## Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Getting Started. .................................................. 3</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview of the Native DataStore Object. .......................... 3</td>
</tr>
<tr>
<td>1.2</td>
<td>Architecture and Tables. ........................................ 5</td>
</tr>
<tr>
<td>1.3</td>
<td>Runtime Aspects. .................................................. 6</td>
</tr>
<tr>
<td>1.4</td>
<td>Design Time Aspects. ............................................. 7</td>
</tr>
<tr>
<td>2</td>
<td>Security Aspects. .................................................. 8</td>
</tr>
<tr>
<td>2.1</td>
<td>Data Privacy and Protection. ...................................... 8</td>
</tr>
<tr>
<td>3</td>
<td>Working with Native DataStore Object. ............................ 10</td>
</tr>
<tr>
<td>3.1</td>
<td>Creating Native DataStore Object. .................................. 10</td>
</tr>
<tr>
<td>3.2</td>
<td>Modeling Native DataStore Object. .................................. 13</td>
</tr>
<tr>
<td>3.3</td>
<td>Preparing Data Load and Privileges for Data Set Activation. ....................... 15</td>
</tr>
<tr>
<td>3.4</td>
<td>Building Native DataStore Object. .................................. 16</td>
</tr>
<tr>
<td>3.5</td>
<td>Managing Native DataStore Object. .................................. 17</td>
</tr>
<tr>
<td>3.6</td>
<td>Creating and Connecting a Flowgraph to Native DataStore Object. .................. 20</td>
</tr>
<tr>
<td>3.7</td>
<td>Executing Flowgraph. ................................................ 22</td>
</tr>
<tr>
<td>3.8</td>
<td>Activating Data Set and Checking Data. ................................ 23</td>
</tr>
<tr>
<td>3.9</td>
<td>Native Data Store Default Values. .................................... 24</td>
</tr>
<tr>
<td>3.10</td>
<td>Use Cases. .......................................................... 26</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Record Mode. ...................................................... 27</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Use Case 1 - NDSO (Initial and Delta Data Load) ......................... 28</td>
</tr>
<tr>
<td>3.10.3</td>
<td>Use Case 2 - NDSO → NDSO (Initial and Delta Data Load) ............... 30</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Use Case 3 - NDSO → NDSO (Delete a Load Request (activationID)) .......... 33</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Use Case 4 - NDSO (Integration with SAP HANA Calculation View) .......... 34</td>
</tr>
</tbody>
</table>
1. Getting Started

The Administration Guide for SAP HANA Data Warehousing Foundation - Native DataStore Object describes the tasks and concepts for managing the merge processes of delta and full data load requests into their reportable content.

The guide covers the following areas:

- **Getting Started [page 3]**
  - Overview of the Native DataStore Object [page 3]
  - Architecture and Tables [page 5]
  - Runtime Aspects [page 6]
  - Design Time Aspects [page 7]
  - Security Aspects [page 8]
    - Data Privacy and Protection [page 8]
- **Working with Native DataStore Object [page 10]**
  - Creating Native DataStore Object [page 10]
  - Modeling Native DataStore Object [page 13]
  - Preparing Data Load and Privileges for Data Set Activation [page 15]
  - Building Native DataStore Object [page 16]
  - Managing Native DataStore Object [page 17]
  - Creating and Connecting a Flowgraph to Native DataStore Object [page 20]
  - Executing Flowgraph [page 22]
  - Activating Data Set and Checking Data [page 23]
  - Native Data Store Default Values [page 24]
- **Use Cases [page 26]**
  - Record Mode [page 27]
  - Use Case 1 - NDSO (Initial and Delta Data Load) [page 28]
  - Use Case 2 - NDSO → NDSO (Initial and Delta Data Load) [page 30]
  - Use Case 3 - NDSO → NDSO (Delete a Load Request (activationID)) [page 33]
  - Use Case 4 - NDSO (Integration with SAP HANA Calculation View) [page 34]

For information about administration of the overall SAP HANA system, including information about table partitioning, and SAP HANA table redistribution, refer to the SAP HANA Administration Guide.

1.1 Overview of the Native DataStore Object

The objective of the native datastore object (NDSO) is to provide a central persistence object with the same capabilities of the standard SAP Business Warehouse (BW) advanced datastore object (ADSO).

For a short introduction, feel free to take a look at DWF: Native DSO - Introduction (Part I) recording available on the HANA Academy Data Warehouse Foundation Channel:
About the Native DataStore

- The NDSO supports the data merge process of delta and full data load requests into their reportable content. The NDSO data merge processes are triggered by the NDSO activation step.
- In addition, it provides the capability to identify and process the delta dataset (modified dataset compared to the previous data load request) for connected NDSOs.
- Measures and attributes of the NDSO can be merged by a simple move or an aggregation step and the NDSO supports the process to handle records that are marked for deletion within the delta data sets processing. Attribute columns are of type MOVE. Measures columns could be of type MOVE or AGGREGATION.
- The NDSO is capable to process records based on the source data specified RECORDMODE values (like: INSERT / UPDATE / DELETE):
  - Simplified Flowgraph design.
    - No need to design specific INSERT / UPDATE / DELETE handling within the flowgraph, as the NDSO manages the individual data processing based on the RECORDMODE values.
- The NDSO offers multiple ways to interoperate between a SAP HANA SQL (native) data warehousing and SAP HANA Business Warehouse systems, which is equal on how the data is processed within a single SAP BW system. The SAP BW ADSO can identify and process the delta data set from the NDSO.
- The NDSO is based on the SAP HANA XS advanced application server framework, which includes SAP HANA Development Infrastructure (DevX), SAP HANA Web IDE, SAP HANA Deployment Infrastructure (HDI) and SAP HANA Core Data Services (HANA CDS).
- The NDSO is an addition to the SAP HANA certified ETL/ELT tools, and not a replacement and functions with the HANA certified ETL tools. The NDSO provides SAP HANA additional integrated technical services, like delta data set processing or request management, which would need to be created, tested, validated and optimized manually within the according ETL/ELT tool capabilities.
- Native datastore objects will be represented as annotated CDS contexts, this includes all tables (meta and metadata). The surrounding context serves as the "datastore object".

NDSO Capabilities:

- Handle insertion, update, and deletion of records (based on the provided RECORDMODE)
- Handle delta data processing to connected data targets incl. NDSO objects
- Handle multiple inbound queues (NDSO InboundQueue) to load data from different sources
- Provide an active/reportable data table (NDSO Active Table) for e.g. query calculation view access
- Keep track of data modifications (NDSO ChangeLog) to support the:
  - rollback of a request (incl. the complete dataset associated to the request)
  - delta data processing to connected data targets
- Simplify the Flowgraph design (insertion, update, and deletion operations are controlled by the record specific RECORDMODE value)
Record Mode

The RECORDMODE provides a description of how the individual records are processed and managed. There are five options available:

- 'N': The record delivers a new image (Insert)
- 'A': The record delivers an additive image (Update)
- 'D': The record has to be deleted (Delete)
- 'R': The record delivers a reverse image
- 'X': The record delivers a before image

For more information, see Record Mode [page 27].

1.2 Architecture and Tables

A native datastore object consists of 3 data and 7 metadata tables. The following graphic displays the overview of the standard datastore object with 3 data tables:

DataStore Object
Data Tables:

**Inbound Queue Table (IQ)**

The *Inbound Queue* table is leveraged to store the recently loaded data. Multiple *Inbound Queue* tables are supported, which could be leveraged in a use case to handle a header or item scenario in a single NDSO, by feeding header and item data using different inbound queues/data sources.

**Active Data Table (AT)**

The data in the *Inbound Queue* table need to be activated to be prepared for reporting and further processing. The data gets transferred into the *Active Data* table during the *Data Activation* process, where the data aggregation (measures) and data move (attributes) take place.

**Change Log Table (CL)**

The content of the *Change Log* table is maintained in parallel, to track the data modifications (changes) and delta (modified) records. The delta data set is derived out of the *Change Log* table in case multiple NDSOs or ADSOs get connected via a SAP HANA EIM, SDI Flowgraph or a BW Transformation and only the modified (delta) data set is processed.

Metadata Tables:

The NDSO also consists of metadata tables storing the:
- aggregation history, to ensure a consistent rollback of requests,
- last used request ID,
- messages,
- overall operations that were performed in a particular NDSO,
- overall registered subscribers that are needed to ensure a consistent delta download to all connected data targets.

1.3 Runtime Aspects

The Manage-UI as part of HRTT (HANA Runtime Tools/Database Explorer) is the central tool for managing loads and activations, housekeeping operations (clean-up change log, delete all), subscriber management, development tools (e.g. generate data), and selective deletion.
Tools supporting managed loads, like EIM Flowgraph, write metadata of the corresponding load into the metadata tables of the NDSO. This enables the Manage-UI to show the details of the load like “number of records”, “status”, etc.

A database administrator can then activate the request to make the data available for reporting and to calculate the delta information for the change log. The activation process is built as a HANA core module in the HANA kernel.

A failed activation can be rolled back. The active data is then reconstructed from the change log.

### 1.4 Design Time Aspects

A native datastore object is represented as an annotated **CDS** (Core Data Services) context. Each table (data and metadata) is exposed as a CDS entity. Tools like the **SAP EIM FlowGraph** can use these entities to load data into or extract data from an NDSO. The graphical table editor offers a convenient way to use these annotations. The annotations are automatically generated but can be changed in the **Web IDE** text editor. The **HDI/CDS** plug-in prevents activation in case of mismatches in the annotations (e.g. wrong structure in the referenced change log entity).

The annotation definition can be found in the DWF project template in the Web IDE (new project from DWF template → module database → folder src/DataStore).
2 Security Aspects

2.1 Data Privacy and Protection

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with general data privacy regulation, it is necessary to consider compliance with industry-specific legislation in different countries. SAP provides specific features and functions to support compliance with regards to relevant legal requirements, including data protection. SAP does not give any advice on whether these features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, this information does not give any advice or recommendation in regards to additional features that would be required in particular IT environments; decisions related to data protection must be made on a case-by-case basis, under consideration of the given system landscape and the applicable legal requirements.

**Note**

In the majority of cases, compliance with applicable data protection and privacy laws will not be covered by a product feature. SAP software supports data protection compliance by providing security features and specific data protection-relevant functions, such as deletion of personal data. SAP does not provide legal advice in any form. Definitions and other terms used in this document are not taken from any given legal source.

Native datastore object is designed to work as a data container. Native datastore object modeling application is a data agnostic application, where no personal data is collected on purpose. The user defines the structure of the native datastore object, thus he or she defines the columns and the types. The application is not aware of which columns may hold personal data. The application is designed to work as a data container and doesn’t provide a UI to explore, modify or delete data. If personal data will be stored then the user needs to implement the respective functionality.

**Read Access Logging**

As there is no notion of personal data in the Data Warehousing Foundation layer (data-agnostic), if read access need to be logged, there is an auditing mechanism in SAP HANA which can be used. For more information about auditing activity in the SAP HANA database, see the SAP HANA Administration Guide in the SAP HANA Platform 2.0 documentation at [https://help.sap.com/viewer/product/SAP_HANA_PLATFORM/](https://help.sap.com/viewer/product/SAP_HANA_PLATFORM/).
Information Report

The application is designed to work as a data container (there is no notion of personal data) and doesn’t provide a UI to explore data. If personal data will be stored then the user needs to implement the reporting functionality.

Deletion of Personal Data

As the application is a data agnostic application, the user has to implement the functionality to pseudonymize personal data. This can be done by pipelining native datastore objects to mask or anonymize the personal data, and deleting the change log history of the original object.

Logging Changes

There is no notion of personal data in the native datastore object. All changes performed in the active data table are reflected in the change log table. However, if additional logging is required then the user needs to implement the necessary functionality. SAP HANA functionality can be used for this purpose.
3 Working with Native DataStore Object

This section explains the NDSO workflow from creating to activating data sets in a simple step-by-step procedure.

When creating a project, be sure to use the Data Warehousing project template that contain all necessary modules.

More Information

- Search for "Create a Data Warehousing Project" at https://help.sap.com/viewer/p/SAP_HANA_DATA_WAREHOUSING_FOUNDATION
- Creating Native DataStore Object [page 10]
- Use Cases [page 26]

3.1 Creating Native DataStore Object

This section describes a step-by-step procedure to create an NDSO after it has been installed.

Prerequisites

- You have SAP HANA XS Advanced running on SAP HANA 2.0 or higher.
- You downloaded and installed the NDSO component. To install the component, see the SAP HANA Data Warehousing Foundation Installation Guide for XS Advanced Components.
- You installed and set up (e.g., assigned users to spaces for deployment) SAP HANA Web IDE. For more information on installation and setup, please visit the SAP Web IDE for SAP HANA - Installation and Upgrade Guide.

Note

Data Warehousing Foundation Project Template

To obtain a database module with all the necessary requirements for your NDSO, you can simply install the DWF project template. For more information, see Data Warehousing Foundation Project in the SAP HANA Data Warehousing Foundation - Data Warehouse Scheduler Administration Guide (XSA).

The database module in the DWF project template contains:

- a full function NDSO specification,
- a source-table definition (including a data set) to be connected later via Flowgraph to process the dataset to the NDSO, and
To create a new (local) project without the DWF project template:

**Procedure**

1. **Logon** to the *SAP HANA Web IDE*.
2. Create a new project by selecting a multi-target application project template and name it (here: Sales).
3. Select an appropriate space (here: DEV) and then click *Finish*.

![New Multi-Target Application Project](image)

**MTA Details**

- **Application ID**
- **Application Version**
- **Description**
- **Space**

> Previous  Next  Finish
4. Right-click the project and create an HDB module. Use the CDS editor to model the NDSO. Check the *Create DataStore Artifacts* box. This is necessary to create an NDSO.

Datastore specific annotations need to be in the same folder as the NDSO model file.

When creating a new project from scratch, you need to leverage the .hdbrole and .hdbsynonym files that are provided in the DWF project to successfully activate a loaded data set.

The following static content files are then created in the src-folder within the HDB module:

- Annotations.hdbcds
- Structures.hdbcds
- Types.hdbcds

The next logical step would be to model the NDSO. Please see *Modeling Native DataStore Object* [page 13].

**Note**

Refrain from changing these objects:
3.2 Modeling Native DataStore Object

Procedure

1. Select the project, navigate to the `scr folder` and right-click `New`, then click `CDS Artifact`.
2. Specify the `<Content Area>` of the HANA Core Data Services Artifact (HANA CDS) (here: SalesOrderHeader) and click `Create`.

A new `<Content Area Name>.hdbcds` file will be created (here: SalesOrderHeader.hdbcds).

The **Graphical CDS Editor** starts up and a new icon becomes available to graphically design the NDSO. The icon is only visible once the NDSO component has been installed:
3. Click the NDSO and specify the NDSO name.

4. Double-click to open the Details Screen. Here, Elements and columns can be added and then specified. For more information, see the Native Data Store Default Values [page 24].

5. Switch to the NDSO Details Tab (Properties).
   a. Check the Write Change Log box – if the change log (delta) information should be captured or are required for further delta data set processing.
   b. Specify the different and multiple inbound queues to support header/item data sets to be processed within a single NDSO.
   c. Check the details and columns of the individual inbound queues.
   d. Measure data management within NDSO changelog and inbound queue.

   Aggregation Behavior: Set Measure “Aggregation behavior” to “Sum” to manage the Aggregation behavior for Rollback activities.

   Inbound Queue: Set Measure “Update Mode” behavior to “Sum” for the Inbound-Queue Table to manage multiple Records for identical PK and different Recordmode settings within a single dataset.

   Aggregation Behavior + Inbound Queue measure Aggregation behavior settings should be identical.

   **Support Snapshot:**

   Once the snapshot setting has been flagged, only modified records are processed into the NDSO. If the NDSO is a staging layer type, where only full loads are processed into, only the delta records are processed into the inbound queue table. Request rollback and deletion are still supported.

   The configuration of the NDSO is complete and ready for the build process. For more information on building the NDSO, see Building Native DataStore Object [page 16].
3.3 Preparing Data Load and Privileges for Data Set Activation

To ensure that the NDSO functions, privileges must be granted to the user.

Install the DWF project template that contains an HDB module with:

- the source table definition (incl. data set) to be connected later via Flowgraph to process the dataset to the NDSO, and
- the required .hdbrole and .hdbsynonym files to successfully activate the loaded data set.

Example

For a step-by-step procedure on preparing a data load and granting privileges, feel free to take a look at DWF: Native DSO - Preparing for Data Load (part3/4) recording available on the HANA Academy Data Warehouse Foundation Channel:
3.4 Building Native DataStore Object

To create a build of the native datastore object, follow the instructions below:

Procedure

1. Select the project in the SAP HANA Web IDE local workspace.
2. Click Build from the pull-down menu of the SAP HANA Web IDE for the Hana Deployment Infrastructure (HDI) builder to start creating the respective HANA database artifacts.

Results

After the build has been successfully completed, the NDSO-mtar archive file will be created within the mta_archive archive folder of the SAP HANA Web IDE local workspace.
3.5 Managing Native DataStore Object

The Manage-UI as part of HRTT (HANA Runtime Tools/Database Browser) is the central tool for managing loads and activations of the NDSO.

To manage the NDSO, switch to the HANA Runtime Tools (HRTT).

Then select and connect to the according HDI container, which was previously generated.

The NDSO object is visible within the DataStores section of the lower screen once you click the section:

The NDSO Manage-UI pops up once you click the NDSO object.

The primary use-case is to:

- monitor the data load status,
- react on error-situations (e.g. resolving the error situation or removing a certain dataload) and,
- check and validate the generated table structures.
Within the **Advanced** tab, it’s possible to switch between:

- **Upload File**, to load CSV files into the NDSO inbound table
- **Generate Data**, to generate test data to be loaded into the NDSO inbound table
- **Insert as Select**, to load data from an available HANA object into the NDSO inbound table (SQL `INSERT` statement will be generated to be adjusted to fulfill your needs).

**Note**

The data is generated randomly based on the respective column data types. There is no validation and, thus, no guarantee that the data can be activated. The **Generate Data** functionality generally serves for testing purposes only.

The **Manage** tab provides the NDSO request info on loaded data sets. Here, it’s also possible to activate the data sets (select the certain request), once a data set has been loaded (see also the section on **Creating and Connecting a Flowgraph to Native DataStore Object** [page 20] in the **SAP HANA Data Warehousing Foundation - Native DataStore Object Administration Guide**).
Click the **Inbound Table, Active Data Table or Change Log Table** links to have a separate tab opened displaying the structure and content.

The dependencies between the load request and activation request are visible by clicking on the icon, which is also reflected in the corresponding requests section in the lower part of the screen.

**Housekeeping**

In the housekeeping tab, you can perform checks on the NDSO consistency as well as clean metadata, based on a date or request. You can also clean change logs, to delete the content of the change logs based on a date or a request.

**Subscription**

The subscriptions show all connected data sources (NDSOs connected to other NDSOs). They act as a delta pointer, and prevent data that cannot be extracted from being deleted.

If new data is loaded into the first NDSO (1), with a new activation ID, and this new data is processed into a second NDSO (2), it is possible to only process the delta data from NDSO 1 to NDSO 2 based on the request ID of the activation in the subscription.

If you extract via a Flowgraph with an NDSO as your source, a subscriber will automatically be created and the max request ID will be updated. The subscriber name receives the same name as the Flowgraph name.

**Tables**

Within the Tables section, all created tables of the NDSO on the database as well as all NDSO management tables are visible.
3.6 Creating and Connecting a Flowgraph to Native DataStore Object

In this section, a step-by-step procedure explains how to create and connect a Flowgraph to the NDSO.

Context

Creating the Flowgraph:

Procedure

1. Select the project and navigate to the src folder – or create a new folder within the src folder to locate the flowgraph object in.
2. Right-click New, then Flowgraph to start the Flowgraph Editor.
3. Click the blue + symbol to add a data store and data target.
4. Connect the data source and the data target.

5. Click the **configuration** icon to specify the object details of the data source object.
   a. Select a source object (here: the HANA object table from the GitHub repository).
   b. Check the object structure.

6. Click the **configuration** icon to specify the object details for the data target object.
   a. Select the previous specified NDSO data store and click **OK**.
   b. Check the object structure and then click **Apply**.

7. Arrange the Flowgraph structure by clicking the **Auto-Arrange** Icon.

8. Connect the data source object with the **Data Target (NDSO) Projection - "In"** object.

9. Click the **Data Target (NDSO)** projection icon to validate the field mappings.

10. Click the icon of the **Data_Target_addRecordNo** object to add a new field **RecordMode** (Length 1), to flag the changes of the data flow. Within the mapping, specify a value like: N for New, U for Update or D for Deletion.

11. Click **Apply** to complete the configuration.

12. Map the previously added field **RecordMode** to the **Data_Target_inbound_table** object.

13. Click the icon to map the field **technicalAttributes.recordmode** to the previous created field **RecordMode** and click **Apply** once more to complete the configuration.
14. Click **Save** and then select the Flowgraph from the workspace folder and right-click to perform a **Build Selected Files**. For more information on Flowgraphs see **Creating a Flowgraph** in the section **Flowgraphs** in the SAP HANA Administration Guide at https://help.sap.com/viewer/p/SAP_HANA_PLATFORM.

**Example**

See SAP HANA Academy Data Warehouse Foundation Channel:
- DWF: NDSO - Creating a Flowgraph to load data IV [2.0 SPS 02]
- DWF: NDSO - Creating a Taskchain to execute and monitor V [2.0 SPS 02]

### 3.7 Executing Flowgraph

Once the **Build** completed successfully, you need to execute the **Flowgraph** by clicking the **Run** icon to process the data from the **DataSource** to the **DataTarget**.

Alternatively, you can schedule its execution via the Data Warehouse Scheduler. For more information, see the SAP HANA Data Warehouse Scheduler Administration Guide at https://help.sap.com/viewer/p/SAP_HANA_DATA_WAREHOUSING_FOUNDATION.
3.8 Activating Data Set and Checking Data

Procedure

1. Switch to the the Database Explorer (HANA Runtime Tools, HRTT) to manage the NDSO (see section on Managing Native DataStore Object [page 17]).
   
   The Manage Tab provides the NDSO request info on loaded data sets.

2. Select a given load ID and click Activate to start the activation of data set records.

Results

The Highlight icon, Last Action Status and the Reporting flag will be adjusted, once the activation has been successfully completed.

The Activation Request will be listed to illustrate the successful completion of the activation process.
The activated data is now available within the NDSO active data table.

3.9 Native Data Store Default Values

View, change and apply the default values of your NDSO.

When creating an NDSO, the primitive data types of the data store objects are created by default and have the following values:
Default Values

<table>
<thead>
<tr>
<th>Primitive Data Types</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td></td>
</tr>
<tr>
<td>BinaryFloat</td>
<td>0.00</td>
</tr>
<tr>
<td>Boolean</td>
<td>false</td>
</tr>
<tr>
<td>Decimal</td>
<td>0.00</td>
</tr>
<tr>
<td>DecimalFloat</td>
<td>0.00</td>
</tr>
<tr>
<td>Double</td>
<td>0</td>
</tr>
<tr>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td>Integer64</td>
<td>0</td>
</tr>
<tr>
<td>LargeBinary</td>
<td></td>
</tr>
<tr>
<td>LocalDate (yyyy-MM-dd)</td>
<td>1900-01-01</td>
</tr>
<tr>
<td>LocalTime (HH:mm:ss)</td>
<td>00:00:00</td>
</tr>
<tr>
<td>String</td>
<td></td>
</tr>
<tr>
<td>UTCDatetime (yyyy-MM-dd HH:mm:ss)</td>
<td>1900-01-01 00:00:00</td>
</tr>
<tr>
<td>UTCTimestamp (yyyy-MM-dd HH:mm:ss)</td>
<td>1900-01-01 00:00:00</td>
</tr>
</tbody>
</table>

Changing Values

These values can be changed and saved.

1. In SAP Web IDE, go to your Preferences tab on the left and then select DataStore Default Values.
2. Change or enter new values and then click Save.
3. Apply these settings on your selected NDSO by clicking the Apply Default Value Preferences icon in the Details Screen of the NDSO.
4. Click **Apply** or override previous settings by entering new values and selecting the override flag.

**Related Information**

Modeling Native DataStore Object [page 13]

## 3.10 Use Cases

### Overview

**Use Case 1**: NDSO (Initial + delta data load)
- Initial + delta data load to a single NDSO
- NDSO data processing controller by specified source data RECORDMODE values
- NDSO granularity specified by key column on ACTIVE DATA TABLE

**Use Case 2**: NDSO -> NDSO (Initial + delta data load)
- Initial + delta data processing via 2 connected NDSOs
- NDSO data processing controller by specified source data RECORDMODE values
- NDSOs are configured with different granularity (key columns on ACTIVE DATA TABLE)

**Use Case 3**: NDSO -> NDSO (Delete a load request (activationID))
- Initial + delta data processing via 2 connected NDSOs
- NDSO data processing controller by specified source data RECORDMODE values
- NDSO are configured with different granularity (key columns on ACTIVE DATA TABLE)
- Delete the last load request from both NDSOs

**Use Case 4**: NDSO (Integration with SAP HANA Calculation View)
- Initial + delta data load to a single NDSO
• Integrate NDSO active data table with HANA Calculation View

Links

• Use Case 1 - NDSO (Initial and Delta Data Load) [page 28]
• Use Case 2 - NDSO → NDSO (Initial and Delta Data Load) [page 30]
• Use Case 3 - NDSO → NDSO (Delete a Load Request (activationID)) [page 33]
• Use Case 4 - NDSO (Integration with SAP HANA Calculation View) [page 34]
• Record Mode [page 27]

3.10.1 Record Mode

Definition

This indicator describes how a record is processed.

The characteristic values are as follows:

’N’: The record delivers a new image (Insert)
• The content of this record is the same as for an after image without a before image.
• When a record is created, a new image is transferred instead of an after image.
• The new image is complementary to the reverse image.

’A’: The record delivers an additive image (Update)
• For attributes that can be aggregated, only the change is transferred.
• For attributes that cannot be aggregated, the status after a record has been changed or created is transferred.
• This record can replace an after image and a before image if there are no non-aggregation attributes or if these cannot be changed.
• You can update the record into an InfoCube without restriction, but this requires an additive update into an NDSO.

’D’: The record has to be deleted (Delete)
• Only the key is transferred.
• This record can only be updated into an NDSO.

’R’: The record delivers a reverse image
• The content of this record is the same as the content of a before image.
• The only difference is with an NDSO update: Existing records with the same key are deleted.

’X’: The record delivers a before image
• The status is transferred before data is changed or deleted.
All record attributes that can be aggregated have to be transferred with a reverse +/- sign.

The reversal of the sign is carried out either by the extractor (default) or the service API.

In this case, the indicator 'Field is inverted in the cancelation field' must be set for the relevant extraction structure field in the datasource.

These records are ignored if the update is a non-additive update of an NDSO.

The before image is complementary to the after image.

'<'SPACE'>': The record delivers an after image (FULL Load Scenarios – Set Measure Aggregation: MOVE)

NDSO internal/generated Change-Log record values:

' ': The record delivers an after image

- The status is transferred after something is changed or added.

'Y': The record is an update image

- This kind of record is used in the change log of an NDSO in order to save the value from the update.
- This is for a possible rollback and roll-forward for key figures with minimum or maximum aggregation.
- This record also has the update value for characteristics (in this case, it is the same as the after image).
- Null values are stored for key figures with totals aggregation.
- An update image is only required when the value from the update is smaller or larger than the before image for at least one key figure with minimum or maximum aggregation.

3.10.2 Use Case 1 - NDSO (Initial and Delta Data Load)

1. Use Case 1: NDSO (Initial- + delta data load)

- Initial- + delta data load to a single NDSO
- NDSO data processing controller by specified source data RECORDMODE values
- NDSO granularity specified by key column on ACTIVE DATA TABLE

---

3.10.2 Use Case 1 - NDSO (Initial and Delta Data Load)
2.

**NDSO: SalesOrderHeader_Stage (Initial Load)**

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>RecordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>1</td>
<td>2010-10-03</td>
<td>00:00:00</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>2</td>
<td>2010-10-01</td>
<td>00:00:00</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2010-10-03</td>
<td>00:00:00</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-09-28</td>
<td>00:00:00</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
</tr>
</tbody>
</table>

**NDSO DATA ACTIVATION**

**ACTIVE DATA TABLE (AT)**

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2010-10-03</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2010-10-01</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2010-10-03</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2010-09-28</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

**CHANGE LOG (CL)**

<table>
<thead>
<tr>
<th>ActivityID</th>
<th>RecordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>1</td>
<td>2010-10-03</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>2</td>
<td>2010-10-01</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2010-10-03</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-09-28</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

3.

**NDSO: SalesOrderHeader_Stage (Delta Load)**

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>RecordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>1</td>
<td>2010-10-03</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>2</td>
<td>2010-10-01</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2010-10-03</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-09-28</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

**NDSO DATA ACTIVATION**

**ACTIVE DATA TABLE (AT)**

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2010-10-03</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2010-10-01</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2010-10-03</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2010-09-28</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

**CHANGE LOG (CL)**

<table>
<thead>
<tr>
<th>ActivityID</th>
<th>RecordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerName</th>
<th>ProductName</th>
<th>OrderQuan</th>
<th>SalesPrice</th>
<th>CustomerID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>1</td>
<td>2010-10-03</td>
<td>Barry Fine</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>2</td>
<td>2010-10-01</td>
<td>Cardinal</td>
<td>27,244.57</td>
<td>2000</td>
<td>20000</td>
<td>2013</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2010-10-03</td>
<td>Clay Rosedal</td>
<td>30,495.76</td>
<td>3000</td>
<td>30000</td>
<td>2011</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-09-28</td>
<td>Carlos</td>
<td>19,394.27</td>
<td>4000</td>
<td>40000</td>
<td>2010</td>
<td></td>
</tr>
</tbody>
</table>

**SAP HANA Data Warehousing Foundation - Native DataStore Object Administration Guide (XSA)**

**Working with Native DataStore Object**
3.10.3 Use Case 2 - NDSO → NDSO (Initial and Delta Data Load)

1. Use Case 2: NDSO → NDSO (Initial + Delta Data Load)
   - Initial and delta data processing via 2 connected NDSOs
   - NDSO data processing controller by specified source data RECORDMODE values
   - NDSOs are configured with different granularity (key columns on ACTIVE DATA TABLE)

   Source Data: Initial Load Dataset

2. Native DataStore Object (NDSO): NDSO → NDSO (Initial Load)
   - Initial data processing via 2 connected NDSOS
   - NDSO data processing controller by specified source data RECORDMODE values
   - NDSOs are configured with different granularity (key columns on ACTIVE DATA TABLE)
3. **NDSO: SalesOrderHeader_Stage (Initial Load)**

**INBOUND QUEUE (IQ)**

<table>
<thead>
<tr>
<th>RecordID</th>
<th>recordNo</th>
<th>recordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
<th>Customer/ProductID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>N</td>
<td>2</td>
<td>2010-10 12</td>
<td>Barry Frank Carriage</td>
<td>2</td>
<td>100</td>
<td>10000000</td>
<td>2010</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2011-07 13</td>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
<td>30000000</td>
<td>2011</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-10 12</td>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
<td>30000000</td>
<td>2010</td>
</tr>
</tbody>
</table>

**ACTIVE DATA TABLE (AT)**

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2010-10 12</td>
<td>Barry Frank Carriage</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>2011-07 13</td>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>2010-10 12</td>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
</tr>
</tbody>
</table>

**CHANGE LOG (CL)**

<table>
<thead>
<tr>
<th>activationId</th>
<th>recordNo</th>
<th>recordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
<th>Customer/ProductID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>N</td>
<td>2</td>
<td>2010-10 12</td>
<td>Barry Frank Carriage</td>
<td>2</td>
<td>100</td>
<td>10000000</td>
<td>2010</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>N</td>
<td>3</td>
<td>2011-07 13</td>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
<td>30000000</td>
<td>2011</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>N</td>
<td>4</td>
<td>2010-10 12</td>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
<td>30000000</td>
<td>2010</td>
</tr>
</tbody>
</table>

4. **NDSO: SalesOrderHeader (Initial Load)**

**INBOUND QUEUE (IQ)**

<table>
<thead>
<tr>
<th>RecordID</th>
<th>recordNo</th>
<th>recordMode</th>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
<th>Customer/ProductID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>N</td>
<td>Barry Frank Carriage</td>
<td>2</td>
<td>100</td>
<td>10000000</td>
<td>2010</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>N</td>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
<td>30000000</td>
<td>2011</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>N</td>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
<td>30000000</td>
<td>2010</td>
</tr>
</tbody>
</table>

**ACTIVE DATA TABLE (AT)**

<table>
<thead>
<tr>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barry Frank Carriage</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
</tr>
</tbody>
</table>

**CHANGE LOG (CL)**

<table>
<thead>
<tr>
<th>activationId</th>
<th>recordNo</th>
<th>recordMode</th>
<th>Customer/ProductName</th>
<th>OrderQty</th>
<th>Sales</th>
<th>Customer/ProductID</th>
<th>CalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>N</td>
<td>Barry Frank Carriage</td>
<td>2</td>
<td>100</td>
<td>10000000</td>
<td>2010</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>N</td>
<td>Clay Roper</td>
<td>3</td>
<td>300</td>
<td>30000000</td>
<td>2011</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>N</td>
<td>Carlos Soft Holmes H</td>
<td>4</td>
<td>300</td>
<td>30000000</td>
<td>2010</td>
</tr>
</tbody>
</table>
Native DataStore Object (NDSO): Use Case NDSO → NDSO (Delta Load)

- Delta data processing via 2 connected NDSOs
- NDSO data processing controller by specified source data RECORDMODE values
- NDSOs are configured with different granularity (key columns on ACTIVE DATA TABLE)

NDSO: SalesOrderHeader_Stage

1. Query
   - INBOUND QUEUE (IQ)
     - key loadID (Tech-A)
     - key recordMode (Tech-A)
     - SalesOrderID (A)
     - OrderDate (A)
     - CustomerNo (A)
     - CustomerName (A)
     - ProductID (A)
     - ProductName (A)
     - OrderQuantity (M)
     - Sales (M)
   - CHANGE LOG (CL)
     - key activationID (Tech-A)
     - key recordMode (Tech-A)
     - SalesOrderID (A)
     - OrderDate (A)
     - CustomerNo (A)
     - CustomerName (A)
     - ProductID (A)
     - ProductName (A)
     - OrderQuantity (M)
     - Sales (M)

2. Query
   - INBOUND QUEUE (IQ)
     - key CustomerNo (A)
     - key ProductID (A)
     - key CalYear (A)
     - key OrderID (A)
     - key ProductName (A)
     - ProductName (A)
     - OrderQuantity (M)
     - Sales (M)
   - CHANGE LOG (CL)
     - key activatonID (Tech-A)
     - key recordMode (Tech-A)
     - key recordID (Tech-A)
     - CustomerNo (A)
     - CustomerName (A)
     - ProductID (A)
     - ProductName (A)
     - OrderQuantity (M)
     - Sales (M)

3. Delta Load
   - NDSO: SalesOrderHeader_Stage (Delta Load)

<table>
<thead>
<tr>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerProductID</th>
<th>OrderQuantity</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2010-10-01</td>
<td>5807</td>
<td>1000</td>
<td>2000000</td>
</tr>
<tr>
<td>2</td>
<td>2011-07-10</td>
<td>340</td>
<td>3000</td>
<td>300000</td>
</tr>
<tr>
<td>3</td>
<td>2011-06-30</td>
<td>340</td>
<td>3000</td>
<td>300000</td>
</tr>
</tbody>
</table>

4. Delta Load
   - NDSO: SalesOrderHeader

<table>
<thead>
<tr>
<th>RecordMode</th>
<th>RecordMode</th>
<th>SalesOrderID</th>
<th>OrderDate</th>
<th>CustomerProductID</th>
<th>OrderQuantity</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2010-10-01</td>
<td>5807</td>
<td>1000</td>
<td>2000000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2011-07-10</td>
<td>340</td>
<td>3000</td>
<td>300000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2011-06-30</td>
<td>340</td>
<td>3000</td>
<td>300000</td>
</tr>
</tbody>
</table>

5. Delta Load
   - SalesOrderID: 1
     - Records available for Reporting
     - OrderQty: 10
     - Sales: 498.54
   - SalesOrderID: 2
     - Records available for Reporting
     - OrderQty: 5
     - Sales: 261.54
6. Delta Load
3.10.4 Use Case 3 - NDSO → NDSO (Delete a Load Request (activationID))

Use Case 3: NDSO → NDSO (Delete a Load Request (activationID))

- Delta data processing via 2 connected NDSOs
- NDSO data processing controller by specified source data RECORDMODE values
- NDSOs are configured with different granularity (key columns on ACTIVE DATA TABLE)
- Delete the last load request from both NDSOs
3.10.5 Use Case 4 - NDSO (Integration with SAP HANA Calculation View)
Use Case 4: NDSO (Integration with HANA Calculation View)

- Initial and delta data load to a single NDSO
- Integrate NDSO active data table with HANA Calculation View
Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon 🌐: You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP’s gross negligence or willful misconduct.
- Links with the icon 🌐: You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP’s gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.