49
7.2 System and Transport Landscape at CDC design-time (for source type ABAP „RFC to SAP
ABAP System“) .................................................................................................................. 49
7.3 System Landscape at CDC run-time for Data Extraction (for source type ABAP „RFC to SAP
ABAP System“) .................................................................................................................. 49
7.4 System Landscape at CDC design-time (for source type ADBC „Remote Database“) .... 51
7.5 System Landscape at CDC run-time for Data Extraction (for source type ADBC „Remote
Database“) ......................................................................................................................... 53
8 MASS PROCESSING/MASS MAINTENANCE ............................................................................. 54
8.1.1.1 Procedure ................................................................................................................ 54
8.1.1.2 Procedure ................................................................................................................ 54
8.1.1.3 Procedure ................................................................................................................ 54
8.1.1.4 Procedure .................................................................................................................. 55
8.1.1.5 Procedure .................................................................................................................. 55
8.1.1.6 Procedure .................................................................................................................. 55
8.1.1.7 Procedure .................................................................................................................. 56
9 BLOCK SIZE ..................................................................................................................... 57
INTRODUCTION TO CROSS DATABASE COMPARISON (CDC)

You use this application to compare data sources with a complex structure or hierarchy. You can also do this across different systems. By doing so, you check whether the data in the source and target systems is consistent, for example, whether updates in the source system have been correctly replicated to the target system. Examples of complex data sources are sales orders with several billing items or a customer master record containing multiple addresses, partner roles, and bank details.

Even though the data sources must contain the same data logically, they can have different values once they have been forwarded to a different system, for example, from an ERP system to a CRM system.

You can:

- Model the data you want to compare
- Compare data at any time or schedule regular comparisons
- Display the results in an overview containing all the relevant details

This is a generic application, which means that you can use it to check the data consistency in both ABAP and non-ABAP systems. You can use this application to compare the following data sources:

- Tables from the ABAP Dictionary
- Tables that are linked via ABAP Database Connectivity (ADBC)
- XML files on a Solution Manager application server
- XML files on your local PC (new in Solution Manager 7.1 SP05)

You use the application as follows:

- Once to check new links between data sources
- When necessary to check existing links
- Regularly to monitor important data
In short, during Cross Database Comparison

- You create comparison objects in the solution manager system. In a comparison object, you specify which data from which data sources (SAP/Non-SAP systems) you want to compare and create filters. The data extractors provided in the managed systems, extract the required data and save it either centrally on a file server and later used by comparison run or directly used for comparison run in the Solution Manager system.
- You create comparison instances. A comparison instance contains a comparison object. In a comparison instance, you specify which specific systems are to be compared using which filter settings.
- You create comparison groups. A comparison group contains several comparison instances which are grouped logically and started and stopped together during the comparison run.
- You schedule comparison runs. By doing so, you specify when and how often you compare the relevant data sources.
- You display the comparison results. By doing so, you can see where the inconsistencies are in the data sources. You can display the results of the comparison in business process monitoring and in the BPO dashboard as well. To do so, configure the relevant settings in Customizing for business process monitoring and for the BPO dashboard.

1.1 Implementation considerations

You use this application because you suspect that there are inconsistencies between systems and want to check the data consistency of your systems.

1.1.1 Prerequisite

You have the roles and authorizations required to use the application. This will be discussed more in depth in Prerequisites for Cross Database Comparison.
1.1.2 Integration

The Cross Database Comparison is a separate application and therefore independent of other applications. No further components are required to use this application.

1.1.3 Features

Cross Database Comparison:

- Is contained in SAP Solution Manager 7.0 SP28 or SAP Solution Manager 7.1, no add-on needed (except for using the generic ABAP extractor function Module or integration into Business Process Monitoring). We strongly recommend using Solution Manager 7.1 SP05 or higher. Additional features are available from Solution Manager 7.1 SP10.
- Offers comparison of multiple tables in two source systems
- Supports multiple source types including SAP ABAP system, remote database and XML files
- Can be enhanced by custom source types
- Offers easy to use graphical UI to enter data model
- Extracts data from sources and executes comparison in SAP Solution Manager
- Offers possibility to save and re-use comparison parameters and data model
- Offers functionality to save and display comparison results
- Allows splitting comparison data into multiple blocks of configurable size
- Supports integration into Business Process Monitoring

1.1.3.1 New features with Solution Manager 7.1 SP5

- Supports iteration
- Supports data conversion
- Upload and download of Comparison Objects and data models as XML
- Proposals for foreign key relations and mappings
- Key fields, filters and conversions are visible in graphical data model
- New Source Type “Local File”
- Pie-Chart for distribution of inconsistency types and affected fields
- Trend-Charts
- Confirmation of Comparison results

1.1.3.2 New features with Solution Manager 7.1 SP10

- Supports all kinds of filtering on Comparison Object and Comparison Instance level
- Integration into Incident Management
- Grouping of Comparison Instances and correlation of results
- Integration into BPO Dashboards
- Supports leading systems via different extraction strategies
- New Source Type to access ABAP-based systems dynamically
- Version Management and history of changes
- Extraction of context fields
- Improved authorization concept
- Mass generation of Comparison Objects and Comparison Instances

1.1.3.3 New features with Solution Manager 7.1 SP12

- New extraction strategy to execute comparison in one system
• New source type HANA for comparison in one HANA database to compare data between one HANA database and a second database
• Integration of Internal Database Comparison with new source type
• Improved SLT integration
• Improved source type XML file access on application server
• Offers new comparison scenarios and special options to execute a comparison run
• Allows to set block size for each source system in comparison instance
2 PREREQUISITES FOR CROSS DATABASE COMPARISON

Following are the technical prerequisites which have to be fulfilled before using the SAP Solution Manager for Cross Database Comparison:

- Correct versions of the software components
- Activation of needed services
- Authorization for users
- Connections to the data source system
- Whitelists

2.1 Correct versions of the software components

In order to set up CDC, you require having a Solution Manager system with software component ST release 400 with the minimum Support Package Stack SP28 (down port from 7.1 SP01). Solution Manager 7.1 SP05 or higher is strongly recommended. Additional features are available from Solution Manager 7.1 SP10.

The newest releases and latest versions are always available via the SAP Service Marketplace by using the URL http://service.sap.com/swdc.

For the latest information regarding software requirements (Support Packages and SAP Notes) please refer to SAP Note 521820.

ST/A-PI needs to be considered only in such cases, where in you would want to integrate CDC comparison results into BPMon. ST-PI needs to be considered in cases, where you would want to use the generic ABAP extractor function module instead of generating specific ABAP extractor function modules for each data model.

2.2 Activation of services

In order to run the CDC WebDynpro application inside your browser, you have to activate the corresponding services per transaction SICF (right-click on service name and choose Activate Service):

- WDA_DSWP_CDC_MAIN as of Solution Manager 7.1 SP10 or WD_DSWP_CDC_MAIN for earlier versions
- WD_DSWP_CDCRESULT until Solution Manager 7.1 SP11
- WD_DSWP_CDC_ACF if you want to compare XML files from your local PC (available as of Solution Manager 7.1 SP05)
- WDA_DSWP_CDC_GENERATION if you want to use mass generation of comparison objects (available as of Solution Manager 7.1 SP10)
- WDA_DSWP_CDC_GENER_INST if you want to use mass generation of comparison instances (available as of Solution Manager 7.1 SP12)
- WDA_DSWP_CDCGROUP_M if you want to use comparison groups (available as of Solution Manager 7.1 SP10)

2.3 Authorization for users

Following different types of authorization are required to be able to create and run CDC comparison in solution manager:

2.3.1 CDC authorization objects on Solution Manager

Several authorization objects control the access to the different features of the Cross Database Comparison.
To access and work with the Comparison Object the following authorizations are needed:

- Authorization object SM_CDC_OBJ, activities
  - 01 – Create
  - 02 – Change
  - 03 – Display
  - 06 – Delete
  - 07 – Activate/Generate (= generate extractor coding)
  - 36 – Extended Maintenance (= create comparison instances for a comparison object)

To access and work with the Comparison Instance the following authorizations are needed:

- Authorization object SM_CDC_INS, activities
  - 01 – Create
  - 02 – Change
  - 03 – Display
  - 06 – Delete
  - 16 – Execute (= run the data comparison)
  - 35 – Output (= display the results of a comparison run)
  - 65 – Re-Organize (= delete the results of a comparison run)

To trigger the Extractor Function Module Generation in the managed system

- Authorization object SM_CDC_OBJ, activity
  - 07 – Activate, Generate

Extraction from remote database (ADBC connection)

- Authorization Object S_DBCON, activity
  - 71 – Analyze

In addition to the above, the access to a particular type of source system (SAP, Non-SAP) could also be limited with the use of authorization fields CDC_S_TYPE & CDC_CONN (ADBC connection or RFC-connection).

We strongly recommend creating your own specific user roles based on template roles that are specifically available for CDC viz. SAP_CDC_ADMIN (all authorizations) & SAP_CDC_DISPLAY (display-only).

Other optional roles which could be required depending upon extent to which Solution Manager system is being used and the level of software components.

With Solution Manager 7.1 SP10 you can use preconfigured roles in order to establish a better segregation of duties. The typical CDC tasks are split up to different user roles:

- **Development**: The object modeler role SAP_CDC_OBJECT_MODELER can create/change/delete comparison objects to perform the modeling and generation, but is not allowed to create a comparison instance of it.
- **Administration**: The instance creator role SAP_CDC_INSTANCE_CREATOR can display and use comparison object in order to create/change/delete comparison instances, but cannot change the comparison object model.
- **Scheduling**: The instance executor role SAP_CDC_INSTANCE_EXECUTER can execute and reorganize the comparison run, but can neither output the result nor change the comparison objects and instances.
2.3.2 CDC authorization objects on Managed System

The user in Solution Manager system should be able to create a function group and RFC-enabled function modules in case of RFC connections.

To generate extractor function modules in the managed system, the following authorizations are needed:

- Authorization object S_DEVELOP, activities
  - 01 – Create
  - 02 – Change
  - 03 – Display
  - 06 – Delete
  - 07 – Activate/Generate
  - 16 – Execute

- Authorization object S_RFCACL, activities
  - 16 – Execute

- Authorization object S_RFC, activities
  - 16 – Execute

To execute Comparison runs the following general authorizations are needed to secure the access to the needed tables:

- Authorization object S_TABU_DIS
- Authorization object S_RFC
- Authorization object S_TCODE

2.4 Connections to the data source system

You can select whether you want to extract the data from the following source systems:

- SAP ABAP System (using RFC)
- Remote Database (using ADBC)
- XML file on Application Server
- XML file on local PC (new in Solution Manager 7.1 SP05)

For RFC connection to generated ABAP extractor function modules, the following two RFC connection types need to be created:

- RFC Destination (Read): Specify the connection that is used to read the data from the ABAP Dictionary of the source system.
- RFC Destination (Trusted): Specify the connection that is used to create ABAP extractor function modules in the source system. With a trusted connection you can execute RFCs in the target system without specifying a password explicitly.

For RFC connection to the generic ABAP extractor function module, the following RFC connection type needs to be created;
CROSS DATABASE COMPARISON – SETUP AND ENDDUSER GUIDE

- RFC Destination: Specify the connection that is used to read data from the ABAP Dictionary of the source system using the generic extractor function module.

For remotely accessing DB using ADBC connection, you need to specify the following:

- Database Connection: Specify the connection entry that is used to read the data for the comparison (as maintained per transaction DBACOCKPIT or DBCO)
- Database Schema: Specify the name of the database schema from which the data is read.

For accessing data that is stored in a XML file on an SAP Solution Manager application server, you need to provide with the following:

- File Path: Specify the path on the SAP Solution Manager application server under which the XML file containing the data to be compared is saved.
- File Name: Specify the name of the XML file

For accessing data that is stored in an XML file on the local PC, you need to specify the following:

- XML File Path: Specify the path on the local PC where the XML file containing the data to be compared is stored.

2.5 Whitelists

2.5.1 Whitelist to access the application server

If you want to compare data that is stored in an XML file on application server, for security reasons it is required that you maintain a physical path where the data is stored on the application server (new with Solution Manager 7.1 SP05).

2.5.1.1 Procedure
1. Start transaction FILE to maintain logical file path definitions.
2. Mark logical file path “CDC_FILE_PATH” and choose Assignment of Physical Paths to Logical Path in the dialog structure.
3. Enter one or multiple physical paths where the XML files are stored.

2.5.2 Whitelist to access the local PC

If you want to compare data that is stored in an XML file on the local PC (new with Solution Manager 7.1 SP05), for security reasons it is required that you maintain paths on the local PC from where data can be uploaded and servers to which data can be uploaded in a whitelist.

2.5.2.1 Procedure
1. In transaction SPRO, press SAP Reference IMG and choose SAP Solution Manager Implementation Guide → SAP Customizing Implementation Guide → Application Server → Web Dynpro ABAP → Set-Up Active Controls White List or directly call transaction WDR_ACF_WLIST.
2. Press Change Whitelist (F5).
3. If whitelist “WL_CDC” does not yet exist, add it (with description “CDC Whitelist”).
4. Mark whitelist “WL_CDC” and navigate to Upload in the dialog structure.
5. Check/maintain the following entries:
   - The name of the directory on the local PC where the XML file is located. File location type should be “Directory”. Example: “C:\Users\Username\Transfer”.
SAP Solution Manager host name (name + port) where the CDC application is running. If necessary, maintain all servers from SM51 transaction. Keep in mind that the system can substitute the host name via entries in the HTTPURLLOC database table. In the last case maintain the name from HTTPURLLOC DB table instead of SM51 transaction. File location type for these entries should be “Server”. Example: “abc.wdf.sap.corp:44390”.

6. Press *Install Certificate (Ctrl +F10)* to install the corresponding certificate on your local PC.
3 SETTING UP CROSS DATABASE COMPARISON

3.1 Starting Cross Database Comparison

You can start the Cross Database Comparison application from the Business Process Operations work center:

- In the navigation panel, choose Data Consistency Management
- In the Consistency Checks tray, press hyperlink Cross Database Comparison

Alternatively, you can add a direct link to the Cross Database Comparison application to the favorites in your SAP Easy Access menu:

- Choose Favorites → Add other objects
- Select Web Dynpro Application
- Enter Web Dynpro Applicat. ‘WD_DSWP_CDC_MAIN’ and a corresponding description

- From Solution Manager 7.1 SP10 enter Web Dynpro Applicat. ‘WDA_DSWP_CDC_MAIN’ and additionally Application Configuration ‘WDA_DSWP_CDC_MAIN_001’
Recommendation: With Solution Manager 7.1 SP09/10 there are also transaction codes for CDC, which hide the technical WebDynpro application name and are easier to use in role menus

- Transaction DSWP_CDC starts CDC embedded in SAP GUI window
- Transaction DSWP_CDC_UI starts CDC in an own Browser window

3.2 Defining Comparison Objects

Comparison Objects describe how data is compared between data sources. A Comparison Object contains the data fields in the data sources that you compare as well as the connection information during the design-time, for example to access Data Dictionaries. In the case of ABAP systems, you then generate the function modules that read data from the ABAP Dictionary tables. As of Solution Manager 7.1 SP10, you can alternatively generate SQL statements that are sent to a generic extractor function module to read the data from ABAP systems. In the case of non-ABAP systems, you generate the native SQL statements that the application uses to read the data.

You create a comparison object in a series of steps. Apply the sequence specified here. You can change settings from previous steps at any time.

You define a comparison object by:

- Creating a Comparison Object
- Maintain comparison tables and fields
- Maintain foreign keys
- Assign comparison table fields ("mapping")
- Generating extractors for a comparison object

3.2.1 Creating Comparison Objects

A comparison object contains information about the systems that contain the data sources that you are comparing. You specify the type of data source that contains the data that you want to compare. This means the data can be in different types of data sources – such as ABAP or non-ABAP systems.

3.2.1.1 Procedure

1. Choose 

2. Enter the name of the comparison object
3. Press Enter.

4. Enter a description for the comparison object.

5. The following settings are relevant for the comparison. Therefore, specify the following for all data sources:
   - **Extraction Strategy**: Define the extraction strategy for data selection
     - **Default**: Individual data selection in each source system: As default the extraction happens on both source systems individually.
     - **Use keys of source system 1 to select in source system 2 / Use keys of source system 2 to select in source system 1**: You need to filter on objects, that can only be identified in one source system based on filters and you want to compare exactly these objects with the other source system
     - **Execute comparison in one system** (new in Solution Manager 7.1 SP12): Execute the comparison in one source system and only pass the result to Solution Manager instead of executing the comparison in Solution Manager. There are two source types for this extraction strategy
       - Determine inconsistent entries in one system (IDC)
       - Comparison in one HANA data base

6. Specify the type of data source in which the data is stored. The remaining details depend on the type of data source.

   Depending on **Extraction Strategy** selected you can use the following data sources:

<table>
<thead>
<tr>
<th>Extraction Strategy</th>
<th>Default: Individual data selection in each source system</th>
<th>Use keys of source system 1 to select in source system 2 / Use keys of source system 2 to select in source system 1:</th>
<th>Execute comparison in one system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Type:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAP ABAP System (using RFC to generated extractor)</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>SAP ABAP System (using RFC to generic extractor)</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Remote Database (using ADBC):</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>XML File on Application Server</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>XML File on local PC</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
</tbody>
</table>
• **SAP ABAP System (using RFC to generated extractor):** You choose this for an ABAP-based system. The data is read using a function module, which needs to be individually generated for each comparison object.

Enter the following data:

- **RFC Destination (Read):** Specify the connection that is used to read the data from the ABAP Dictionary of the source system.
- **RFC Destination (Trusted):** Specify the connection that is used to create the function module in the source system.

Choose ![Test Connection](image) to test the connection.

- **Function Group:** Specify the name of the function group (new or existing function group).
- **Function Name:** Specify the name of the function module that is generated by the application.
- **Package (new in Solution Manager 7.1 SP05):** Specify the name of the package in which a new function group should be created. Not required for already existing function groups.
- **Transport Request (new in Solution Manager 7.1 SP05):** Specify the transport request that should be used for the new function group and/or new function module. Not required for new function groups and function modules stored as local object (package $TMP).

• **SAP ABAP System (using RFC to generic extractor (new in Solution Manager 7.1 SP10)):** You choose this for an ABAP system. The data is read using a generic extractor function module. It is not necessary to generate individual extractor function modules but you generate SQL statements that are used by a generic extractor function module to extract the data. You can find more detail about generic extractor function module in SAP note 1819794 - CDC: Generic extractor function module.

Enter the following data:

- **RFC Connection:** Specify the connection that is used to read the data from the ABAP Dictionary of the source system.

Choose ![Test Connection](image) to test the connection.

• **Remote Database (using ADBC):** You choose this for a non-ABAP system. The data is read via a secondary database connection using a native SQL statement.

- **Database Connection Name:** Specify the connection that is used to read the data for the comparison.

Choose ![Test Connection](image) to test the connection.

- **Database Schema Name:** Specify the name of the database schema from which the data is read.
- **SQL Statement (Count):** The SQL statement used to count the number of expected objects is displayed after generation.
- **SQL Statement (Extract):** The SQL statement used to extract the source data is displayed after generation.
• **XML File on Application Server**: You choose this for data that is stored in a XML file on an SAP Solution Manager application server. The source type is able to process single and multiple XML files (new in Solution Manager 7.1 SP12).
  
  - **Host name**: Specify the name of the application server
  - **File Path**: Specify the path on the SAP Solution Manager application server under which the XML file containing the data to be compared is saved.
  - **File Name**: Specify the name of the XML file. In no file name but only a path name is supplied, all files in the given path are processed. Moreover, the file name allows wild-cards *, +, $TODAY and $YESTERDAY. In case multiple files are selected, data from all these files is processed together.
  - **Created from, Created to** (new in Solution Manager 7.1 SP12): Enter a time from when and a time until when files are considered. The following key words are possible for entering the times:
    - $TIMES – timestamp now with offset in seconds (example: $TIMES-10)
    - $TIMEM – timestamp now with offset in minutes (example: $TIMEM-30)
    - $TIMEH – timestamp now with offset in hours (example: $TIMEH-12)
    - $TIMED – timestamp now with offset in days (example: $TIMED-1)
  - **XML transformation** (new in Solution Manager 7.1. SP12): Enter a parameter in order to execute an XLS transformation which changes data from a different format to the required asXML format. In this way it is now possible to process XML files which are originally not in the required asXML format.

• **XML File on local PC** (new in Solution Manager 7.1 SP05): You choose this for data that is stored in an XML file on the local PC. For this source type, you do not have to enter any further data during Comparison Object creation. The path and name of the XML file is specified later on during Comparison Instance creation.

• **Comparison in one HANA database** (new in Solution Manager 7.1 SP12): you choose this to compare data between one HANA database and a second database (HANA or a different database) directly in the HANA database.
  
  - **Database Connection Name**: Specify the connection that is used to read the data for the comparison
  - **Database Schema Name**: Specify the name of the database schema from which the data is read
  - **SQL Statement (Count System 1, Count System 2)**: The SQL statement used to count the number of expected objects is displayed after generation.
  - **SQL Statement (Comparison)**: the SQL statement used for comparison between the data in the two source systems.
  - **SQL Statement Extract**: The SQL statement used to extract the source data is displayed after generation.

• **Determine inconsistent entries in one system (IDC)** (new in Solution Manager 7.1 SP12): you choose this for comparison in a single database to compare data in one database (usually missing header entries for dependent tables). This includes the existing use case for Internal Database Comparison (missing entries in one database) and improves it with further features like more complex data models as now joins between multiple tables are allowed.

  The generic IDC extractor function module /SDF/DCC_CDC_IDC_GEN_EXTRACTOR must exist in the managed system - SAP note 1984964.

  Procedure:
Choose extraction strategy *Execute comparison in one system and source type* *Determine inconsistent entries in one system (IDC).*

- **RFC Destination:** Specify the connection that is used to read the data from the ABAP Dictionary of the source system.

7. The following settings are relevant for the comparison. Therefore, specify the following for all data sources:

- **Comparison Block Size:** Specify the block size used to read the data for the comparison. You can use this value for the comparison according to your system performance.
  - Typically a higher block size is more effective, as it reduces the technical overhead for repetitive block extractions. However, choosing a too high block size may lead to resource issues such as memory overflow.
  In comparison instance you can give an alternative block size per source system.
- **Max No. Errors:** Specify the number of error messages that are permitted until the system terminates the comparison.
  - Enter 0 if you want the comparison job to be executed fully. Use this setting to prematurely terminate an incorrectly configured comparison that returns an error for every data record.
  In Comparison Run in Schedule Option is possible to choose the option “Continue comparison after “Maximum Errors” reached, but without details” which means that the comparison will continue after the number of identified inconsistencies exceeds the maximum error, but no further details will be written. (See chapter 3.5.1 Scheduling Comparison Run)

8. Save your entries.

You have created a comparison object.

In this comparison object, now create the data fields that you want to compare with one another.

### 3.2.2 Maintain comparison tables and fields

For the comparison, you select data fields from one table or multiple tables arranged hierarchically after one another. To do this, create comparison tables in which you specify the data sources and data fields that you want to compare.

Regardless of the type of data source, the objects that you compare are represented as tables in this application.
You can select as many tables as you want for each system. You join the tables to one another using foreign keys (see Creating Foreign Keys). You assign foreign keys in the root table or the higher-level child table to the primary keys in the child tables. First create the root table: The first table that you create in a system automatically becomes the root table. Subsequent tables that you create become child tables. When the systems are compared, the information from the child tables is read using the foreign keys for the root table.

3.2.2.1 Procedure

1. Choose Comparison Object → Edit

2. Inside the box Data Model for Comparison, on the editor, make a right click, choose Node → Add Node → TABLE_SOURCE_1.

A dialog box appears.

3. Specify the data source that you want to compare and select the data fields that you want in include in the comparison.
4. Press pushbutton OK.

The data sources and data fields that you have selected are displayed graphically.

5. Save your entries.

6. To create additional tables in the first system, in the context menu for the comparison table, choose Node → Add Node → TABLE_SOURCE_1.

7. To create the data fields for additional tables, repeat steps 2 to 3.

8. To create tables for the second system, in the context menu for the comparison table, choose Node → Add Node → TABLE_SOURCE_2 and repeat steps 2 to 7.

9. Save your entries.

You have selected the tables and data fields that you want to compare. Now create the foreign keys.

Hint: For ABAP-based systems, do not include client fields (CLIENT or MANDT) in your data model as in this case the selection will not work as expected. The generated extractors will contain OpenSQL coding that handles the client automatically (based on the logon client of the RFC destination).

3.2.3 Maintain Foreign Key

Foreign keys specify the relationship between root and child tables within a system. Assign data fields in the root table or higher-level child table, as foreign keys, to the primary keys in the child tables.

When you create multiple tables in a system, you assign foreign keys. This specifies relationships between the tables in a system. You must have defined the relationships between the various tables in a system to be able to validate the comparison object (Validate Comparison Object) and generate the function modules and SQL statements.
3.2.3.1 Procedure
1. To create foreign keys for the first system, choose *Comparison Object* in Edit mode.

2. Create the foreign keys by dragging the arrows on the outer side of the child tables (left for Source 1 and right for Source 2), to its higher-level tables. Join the tables so that the arrow goes from the child table to the root table (higher-level).

3. Save your entries.

You have created the foreign keys.

To delete foreign key relations, drag the arrow head onto some empty space in the data model editor screen.

3.2.4 Maintain Filters

In this step you can choose the filter type and enter the filter criteria. You can use various types of filters in Cross Database Comparison to restrict the data being compared for the sources. You can create filters for both source systems or for one source only. (Further details about filtering see [Filtering in Cross Database Comparison](#)).

3.2.4.1 Procedure
1. To maintain filter for the first or the second source system, choose *Comparison Object* in Edit mode.

2. Choose the field in the data model for that you want to set filter. If you want to use an instance filter, set the check box. If you want to use object filter, enter the filter value.
You can enter a filter type for an instance filter which allows that the filter values in the comparison instance can be entered in a field with the correct data type. Use the dropdown list box to select the correct filter type. Use the special filter type "relative timestamp" to enter relative timestamps in the comparison instance. (Further details about filtering see Filtering in Cross Database Comparison.)

3.2.5 Maintain Context Fields (Display fields)

In the Cross Database Comparison (CDC) application, you usually extract only data, which is needed for the comparison. In addition (with Solution Manager 7.1 SP10) you can now extract further context fields just for information purpose on the comparison result UI (see Display Comparison Results -> Result Details).

There are two types of the context fields:

- The field is only available in one of the source systems (e.g. some native key field or additional attribute), and therefore cannot be included as a common mapping field
- The field is available in both source systems but always has different content (e.g. created-by fields) by purpose, and therefore is not suitable as a mapping/comparison field
Now specify the assignment between the comparison tables.

### 3.2.6 Assign comparison table fields

In this step, you specify which data fields in the tables in the first system you want to compare with which data fields in the tables in the second system and which data fields are used as comparison keys. You assign this mapping a name with which you can later identify it.

You also create fixed and variables filters that you use to restrict the comparison. For fixed filters, you specify the data field and the exact value of the filter as soon as you create the comparison object. For example, you specify the restriction Country="DE" when you create the comparison object.

For variable filters, you specify when you create the comparison object that a certain data field is to be used as a variable filter, for example the Country data field. The variable filters are displayed in the comparison instance, where you can specify values for them, such as Country="DE". This allows you to create multiple comparison instances for the same comparison object and specify different filter values for each – for example, Country="DE" for one comparison instance and Country="IN" for another comparison instance.

#### 3.2.6.1 Procedure

1. Choose Comparison Object → Edit.

2. Define the assignment by dragging the arrows provided between the tables from left to right.

A dialog box is displayed for each mapping assignment.
Enter the following data:

- **Tag Name**: Enter a name for the assignment.
- **Comparison Key No.**: You create at least one comparison key for the comparison and specify the sequence of the comparison keys.

Enter 1, 2, 3, and so on, to define the sequence of the comparison keys. Leave the field empty (0) for the data fields for the actual comparison. Ideally you can use the primary keys of the source tables as a comparison key. Alternatively make sure you chose another unique secondary key. It is mandatory to extract a unique and sorted comparison key. Extracting non-unique (=duplicated) comparison keys from a source system will later on result in a termination of the comparison run.

3. **Variable Filter**: Enter X to indicate that the assignment is a variable filter.

If you do not want to use the assignment as a filter, leave the field empty.

With Solution Manager 7.1 SP05, you have the possibility to enter a filter type for the variable filter which allows that the filter values in the comparison instance can be entered in a field with the correct data type. Use the dropdown list box to select the correct filter type. Use the special filter type “relative timestamp” to enter relative timestamps in the comparison instance. For further details about filtering see Filtering in Cross Database Comparison.

Moreover, with Solution Manager 7.1 SP05, you have the possibility to choose pre-defined conversion IDs (such as delete leading zeroes, convert to upper case) for both source systems that allow you to convert the data before it is compared. You can also define and use your own conversion IDs by creating an implementation for enhancement spot DSWP_CDC_CONVERSION and entering the conversion ID in maintenance view CDC_V_CONV_TYPE. For further details about data conversion see Data Conversion in Cross Database Comparison.
4. Save your entries.

To undo the mapping, drag the arrow from the mapping onto the empty space in data model editor screen.

You have defined the assignment between the data fields of the comparison tables. Now create the function module for the comparison.

### 3.2.7 Generating Extractors for a Comparison Object

When you have created the comparison object with the comparison tables, and have assigned the table relationships in and between the systems, you generate the extractors for the comparison. For ABAP systems, the system creates the function modules executed during a comparison when you generate these extractors or generates the SQL statements that are used by the generic extractor function module. For non-ABAP systems, the system creates SQL statements to read the data during a comparison when you generate these extractors. You can test the SQL statements in the application before you generate them.

When you have changed settings in the comparison object model, you must generate the comparison object again so that the extractors take into account the changes in the model.

Before the application generates the extractors, it tests the connection and validates the comparison object. If errors occur, the application terminates generation. Before generation starts, you should therefore check the connection and the comparison object.

If you have saved the comparison data in an XML file in SAP Solution Manager or on a local PC, do not generate anything because the system reads the data directly from the XML file. However, you can test the specified path.

#### 3.2.7.1 Procedure

1. To test the connection, choose (Test Connection).

2. To test the comparison object, choose (Validate Comparison Object).

3. The remaining details depend on the type of data source:

- **SAP ABAP System (RFC to generated extractor)**
  - Choose Comparison Object → Display and then (Generate).

  Here, you specify the connection for initial function module creation. Create the function module in a development system. Test the function module and the function group in the specified development system. Then transport the function module to a test system.

  In comparison instances, you can connect systems different to those in the comparison objects. To execute the comparison object for multiple comparison instances, transport the function module to the systems that contains the tables that you want to compare. In other words, if you have created a function module in multiple systems, create multiple comparison instances to compare the tables in the various systems.

- **SAP ABAP System (RFC to generic extractor)**
  - Choose Comparison Object → Display and then (Generate)
  - Check the SQL statement for the generic extractor.

- **Remote DB (Generic ADBC)**
  - Choose Comparison Object → Display and then (Generate).
  - Check the SQL statement for the extractor and the count statement.
You have now fully created the comparison object.

### 3.2.8 Generation of Comparison Object from SAP Landscape Replication Server (SLT) repository

With Solution Manager 7.1 SP10 you can use a guided self-service to generate comparison objects from the LT-Replication server’s repository for comparison of data in an ABAP source system to the data replicated to HANA.

#### 3.2.8.1 Procedure

1. **Choose** `Comparison Object → Mass Maintenance → Start Mass Creation from SLT Replication Model for HANA`

   Enter the following data:

   - **RFC destination SLT system**: Specify the connection that is used to read the data from the SLT system.
   - **Mass Transfer ID** (new in Solution Manager 7.1 SP12)
   - **RFC destination source system**: Specify the connection that is used to read the data from the ABAP Dictionary of the source system.
   - **DB connection for target system**: Specify the database connection that is used to read the data from HANA.
   - **DB schema for target system**: Specify the name of the database that is used to read the data from HANA.
   - **Prefix for Comparison Object name**: Specify a prefix that – together with the table name – is used as comparison object name.
   - **Extractors to be generated for Source** (new in Solution Manager 7.1 SP12): Both Source Systems (“both side” of comparison object), Source System 1 (“left side” of comparison object), Source System 2 (“right side” of comparison object), No Source System
   - **Include client field**: Specify if the client field should be included in the comparison.
   - **Test**: If this checkbox is marked, the generation of comparison objects will only be tested. You will see the tables available in the SLT repository and the names of the comparison objects that would be generated, but the comparison objects will not be created.

2. **Choose** `Next`

   - Select the tables from the LT-Replication server’s repository for which comparison object should be created.

3. **Choose** `Next`

   - You will have an overview about the selected tables and corresponding comparison objects which you have to submit and to create the comparison objects.

You generated the Comparison Object from the SLT repository.

### 3.2.9 Change Documents for Comparison Objects

Change Documents allow to track by whom and when a field value was inserted/changed/deleted. This is supported for header data attributes, multi-lingual descriptions, source-specific parameters and variable filter criteria.

#### 3.2.9.1 Procedure

1. To display changes, choose `Comparison Object → Display`

2. Choose `Display Change Log`
3.2.10 Versioning of Comparison Object Data Models

With Solution Manager 7.1 SP10, each change of the Comparison Object Data Model automatically creates a new version. You can review older version of the Data Model and you can reactivate an older version as current state.

### 3.2.10.1 Procedure

1. To load older version, choose Comparison Object → Display

2. Choose → Load old version

3.3 Defining Comparison Instances

In this view, you create comparison instances. You run a comparison by scheduling and executing comparison instances in a subsequent step.
A comparison instance contains one comparison object. You can assign a comparison object to multiple comparison instances. For example, you create multiple comparison instances for a comparison object to access function modules transported to various systems (test and production systems). To be able to do this, the names of the tables and data fields that you specified in the comparison object must remain unchanged.

The Instance Specific Filters area lists the filters. You can specify values for the variable filters that you created in the comparison object. This allows you to create multiple comparison instances for the same comparison object and specify different filter values for each – for example, Country="DE" for one comparison instance and Country="IN" for another comparison instance.

A comparison instance can be changed later at any time.

### 3.3.1.1 Procedure to Create Comparison Instances

1. Choose **Comparison Instance → Create**.
2. Select the comparison object for which you want to create the comparison instance.
3. Enter the name of the comparison instance and choose **Next**. The global details for the comparison object are displayed in the comparison instance.
4. Enter the description of the comparison instance.
5. You can enter an alternative block sizes per source system. If you do not change values, the default block size specified in the Comparison Object will be used.
6. Enter the connection information that applies to this comparison instance:
   - For source type SAP ABAP System (RFC to generated extractor)
     - RFC Destination (Read): Specify the connection that the system uses to read the data from the ABAP Dictionary of the source system. Choose (Test Connection) to test the connection.
     - Function Module: Specify the name of the generated extractor function module that the application uses to read the data.
   - For source type SAP ABAP System (RFC to generic extractor) (new in Solution Manager 7.1 SP10):
     - RFC Destination: Specify the connection that the system uses to read the data from the ABAP Dictionary of the source system using the generic extractor function module.
   - For source type Remote DB (Generic ADBC):
     - Database Connection: Specify the connection that the system uses to read the data for the comparison. Choose (Test Connection) to test the connection.
     - Database Schema: Specify the name of the database schema from which the system reads the data.
     - SQL String (Extractor): The SQL statement used to run the comparison is displayed after generation.
   - For source type XML File on Application Server:
     - File Path: Specify the path on the SAP Solution Manager application server under which the XML file containing the data to be compared is saved.
     - File Name: Specify the name of the XML file.
   - For source type XML on local PC (new in Solution Manager 7.1 SP05):
     - XML file path: Specify the path and file name on your local PC where the XML file with the data to be compared is stored. To be able to use the input help for the XML file path, there must be corresponding entries in the CDC whitelist, see Whitelists.
7. With Solution Manager 7.1 SP05, the CDC application supports iteration, which means that you have the possibility to re-compare inconsistent objects. Iteration is useful if data changes frequently. It helps you to exclude temporary differences from the comparison result. Iteration can be started automatically after the first comparison is finished or manually. To start iteration automatically, enter at least one of the following parameters:

- **Maximum Number of Iterations**: The maximum number of iterations that will be started automatically. If one of the exit conditions is met before the maximum number of iterations is reached, iteration will stop anyhow.
- **Delay**: Time delay between two iterations (in seconds)
- **Exit Condition (%)**: Iteration stops if this percentage of inconsistent objects or a value below is reached.
- **Exit Condition (total)**: Iteration stops if this number of inconsistent objects or a value below is reached.

8. If the traffic light icon is yellow, the source type specific parameters in the comparison instance are not the same as the function modules in the comparison object. This can be useful if you have changed settings in the comparison object and regenerated the function modules, but are not yet finished making your changes. The comparison instance can then continue to apply the existing settings until the changes are finalized.

9. To copy the current source type specific parameters from the comparison object to the comparison instance, choose the traffic light icon when you have finished making your changes to the comparison object, for example.

10. In the **Instance Specific Filters** screen area, specify values for the instance specific filters. You can create multiple comparison instances with different instance specific filters for the same comparison object. As of Solution Manager 7.1 SP05, the data type of the fields depends on the filter type you entered during comparison object creation. If you chose filter type “relative timestamp”, you can use
11. Save your entries.

The comparison instance is now created.

### 3.3.2 Change Documents for Comparison Instances

Change Documents in Comparison Instance allow to track by whom and when a field value was inserted/changed/deleted.

Supported for header data attributes, multi-lingual descriptions, source-specific parameters, variable filter criteria.

#### 3.3.2.1 Procedure

1. To display changes, choose Comparison Object → Display

2. Choose → Display Change Log

### 3.4 Defining Comparison Groups

In this view you can group comparison instances logically (e.g. by systems to be checked, data to be checked). The grouped instances will be started and stopped together in the comparison run.
There are three use cases:

- **Display** is used to group comparison instances according to logical criteria without additional functionality.
- **Distribution** is used to model a distribution from one system into many dependent systems (1:n relationship).
- **Sequence** is used to model transfer of data across a chain of systems.

3.4.1.1 **Procedure to maintain Comparison Group**

1. Choose Comparison Group

2. Position the cursor on Comparison Group type in that you want to create the new group and click the right mouse button.

3. Choose Create Group
4. Choose Edit/Display

5. In this view you give name and description to the newly created group.

6. After assigning the comparison instances to the group save your entries

7. With Start Group you schedule the comparison run for your group.

3.5 Executing and Evaluating Comparison Runs

A Comparison Run is the actual execution of a Comparison Instance. You plan when comparison instances are to be executed and display the actual results of a Comparison Run in the Comparison Run Overview. If you execute the comparison instances a number of times, you can switch between the results for each execution in a detail screen and display them individually. It is also possible to define whether the comparison is to be executed immediately or scheduled for another time.
3.5.1 Scheduling Comparison Runs

To run a comparison, start the comparison instance. The comparison job is scheduled. To do this, the system creates a job in transaction SM37, which you can either execute immediately or schedule manually in transaction SM37.

You can also schedule a comparison job to execute regularly and start the job immediately. The run ID is incremented chronologically.

3.5.1.1 Procedure to schedule a job

1. In the Comparison Run Overview view, select a comparison instance and choose Start. The Scheduling of a Comparison Job dialog box appears.

2. You have the following options:
   - To create a job that can be scheduled afterwards, choose Create job only. Call transaction SM37 and schedule the comparison job. The name of the job is CDC_<instance name>. The job is now scheduled. To display the scheduling, choose Update.
   - To create a job and execute it immediately, choose Create job and start immediately.
   - To start the comparison run immediately in dialogue, choose Start Comparison Run Immediately in dialogue (new in Solution Manager 7.1 SP05).

If certain source types are used, not all options might be available. For source type “XML file on local PC” which is available with Solution Manager 7.1 SP05, only option Start Comparison Run Immediately in dialogue is available.

There are further options available in WebDynpro UI on the Scheduling Popup (you may need to press the “Expand tray” icon to see the Special Option tray area):

3. Furthermore, you can check Rescan identified inconsistencies to manually trigger iteration. During iteration, only the inconsistencies identified during the last iteration will be re-compared.

4. With setting of checkbox Try to restart the cancelled/aborted previous run you can rerun a previous unsuccessful comparison run.
5. Before the actual comparison, a “count” is requested from both source systems, to know the number of expected objects and to be able to calculate the progress during the comparison run. The way how the counts are handled and to what extent the comparison is being performed, can be controlled by different Comparison Scenarios (new in Solution Manager 7.1 SP12):

- **Count expected records only (no comparison)**: This will only trigger the pure counting of the number of expected records in both source systems. This does not include a check for identical keys. So an identical count does not necessarily mean that these are the same objects.

- **Only do detailed comparison if counts are not identical**: This will first trigger the counting of the number of expected records in both source systems. If it is identical, no comparison will be done. If the counts are different, a detailed extraction and comparison will be done. A check for identical keys is not performed. So an identical count does not necessarily mean that these are the same objects.

- **Standard comparison with detailed results up to Maximum Errors**: This is the recommended default option for the standard comparison runtime behavior. It will first trigger the counting of the number of expected records in both source systems. Afterwards a detailed extraction and comparison will be done. If the number of identified inconsistencies exceeds the maximum error count (as defined in the comparison object), the comparison run will be stopped prematurely.

- **Continue comparison after “Maximum Errors” reached, but without details**: This is the same as the standard comparison. The only difference is that the comparison will still continue, even after the number of identified inconsistencies exceeds the maximum error count. However, no further result details will be written. It just increases the counts of the inconsistency key figures. Caution: This option can dramatically increase the total runtime, especially if there is some systematic error, like an unsuitable definition of the comparison object or at comparing completely non-matching source systems.

### 3.5.2 Stopping Comparison Runs

You can stop a comparison instance job that it is currently being executed. For example, you might want to stop the job if the comparison is taking too long and you therefore want to adjust the settings, or if the comparison has already found numerous inconsistencies and you want to first analyze these errors.

If you want to reschedule the job for a comparison instance, call transaction SM37 and reschedule the comparison there. The name of the job is CDC_<instance name>.

#### 3.5.2.1 Procedure to stop a Comparison Run

1. In the Comparison Run Overview view, select a comparison instance and choose Stop.

2. The comparison instance job stops as soon as possible.

You can now make changes to the comparison object, the comparison instance, or the execution. Then execute the comparison again.

### 3.5.3 Display Comparison Results

Once you have executed a comparison instance, you can display the results. You can also display results from previous comparisons. The Result Details screen lists all objects for which differences between the compared data sources are identified.
3.5.3.1 Procedure

1. In the Comparison Run Overview, select a comparison instance. The Comparison Instance Overview provides an overview of the status of the evaluations, for example:

   - *Run Status*: A grey LED indicates that the comparison instance has not yet been executed. A green traffic light means that the comparison instance run finished successfully, a yellow traffic light that the comparison instance is running and a red traffic light that the comparison instance run was canceled or aborted because of an error. A green checkmark means that the comparison instance run was confirmed.
   - *Incon. Status*: A grey LED indicates that the comparison instance has not yet been executed or the comparison instance run has not finished successfully. A green traffic light means that no inconsistencies were detected, a yellow traffic light that the comparison instance is running and a red traffic light that inconsistencies were detected.
   - *Run ID*: Indicates how often the comparison has been executed
   - *Status Text*: Indicates how the last comparison was executed. The status text can be one of the following:
     - *Initial*: You have created but not yet scheduled the comparison instance.
     - *Running*: The comparison job is currently being executed.
     - *Finished*: The comparison job was executed successfully.
     - *Terminated*: The comparison job terminated with an error.
     - *Paused*: The comparison job was stopped.
   - *Progress*: Indicates to what extent the comparison job was last executed.
   - *Timestamp*: Indicates when the comparison job was last started.
   - *User Name*: Indicates the name of the user who last started the comparison job.

   When the comparison has run successfully, you can display the comparison results.

2. Once the correct comparison instance is selected choose Show Last Result.
   The results of the last comparison are displayed.

   - In the Run ID field, enter another value to the result of another run of the same comparison instance. In the Iteration ID field, enter another value to see the result of another iteration of the same comparison instance run.
   - The results are broken down as follows:
     - *Global Result Parameters*: Provides an overview for the comparison instance and comparison object used in the comparison.
     - To display the comparison instance details, choose Show Comp. Instance Definition.
     - To display the comparison object details, choose Show Comp. Object Definition.
     - *Result Overview*: Provides a summary of the comparison results.

3. In the Result Overview, select Objects Existing in System 1 Only, Objects Existing in System 2 Only, or Objects with Differences. The details are then displayed in the Result Details. The distribution of context fields can be displayed in Result Details by choosing Show context field distribution:
4. Expand Comparison Runtime Statistics. Information about the duration and speed of the job are displayed.

5. Press Guided Procedures. This is a new feature in Solution Manager 7.1 SP05. The Guided Procedure Browser will be displayed which allows you to maintain and display guided procedures describing how to react on inconsistent data for a certain comparison object and comparison instance.

6. Press Result History Graphics. This is a new feature in Solution Manager 7.1 SP05. You can choose among History of Inconsistencies, Convergence of Inconsistencies, Runtime Statistics for Comparison Runs and Runtime Statistics for Iteration. The desired graphic will be displayed.

7. Press Create Incident. With Solution Manager 7.1 SP10 cross data base comparison is integrated with the incident management tool. By choosing Create Incident you can create message for detected inconsistencies.
3.5.4 Deleting Comparison Results

The volume of data recorded in the comparison results grows over time. You can therefore delete data that you no longer require. In doing so, you can specify the following:

- How old the data for deletion is
- Exactly what information you want to delete – just the details for a specific result or the entire result including the summary and details

3.5.4.1 Procedure

1. Call transaction DSWP_CDC_REORG.

2. Enter data as required.
   - Specify the comparison instance from which you want to delete data. If you do not specify a comparison instance, the system deletes the data from all comparison instances.
   - Specify which results you want to delete. For example, all results older than 30 days, but not the last three results.
   - Specify which information you want to delete from the results.

3. Choose Execute (F8).

To ensure that the comparison results are deleted regularly, schedule DSWP_CDC_COMPARE_REORG as a background job in transaction SM37.

The comparison results are deleted based on your selection.
4 FILTERING IN CROSS DATABASE COMPARISON

You can use various types of filters in cross-data comparison to restrict the data being compared for the sources. The following filter types are available in Cross Database Comparison.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Comparison Object</th>
<th>Comparison Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object filters with fixed values (type 1)</td>
<td>You can define a fixed filter value at source level. You can define them for both sources or for one only. The filter value is used during the generation to create a hard-coded where clause.</td>
<td>Not visible in Comparison Instance. Valid for all Comparison Instances of this Comparison Object.</td>
</tr>
<tr>
<td>Instance filters with variable values (type 2)</td>
<td>Instance Filter flag is selected in comparison object, either at mapping level (relevant for both sources) or at source level (relevant for one source only).</td>
<td>In Comparison Instance, the actual filter values can be maintained. These are used for all Comparison Runs of this Comparison Instance.</td>
</tr>
<tr>
<td>Object filters with relative dates (type 3, variant of type 1)</td>
<td>If a filter for a Comparison Object contains the syntax for relative dates (starts with '$'), it will not be generated as hard-coded filter in the extractor.</td>
<td>Not visible in Comparison Instance. Valid for all Comparison Instances of this Comparison Object.</td>
</tr>
<tr>
<td>Instance filters with relative dates (type 4, variant of type 2)</td>
<td>Instance Filter flag is selected in comparison object, either at mapping level (relevant for both sources) or at source level (relevant for one source only).</td>
<td>In Comparison Instance, relative date syntax can be maintained (starts with '$'). These are used for all Comparison Runs of this Comparison Instance.</td>
</tr>
</tbody>
</table>

4.1.1 Example for filter type 1

Enter a fixed value on object level for one source system:
4.1.2 Example for filter type 2

On object level, set flag and date type for “Instance Filter” on mapping details popup

Result: Variable filter on Comparison Instance level. Now you can maintain different values per Comparison Instance.

4.1.3 Example for filter type 3

On object level, enter relative dates syntax for one source system
Result: Filter just prepared in extractor, populated during runtime with translated absolute date

4.1.4 Example for filter type 4

On object level, define “Instance Filter” for “Relative timestamp” on mapping details popup

Result: Now you can maintain different relative date syntax values per Comparison Instance. Filter just prepared in extractor, but populated during with translated absolute date.

4.2 Relative Date Filtering (filter type 3 and 4)

You can enter specific values to define relative filters for dates. These can be used for time dependent comparisons. For example, you would like to compare new documents created on the current day. To perform relative date filtering, that is, calculate relative dates instead of absolute dates, Cross Database Comparison provides a syntax for defining fixed points and offsets for relative dates. Instead of a fixed (absolute) date, you can enter a keyword for the start date (using $ as a prefix) and optionally an additional offset for the difference in days:

Syntax = \langle StartDate\rangle[\langle Difference\rangle]

You can replace \langle StartDate\rangle with the following keywords:
<table>
<thead>
<tr>
<th>Keyword for &lt;StartDate&gt;</th>
<th>Description</th>
<th>Example using 2013/04/25 as Reference Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TODAY</td>
<td>Current date (today)</td>
<td>2013/04/25</td>
</tr>
<tr>
<td>$FDOCW</td>
<td>First day of current week</td>
<td>2013/04/22</td>
</tr>
<tr>
<td>$LDOCW</td>
<td>Last day of current week</td>
<td>2013/04/28</td>
</tr>
<tr>
<td>$FDOCM</td>
<td>First day of current month</td>
<td>2013/04/01</td>
</tr>
<tr>
<td>$LDCM</td>
<td>Last day of current month</td>
<td>2013/04/30</td>
</tr>
<tr>
<td>$FDOCY</td>
<td>First day of current year</td>
<td>2013/01/01</td>
</tr>
<tr>
<td>$LDCY</td>
<td>Last day of current year</td>
<td>2013/12/31</td>
</tr>
<tr>
<td>$FDOPW</td>
<td>First day of previous week</td>
<td>2013/04/15</td>
</tr>
<tr>
<td>$LDPW</td>
<td>Last day of previous week</td>
<td>2013/04/21</td>
</tr>
<tr>
<td>$FDOPM</td>
<td>First day of previous month</td>
<td>2013/03/01</td>
</tr>
<tr>
<td>$LDPM</td>
<td>Last day of previous month</td>
<td>2013/03/31</td>
</tr>
<tr>
<td>$FDOPY</td>
<td>First day of previous year</td>
<td>2012/01/01</td>
</tr>
<tr>
<td>$LDOPY</td>
<td>Last day of previous year</td>
<td>2012/12/31</td>
</tr>
<tr>
<td>$FDONW</td>
<td>First day of next week</td>
<td>2013/04/29</td>
</tr>
<tr>
<td>$LDONW</td>
<td>Last day of next week</td>
<td>2013/04/28</td>
</tr>
<tr>
<td>$FDONM</td>
<td>First day of next month</td>
<td>2013/05/01</td>
</tr>
<tr>
<td>$LDONM</td>
<td>Last day of next month</td>
<td>2013/05/31</td>
</tr>
<tr>
<td>$FDONY</td>
<td>First day of next year</td>
<td>2014/01/01</td>
</tr>
<tr>
<td>$FDONY</td>
<td>Last day of next year</td>
<td>2014/12/31</td>
</tr>
<tr>
<td>$TIMES</td>
<td>Time stamp now (offset in seconds)</td>
<td>n/a</td>
</tr>
<tr>
<td>$TIMEM</td>
<td>Time stamp now (offset in minutes)</td>
<td>n/a</td>
</tr>
<tr>
<td>$TIMEH</td>
<td>Time stamp now (offset in hours)</td>
<td>n/a</td>
</tr>
<tr>
<td>$TIMED</td>
<td>Time stamp now (offset in days)</td>
<td>n/a</td>
</tr>
<tr>
<td>$DELTA</td>
<td>Delta mode (timestamp of last run)</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Optionally, you can enter the <difference> as a positive or negative offset in days, using the following signs:

- + increments days, that is, the start date is moved into the future
- - decrements days, that is, the start date is moved into the past

4.2.1.1 Example

$TODAY-2: day before yesterday

$TODAY+2: day after tomorrow

You can combine the relative date selection in ranges, using the From and To fields of the selection criteria, for example, to define intervals.

4.2.1.2 Example

To count the amount of documents with a creation date in the last month, you enter the following selection criteria: <Field Name> = $FDOPM to $LDOPM.

For the ABAP source type, filter fields of the following data types support the syntax for relative dates:

<table>
<thead>
<tr>
<th>DDict Data Type</th>
<th>ABAP Data Type</th>
<th>Date Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATS</td>
<td>D</td>
<td>YYYYMMDD</td>
</tr>
<tr>
<td>CHAR 10</td>
<td>C 10</td>
<td>YYYYMMDD</td>
</tr>
<tr>
<td>CHAR 14</td>
<td>C 14</td>
<td>YYYYMMDDmmss</td>
</tr>
<tr>
<td>DEC 15</td>
<td>P 8</td>
<td>YYYYMMDDhhmmss</td>
</tr>
<tr>
<td>DEC 21</td>
<td>P 11</td>
<td>YYYYMMDDhhmmss,mmmuun</td>
</tr>
<tr>
<td>TIMS</td>
<td>C 6</td>
<td>hhmmss</td>
</tr>
</tbody>
</table>
Legend:

- YYYY = year
- MM = month
- DD = day
- hh = hour
- mm = minute
- ss = second
- mmmuun = milliseconds / microseconds / nanoseconds

Note: In data types containing time stamps (hhmmss), by default, 000000 is entered by the system. Exception: For start date keywords using a “Last Day” option (for example, $LDOCM) a maximum time stamp (235959) is set by the system. The keywords $TIMEx and $DELTA are filled with exact values.

You can also use relative date filtering for the ADBC source type. However, the supported date and time stamp data types depend on the used database management system.

For the $TIMEx and $DELTA keywords, the system calculates an exact time, and not only a date.

The following keywords use the current system time stamp at the time of data comparison: $TIMES, $TIMEM, $TIMEH, $TIMED. The system calculates exact time stamps using DEC 15 and DEC 21 data types (see table above). You can use different time units for the offset, for example, seconds, minutes, hours, or days.

$DELTA stores the time stamp of each data comparison run, and at the next run, it selects the data created since the last run, by entering >=$DELTA in the time stamp filter. You can enter an additional offset in minutes (+-<Difference>).

Both keywords support an additional <Difference> offset. For $DELTA, the entered value is interpreted as minutes.

The time stamp calculation is based on UTC time. If the used database table stores the time stamps in another time format, you can use the <Difference> operator to adjust the value.
5 DATA CONVERSION IN CROSS DATABASE COMPARISON

5.1 Conversion of comparison key fields

For comparison key fields, conversion might be required although the field contents have the same format in both systems. The reason is that during comparison, the correct sort order of the key fields is checked and the field contents are compared as string. So, for example, the numeric field contents 3, 4, 22, 100 have the correct sort order, but when copied into a string field, the sort order ‘3’, ‘4’, ‘22’, ‘100’ is wrong. You can use a suitable build-in conversion type like ‘INT: length x’ to fill the field contents with leading zeros up to x integer places to get the correct sort order like ‘003’, ‘004’, ‘022’, ‘100’.

5.1.1 Conversion of integer comparison key fields

For integer comparison key fields, conversion is required unless all field contents have the same number of integer places. For example, in the unlikely case of only field contents with two integer places, like 11, 21, 34, 54, no conversion is required, but in the usual case of field contents with a different number of integer places, like 3, 4, 22 and 100, conversion is required.

There are specific build-in conversion types for integers, all with a label starting with ‘INT’. There are conversion types to fill the field contents with leading zeros up to a fixed number of integer places like ‘INT: length 2’, ‘INT: length 3’, etc. and there are conversion types to fill the field contents with leading zeros up to the length specified for the data type in one of the source systems, ‘INT: length source 1’ and ‘INT: length source 2’.

5.1.1.1 Example

An integer key field has length 4 in source system 1 and length 5 in source system 2. As the length in source 2 is greater than the length in source 1, you could choose conversion type ‘INT: length source 2’ as in the following screenshot or alternatively ‘INT: length 5’ which would have the same effect in this example. Please note that conversion is required for both sources as otherwise there will be no leading zeros as required for the correct sort order.

All integer conversion rules also remove possible trailing blanks as ABAP will add a trailing blank during conversion of a positive integer into a string. (A negative integer will get a trailing minus sign.) There is also a special conversion type ‘INT: remove trailing blanks’ available to remove trailing blanks only.
5.1.1.2 Restriction

There is one restriction for integer comparison key fields: Integers with negative content cannot be used as comparison key fields as the correct sort order cannot be granted.

5.1.2 Conversion of decimal floating point number comparison key fields

For decimal floating point number comparison key fields, conversion is required unless all field contents have the same length and the same number of decimal places. For example, in the unlikely case of only field contents with length five and two decimal places, like 123.45, 234.56, 345.67, no conversion is required, but in the usual case of field contents with a different length and/or a different number of decimal places, like 123.4, 234, 345.678 and 9, conversion is required.

There are specific build-in conversion types for decimal floating point numbers, all with a label starting with ‘DEC’. For decimal floating point number comparison key fields, the conversion types that fill the length and decimal places with zeros or remove decimal places up to the length and number of decimals in one of the source systems are suitable. These rules are ‘DEC: length source 1, dec. s.1’ for the length and number of decimals in source system 1, ‘DEC: length source 1, dec. s.2’ for the length in source system 1 and number of decimals in source system 2, etc.

5.1.2.1 Example

A decimal floating point number key field has length 5 and 2 decimals in both source systems. This means that you could choose conversion type ‘DEC: length source 1, dec. s.1’ as in the following screenshot or alternatively all other ‘DEC: length source x, dec. s.x’ which would have the same effect in this example. Please note that conversion is required for both sources as otherwise there will be no leading and trailing zeros as required for the correct sort order.

5.1.2.2 Restriction

There is one restriction for decimal floating point number comparison key fields: Decimal floating point numbers with negative content cannot be used as comparison key fields as the correct sort order cannot be granted.
5.1.3 Conversion of UUID/GUID comparison key fields

For UUID/GUID comparison key fields, conversion is not required as long as the UUID/GUID format is either binary 16 or character 32. These UUID/GUID formats are automatically extracted as character 32 UUIDs/GUIDs and the correct sort order should be given.

There are specific built-in conversion types for character 22 or character 26 UUID/GUID formats, all starting with ‘UUID’.

5.2 Conversion of comparison fields (no keys)

Comparison data fields that are not used as comparison key fields do not necessarily require conversion as there is no such thing like the correct sort order checked. Conversion is only required if there are specific mapping rules between the fields.

5.2.1.1 Example 1

When data is transferred from a field in source system 1 to a field in source system 2, the third of three decimal places is cut off. You can use the built-in conversion type ‘DEC: decrease to 2 decimals’ for source system 1 in this case to cut off the third decimal place of the field content in source system 1.

5.2.1.2 Example 2

Data in a field in source system 1 is case sensitive whereas the data in the corresponding field in source system 2 is in upper case letters only. You can use the built-in conversion type ‘CASE: convert into upper case’ for source system 1 in this case to convert the field content in source system 1 to upper case.
6 CROSS DATABASE COMPARISON FOR XML FILES

The CDC source type FIXS for XML files on application server requires either XML files directly in asXML format or XML files in any different format and a corresponding transformation to bring the required data from these files into asXML format (as of Solution Manager 7.1 SP12). The CDC source type FIXL for XML files on local PC requires XML files in asXML format.

asXML is the format that you get if you use transformation ID for an internal ABAP table. The following is an example for one item with columns PARTNER, NAME_LAST and NAME_FIRST. (For a table with multiple items, you would see multiple <item> tags.)

The XML declaration at the beginning of the file is optional, whereas the namespace definition needs to refer to http://www.sap.com/abapxml version="1.0". The part of the XML file containing the data to be compared needs to be included in tag <asx:values> (and </asx:values> respectively).

Beyond this level a tag <TABLE> marks the start of the data table, whereas several tags opening and closing with <item> (and </item> respectively) mark the beginning and the end of a line item in the table to be compared.

Within the <item> tags the names of the respective table fields need to be used as tag names. These tags contain the actual item data.

Furthermore it is important that the tags in the XML file in asXML format must correspond to the tags in the mapping connection of the CDC data model.

6.1.1.1 Example

ABAP table:

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jason</td>
<td>Li</td>
<td>Shanghai</td>
</tr>
<tr>
<td>Peter</td>
<td>Meier</td>
<td>Berlin</td>
</tr>
<tr>
<td>Michael</td>
<td>Smith</td>
<td>Washington</td>
</tr>
</tbody>
</table>
Converted into asXML this table would look like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <ITEM>
      <FIRST_NAME>Jason</FIRST_NAME>
      <LAST_NAME>Li</LAST_NAME>
      <CITY>Beijing</CITY>
    </ITEM>
    <ITEM>
      <FIRST_NAME>Peter</FIRST_NAME>
      <LAST_NAME>Meier</LAST_NAME>
      <CITY>Berlin</CITY>
    </ITEM>
    <ITEM>
      <FIRST_NAME>Michael</FIRST_NAME>
      <LAST_NAME>Smith</LAST_NAME>
      <CITY>Washington</CITY>
    </ITEM>
  </asx:values>
</asx:abap>
```

In the definition of the comparison object it should like this:

```
+------------------+
| Source System 2   |
+------------------+
| ZBBP_PDBEH       |
|                  |
| >>>              |
|                 |
| FIRST_NAME       |
|                 |
| LAST_NAME        |
|                 |
| CITY             |
|                  |
+------------------+
```

(Here “ZBBP_PDBEH” is used as table name.)
7 CONSIDERATIONS FOR CDC SYSTEM LANDSCAPE

7.1 Overview on System and Transport Landscape

In the context of generating extractor functions, it is important to understand the system landscape, in which CDC is used. Regarding Solution Manager there is typically either a two- or three-system landscape. For CDC it would not really matter whether it runs on the development (DEV), test (QA) or productive (PRD) Solution Manager, as long as the required connections (like RFC destinations or database connections) to the intended source systems are available. Normally you would use CDC on the development Solution Manager only for initial testing, like evaluating the features, building a proof-of-concept, testing custom-enhancements, and so on, with connections to the development or test source systems. For real productive usage, which means comparing data from the productive source systems, CDC runs on the productive Solution Manager obviously.

Please note that it is not possible to transport comparison objects and comparison instances, because they are considered as application data and would also use different connection parameter values depending on the environment. However, it is possible to use a download/upload function for comparison object data models or whole comparison objects. But typically you start building new comparison objects directly in the productive Solution Manager and test them with connections to development/test source systems, before running high-volume comparisons between the productive source systems. This way you only need one comparison object for modeling and you can have multiple comparison instances pointing to the different environments.

7.2 System and Transport Landscape at CDC design-time (for source type ABAP „RFC to SAP ABAP System“)

At design-time, which is the modeling of comparison objects, you need two types of RFC destinations to the development source system.

- A Read-RFC-Destination, which uses a background connection with a system logon user, to access the data dictionary for providing value helps on tables and fields.
- Generated RFC-Destinations (e.g. by Solution Manager Setup) of this type are named SM_<sid>CLNT<clnt>_LOGIN, where <sid> is the System-ID and <clnt> is the client number.

- A Trusted-RFC-Destination, which uses a connection with the same logon user, to generate the specific extractor functions. For this code generation you need developer authorization to create the corresponding function group and the remote-enabled extractor function modules.
  - Generated RFC-Destinations of this type are named SM_<sid>CLNT<clnt>_TRUSTED, where <sid> is the System-ID and <clnt> is the client number.

Afterwards the generated extractor function modules are transported from the development system to the test system to the productive system, by standard transport management tools. It is not possible to generate extractor coding directly in the productive system, because it does not allow repository changes.

To overcome the need of generation and transport, as of Solution Manager 7.1 SP10 you can use the source type “SAP ABAP System (using RFC to generic extractor)”. 
7.3 System Landscape at CDC run-time for Data Extraction (for source type ABAP „RFC to SAP ABAP System“)

At run-time, which is the actual extraction and comparison of data, you just need a Read-RFC-Destination, which in this case needs authorization to read from the respective application database tables. Typically you compare the data from the productive source systems, using the transported extractor function modules. Or alternatively you can use a generic extractor already present inside the productive source systems.
7.4 System Landscape at CDC design-time (for source type ADBC „Remote Database“)

For data sources using a remote database connection, there is no transporting required. During design-time the database connection is used to query the data dictionary, in order to provide search helps for tables and fields. Based on the designed comparison object model, two SQL statements are generated, one for doing a count (to estimate the number of expected records) and one for doing the actual data extraction. These SQL statements are stored centrally on Solution Manager as comparison object parameter values.
7.5 System Landscape at CDC run-time for Data Extraction (for source type ADBC „Remote Database“)

At run-time, the generated SQL statement is used to extract data from the productive non-ABAP database.
8 MASS PROCESSING/MASS MAINTENANCE

With Solution Manager 7.1 SP10 you can use a guided self-service to generate comparison objects from the SLT repository for comparison of data in an ABAP source system to the data replicated to HANA.

### 8.1.1.1 Procedure

Choose **Comparison Object** → **Mass Maintenance** → **Start Mass Creation from SLT Replication Model for HANA**

You can find detailed description in chapter 3.2.8 *Generation of Comparison Object from SAP Landscape Replication Server (SLT) repository*

With Solution Manager 7.1 SP12 there are further functionalities to handle Mass Maintenance of Comparison Objects:

**General Mass Creation of Comparison Objects**: you choose this for creating several comparison objects at the same time.

### 8.1.1.2 Procedure

Choose **Comparison Object** → **Mass Maintenance** → **Start General Mass Creation of Comparison Objects**

- **Extractors to be generated for Source** (new in Solution Manager 7.1 SP12): Both Source Systems ("both side" of comparison object), Source System 1 ("left side" of comparison object), Source System 2 ("right side" of comparison object), No Source System
- **Include client field**: Specify if the client field should be included in the comparison.
- **Prefix for Comparison Object name**: Specify a prefix that – together with the table name – is used as comparison object name.
- **Source Type 1**: Specify the data source for Source 1
- **Source Type 2**: Specify the data source for Source 2

Choose **Next**

- Select the tables for which comparison object should be created.

Choose **Next**

- **Maintain Comparison Object Parameter and Data Model**

Choose **Next**

- You will have an overview about the selected tables and corresponding comparison objects which you have to submit and to create the comparison objects.

**Mass Generation of Extractors**: you can generate the extractors by marking several comparison objects from the list at once.

### 8.1.1.3 Procedure

Choose **Comparison Object** → **Mass Maintenance** → **Start Mass Generation of Extractors**

- **Extractors to be generated for Source** (new in Solution Manager 7.1 SP12): Both Source Systems ("both side" of comparison object), Source System 1 ("left side" of comparison object), Source System 2 ("right side" of comparison object), No Source System

Choose **Next**

- Mark the comparison objects from the list

Choose **Next**

- You will have an overview about the selected comparison objects which you have to submit and to generate the extractors.
**Mass Change of Comparison Object Parameters:** you can execute the change of several comparison objects parameter at once.

**8.1.1.4 Procedure**

Choose ➤ **Comparison Object** ➔ **Mass Maintenance** ➔ **Start Mass Change of Comparison Object Parameters**

- **Extractors to be generated for Source** (new in Solution Manager 7.1 SP12): Both Source Systems ("both side" of comparison object), Source System 1 ("left side" of comparison object), Source System 2 ("right side" of comparison object), No Source System
- **Source Type 1:** Specify the data source for Source 1
- **Source Type 2:** Specify the data source for Source 2

Choose ➤ **Next**

- Mark the comparison objects from the list to be changed

Choose ➤ **Next**

- Maintain Comparison Object Parameter and Data Model

Choose ➤ **Next**

- You will have an overview about the changed comparison objects which you have to submit and to generate the extractors.

**Deletion of Comparison Objects:** you choose this if you want to delete several comparison objects in mass processing.

**8.1.1.5 Procedure**

Choose ➤ **Comparison Object** ➔ **Mass Maintenance** ➔ **Start Deletion of Comparison Objects**

- Mark the comparison objects to be deleted from the list

Choose ➤ **Next**

- You will have an overview about the selected comparison objects which you have to submit and to delete

With Solution Manager 7.1 SP12 there are further functionalities to handle Mass Maintenance of Comparison Instances:

**Generate Comparison Instances for Comparison Objects (1:1):** you choose this if you want to generate comparison instances for comparison objects

**8.1.1.6 Procedure**

Choose ➤ **Comparison Instance** ➔ **Mass Maintenance** ➔ **Start Generate Comparison Instances for Comparison Objects (1:1)**

- Choose Comparison Object from the list for that you want to generate the Comparison Instance

Choose ➤ **Next**

- Configure Comparison Instance

Choose ➤ **Next**

- Optionally Maintain Comparison Group

Choose ➤ **Next**

- You will have an overview about Comparison Instances which you have to submit and to generate

**Generate Multiple Comparison Instances for a Single Comparison Object (n:1):** you choose this if you want to generate multiple comparison instances for a comparison object
8.1.1.7 Procedure
Choose Comparison Instance → Mass Maintenance → Start Generate Multiple Comparison Instances for a Single Comparison Object (n:1)

- Add names of the Comparison Instances to be created
Choose Next
- Configure the new Comparison Instances
Choose Next
- Optionally Maintain Comparison Group
Choose Next
- You will have an overview about Comparison Instances which you have to submit and to generate
9 BLOCK SIZE

As of Solution Manager 7.1 SP12, there are three ways to define the extraction block size.

1. Field “Comparison Block Size” in comparison object (CDC WebDynpro UI):
   - default for all comparison instances and both source systems
2. Fields “Alternative Block Size Source 1 / 2” in comparison instance (CDC WebDynpro UI):
   - individual setting per comparison instance and each source system
3. Fields “Alternative Block Size Source 1 / 2” in SAPGUI Transaction DSWP_CDC_START:
   - individual settings for a single comparison instance run

During and after the comparison run you can check which block sizes have been used. There is a new line at the very bottom of the result detail screen at “Comparison Runtime Statistics”:

<table>
<thead>
<tr>
<th>Block size (System 1 / System 2)</th>
<th>Blocks extracted (System 1 / System 2 / identical)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000 / 10,000</td>
<td>150 / 190 / 190</td>
</tr>
</tbody>
</table>

With “Blocks extracted / identical” you can now also see whether the binary block comparison found totally identical blocks.