PUBLIC

SAP Data Warehouse Cloud

First Guidance: Data Integration for ABAP Source Systems

Version: 2.1
February 2022

TABLE OF CONTENTS

1 AUDIENCE

This document serves as a guideline for everyone who is either connecting an ABAP system to SAP Data Warehouse Cloud or is about to create models on top of ABAP based system connections.

The Guideline is public because:

- It should serve as a general reference, for aspects concerning ABAP connections
- It should give technically detailed possibilities to evaluate the use and integration of ABAP based sources.

2 MOTIVATION

SAP Data Warehouse Cloud offers a rich set of features to integrate data into a cloud data warehouse. It often also offers integration possibilities for systems for release levels well prior to any integration technology built for the cloud.

Especially for ABAP systems, there are some restrictions when accessing data sources that have been provided years ago in a different context. In several early adopter care projects, situations came up, where a deeper understanding of HANA to ABAP connections was necessary and a deeper knowledge could already help in the planning process of using ABAP based connections.

3 GENERAL ASPECTS

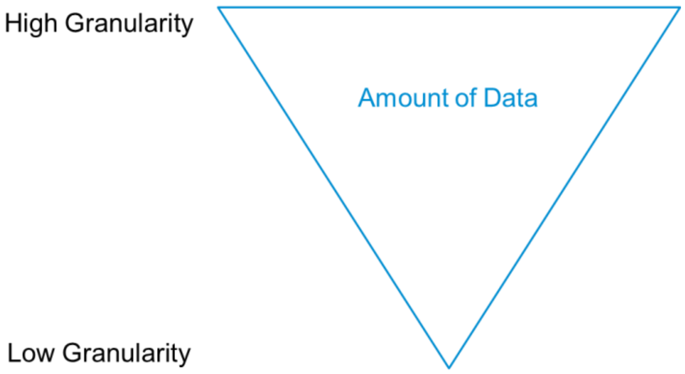
3.1 Applicable Software Versions

SAP Data Warehouse Cloud Version: 2021.24
DP Agent: 2.5.5.

3.2 Definitions

3.2.1 Granularity / Cardinality of Source Data

To avoid misconceptions a high granularity in this document refers to a low level of detail. A low granularity therefore refers to a high level of detail. For example, data can be summed up to show revenue in a year, on low granularity, but the data can also be displayed on a daily level which would be of higher granularity.



The granularity of data always has an effect on the amount of data being transferred. Since we are transferring data to the cloud, data gets uploaded via the corporate intranet bandwidth passing through an agent and the internet, which could all be possible bottlenecks.

The amount of data is mainly the number of lines, times the number of columns, times the memory consumption of the data type of the specific column. Therefore, selecting and transferring all columns, all lines at the same time can lead up to downloading gigabytes of data.

3.2.2 Join Pushdown

Join pushdown refers to whether a defined join on two entities in Data Warehouse Cloud from the same source, can be performed by the specific source instead of Data Warehouse Cloud.
This influences if data needs to be extracted prior joining or can be reduced by a join if performed in the remote source. The rule of thumb is that join pushdown is supported by database to database connections. ABAP connections do not push down joins.

3.2.3 Filter Push-Down

To limit the amount of data transferred by a network, filters should be applied to the remote source. This can be hindered in several ways. Either the remote source does not support a filter on the specific column or an operation in the adapter of the source prevents a correct application.

3.2.4 Change Data Capture (CDC)

Is a general reference to trigger based change recording. This can have multiple flavors, but normally requires the definition of fields the recording should be based on and, in more complex models, the origin of these fields.

3.2.5 Blue/Green Deployment of Models

Refers to a similar practice for data models as for servers. In this context a result set stays stable until a new result set is created. During a loading process changes to the target will only apply fully after the process is done. This should ensure consistency of data and prevent only partially updated sets of data.

3.3 Prerequisites

In this article we generally assume that a connection has been created and the relevant connectivity issues have already been solved. Mainly there are the following articles that can help during the setup process.

SAP Data Warehouse Cloud Connectivity Prerequisites:
https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/f1a39d1a763e48c8872f45c110a5a4e2.html

SDI Agent connectivity:
https://github.com/SAP-samples/data-warehouse-cloud-content-beta/blob/master/SAP%20Data%20Warehouse%20Cloud%20-%20DP%20Agent%20Installation%20V2.pdf

Agent Troubleshooting:
https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/8f6185069a51404ebf23c684fee8cf39.html

3.4 Authorizations in source system

To be able to access data in the source system the technical user from the connection needs to have specific authorizations to access the queue, write entries and execute specific function modules necessary for data transfer. The minimum requirements for a technical user is documented in the prerequisites SAP HANA Smart Data Integration and SAP HANA Smart Data Quality Installation and Configuration Guideline. The list of profiles and their target is listed here: https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SPS05/en-US/bcc0ff2acd6a4476b2912ff4cd71cd91.html

To access BW entities the user should additionally have the relevant analysis authorization to view the data requested.

This leads to several concepts, that depend on the way the system is set up. As stated in https://www.sap.com/documents/2021/09/8a0fc7ca-f67d-0010-bca6-c68f7e60039b.html "Organization of Spaces" there are several approaches to manage an SAP DWC system. When entitling the connection user with specific authorizations there can be two approaches:

1. The user in the source system has full blown authorization, and limiting data access needs to be controlled in DWC via DACs
2. Since connections are space dependent, each space connection can use a specific user with restricted access.

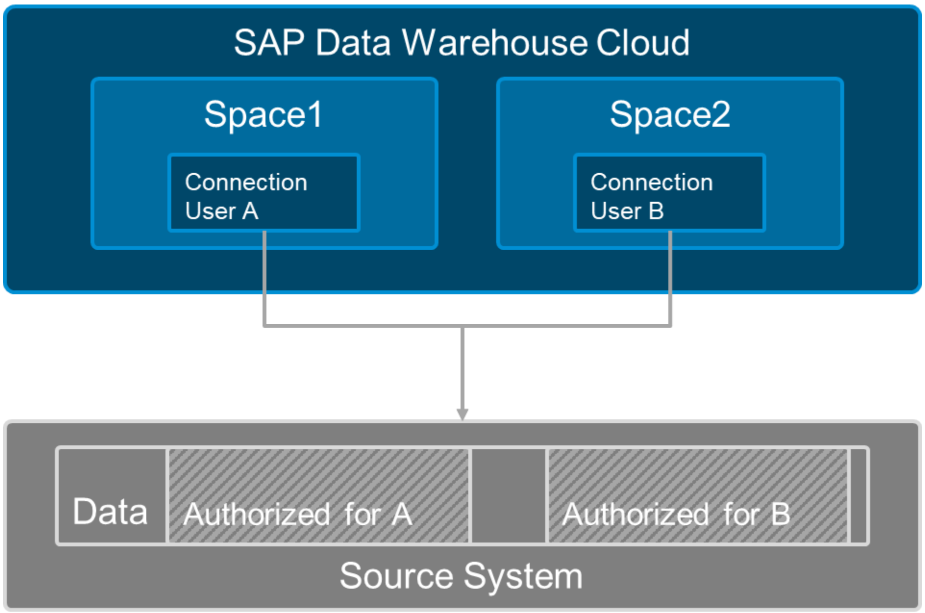


Figure 1: Restricted connection users

3.5 General Virtualization Considerations

3.5.1 Resource Consumption

One general misconception on virtualization, is that it saves hardware resources every time. This is not the case and can be demonstrated with a very simple example.
The target system (Data Warehouse Cloud) is generally not aware of all data changes in the virtual sources, therefore caching mechanisms do not apply (risk of resulting into invalidated real-time data). This means that each user executing a query will retrieve their own data set, since they cannot profit from shared cache entries. This amounts to the following worst-case scenario, where caching does not occur in either source or target (see Figure 2).

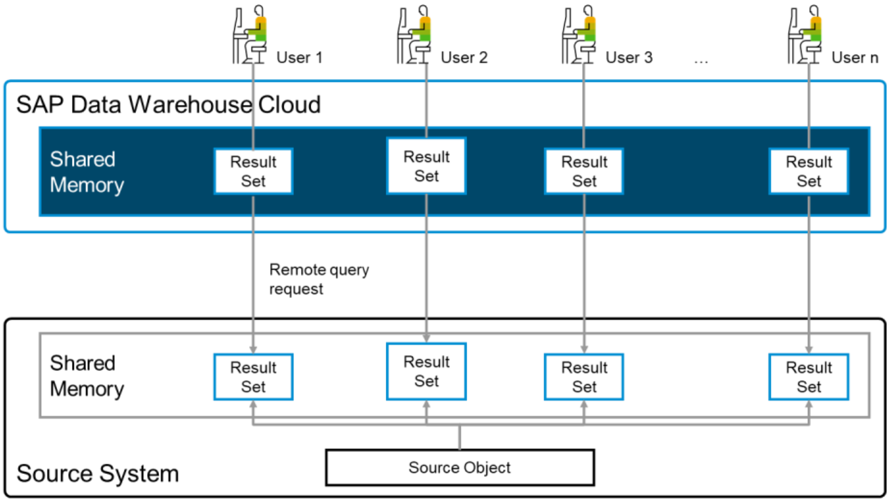


Figure 2: Query execution on virtual sources

These result sets are held as long as the session is held. Navigation in the reports makes this effect worse. In case of a replicated table (see Figure 3), users could profit from shared caches:

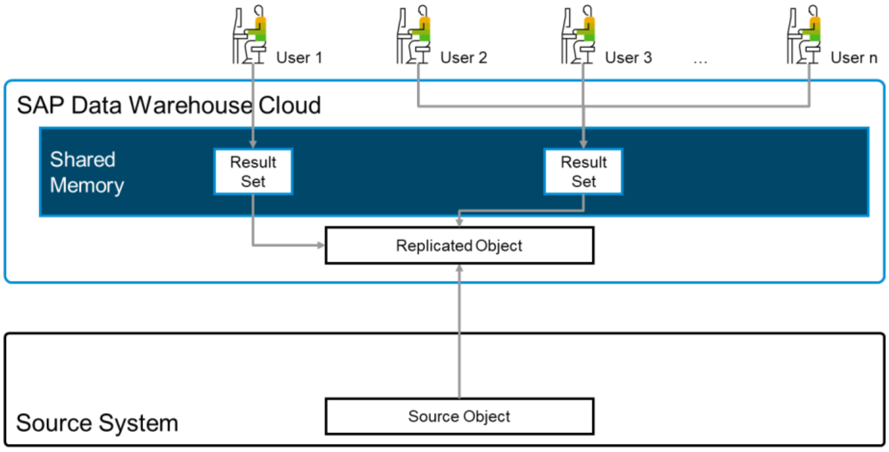


Figure 3: Query execution with replicated data

This is only the view on eventual memory consumption. A similar resource consumption can be seen for CPU consumption, used work processes or threads.

3.5.2 Agent Workload

The SDI Data Provisioning Agent is a translator but also a transmitter, this means data will pass through the agent. The minimum hardware requirement can be found here:
2559981 - System requirements for installing Data Provisioning Agent - SAP HANA Smart Data Integration

The configuration and administration guide can be found here:
https://help.sap.com/viewer/product/HANA_SMART_DATA_INTEGRATION/2.0_SPS05/en-US?expandAll=true

It is necessary to keep in mind, that these documents refer to general administration guidelines for on-premises deployment. In Data Warehouse Cloud not all options will be available as only a restricted set of parameters is exposed in HANA Cloud (underlying Data Warehouse Cloud). For any parameter change required exceptionally on a non-exposed parameter (e.g. issue sent to SAP in a customer case), SAP will take care based on internal service requests.

Memory, the number of connections used for transmission and CPU are the factors to consider when transferring data virtually or via batch load to Data Warehouse Cloud. In case of very high load, it makes sense to separate the load to several agents.

4 ABAP DATA INTEGRATION SCENARIOS

4.1 Overview

There are several connection types available for accessing ABAP systems - namely, SAP ABAP, SAP ECC, SAP S/4HANA, SAP BW. Each connection has the possibility to connect via SDI Agent for virtual access, table replication and view persistency, or via Cloud Connector for consumption of sources via Dataflow.

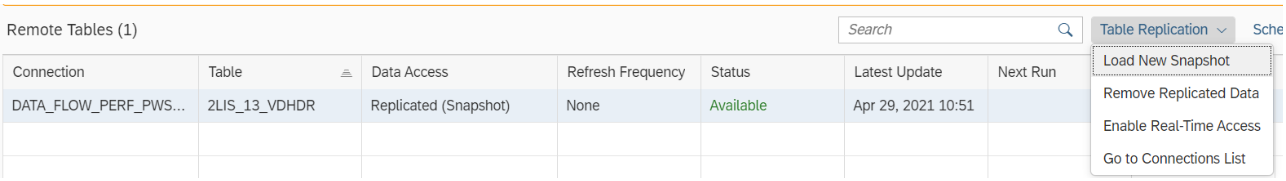
Please note that there is an additional connection type for connecting to SAP BW/4HANA (analytical) models. For more details refer to the relevant section of this document. For replicating data (with change data capture (delta) from underlying Data Warehouse models (InfoProviders) in SAP BW/4HANA the SAP ABAP or SAP BW connection will apply.

The table on the SAP Help Portal gives a general overview of all connection types in SAP Data Warehouse Cloud, as well as the specific connection types for ABAP systems and what they can consume: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/eb85e157ab654152bd68a8714036e463.html

4.2 Remote Table Replication

Remote Tables in SAP Data Warehouse Cloud are meant to integrate source data 1:1 without any ETL operations such as filtering, projections or any other technical or business logic. This is done in Data Flows. Remote Tables are used to:

- Access source data virtually using a generated virtual table, based on the source structure. The virtual table itself cannot be edited. If changes are necessary, they need to be performed in the source.
- To replicate data to SAP Data Warehouse Cloud, an additional replica table is used internally next to the virtual table. Using table replication, the options “Load New Snapshot” is available to load a full extraction, “Enable Real-Time Access” will trigger delta processing and CDC deltas (see Figure 4). If delta processing via CDC is not available, it will show as an information pop-up. “Enable Real-Time Access” will trigger an initialization with subsequent near real-time delta replication. The latter will perform an UPSERT replication of source data.



Connection	Table	Data Access	Refresh Frequency	Status	Latest Update	Next Run
DATA_FLOW_PERF_PWS...	2LIS_13_VDHDR	Replicated (Snapshot)	None	Available	Apr 29, 2021 10:51	

Figure 4: Table replication

To connect Remote Tables to ABAP based systems, currently SAP HANA Smart Data Integration and the Data Provisioning Agent are used. Please find more information here: https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SPS05/en-US/6634db61aa43435a977b608878e5313f.html

For a general overview on “Replication Data and Monitoring Remote Tables” in general please consult these SAP Help Portal pages: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/4dd95d7bff1f48b399c8b55dbdd34b9e.html

For an overview of delta enabled ODP Providers, it is necessary to differentiate by contexts S-API (Extractors), CDS (ABAP CDS Views) or BW (InfoProviders). Please go to the relevant paragraphs for further information.

4.3 View Persistency

View Persistency enables complex operations on sources and snapshots a model into a persistent form. Currently, view persistency enables full execution of a model, can be scheduled and also offers blue/green deployment. In context of view persistency blue/green deployment means during the scheduled load of the next snapshot the prior is available until the load is successfully terminated.

View persistency allows application creators to apply a stable version of a model for reporting. Especially for complex SQL statements or even SQLScript Views, an additional persistency can make sense.

In case of using view persistency on remote tables, it is worth mentioning, that an SQL statement from a model is optimized prior to being applied to remote sources. Therefore, the designed SQL statement can differ from the SQL statement applied to the remote source. This query can be tracked in the remote query monitor.

Persisted views can also be partitioned on columns from the dataset and breaks down the data into smaller and more manageable parts to enable persistency on views with large data volume. More information in the administration guide: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/9b1b595054c34da5b67ac07595093f82.html

4.4 Data Flow

Data Flows are created to import data from a source into a local target table. Sources can be tables and views from the space, or tables / files from remote connections. Once the source operator is added a series of data transformations can be applied by adding the operators join, union, projection, aggregation, and script. The script operator can transform the data coming from the input table using python code and return structured data. Supported and unsupported objects and functions in the script operator are documented in the modeling guide: https://help.sap.com/viewer/c8a54ee704e94e15926551293243fd1d/cloud/en-US/73e8ba1a69cd4eeba722b458a253779d.html.

After applying transformations, a table from the space can be chosen as the target operator. In case the target table does not exist, a target table can be created via the Data Flow editor.
The target operator offers different modes with which to write to the target table or modify the data in the target table:

- **Append**
  Write the data obtained from the data flow as new records appended to the end of the target table. This is useful if new records are delivered.
- **Truncate**
  Erase the existing data in the target table and replace it with the data obtained from the data flow. This mode is useful when a new full data set is provided.
- **Delete**
  Delete records in the target table based on the output of the data flow and based on the columns mapped in the target. All columns mapped in the target table is considered for the match condition and if the value of these columns match the record in the target table, it will be deleted.

In addition to the append mode, the checkbox ***Update Records By Primary Key*** can be used to instruct the data flow to update existing target table records that can be identified by their primary keys – where appropriate. This means the checkbox allows the append mode to work in an upsert mode, where new rows are inserted and the values of existing rows are being updated.

⚠ If the option ***Update Records By Primary Key*** is not selected, then all source records (including those that are already present in the target table) are appended, which may cause duplicate key errors. In case no primary key is defined in the target table, then all records will be appended as the checkbox cannot be selected.

4.5 Data Type Mapping

Numeric Types

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
INT1	3	0	1-byte integer, 0 to 255	b	SMALLINT	abap.int1	Integer
INT4	10	0	4-byte integer, -2,147,483,648 to +2,147,483,647	i	INTEGER	abap.int4	Integer
INT8	19	0	8-byte integer, -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	int8	BIGINT	abap.int8	Integer64
DEC	1-31	0	Packed number BCD format	p, length m DIV 2 + 1	DECIMAL	abap.dec	Decimal
DF16_DEC	1-15	0	Decimal floating point number stored in BCD format	decfloat16	VARBINARY length 8	abap.decfloat16	Decimal
DF34_DEC	1-31	0	Decimal floating point number stored in BCD format	decfloat34	VARBINARY length 16	abap.decfloat34	Decimal
FLTP	16	0	Floating point number	f	DOUBLE	abap.fltp	Double

Character-Like Types

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
CHAR	1-30000, maximum of 1333 for table fields	m blanks	Character string	c, length m	NVARCHAR	abap.char	String
SSTRING	1-1333	Empty string	Character string	string	STRING	abap.sstring	String
STRING	256-...	Empty string	Character string (CLOB)	string	NCLOB	abap.string	LargeString

Byte-like types

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
RAWSTRING	256-...	Empty string	Byte string (BLOB)	xstring	BLOB	abap.rawstring	LargeBinary

Date Types/Time Types

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
DATS	8	00000000	Date in the format YYYYMMDD	d	NVARCHAR 8	abap.dats	String(8)
DATN	8		Date in the internal format of database		DATE	abap.datn	Date
TIMS	6	000000	Time in the format HHMMSS	t	NVARCHAR 6	abap.tims	String(6)
TIMN	6		Time in the internal format of database		TIME	abap.timn	Time
UTCLONG			Time stamp (exact to 100 ns)		TIMESTAMP	abap.utclong	Timestamp

Character-Like Types with Special Semantics

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
NUMC	1-255	m zeroes	Numeric text	n, length m	NVARCHAR	abap.numc	String
CLNT	3	000	Client	c, length 3	NVARCHAR 3	abap.clnt	String(3)
LANG	1	Blank	Language key	c, length 1	NVARCHAR 1	abap.lang	String(1)

Currency Fields and Quantity Fields

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
CURR	1-31	0	Currency field in BCD format	p, length m DIV 2 + 1	DECIMAL	abap.curr	Decimal
CUKY	5	5 blanks	Currency key for currency fields	c, length 5	NVARCHAR 5	abap.cuky	String(5)
QUAN	1-31	0	Quantity field in BCD format	p, length m DIV 2 + 1	DECIMAL	abap.quan	Decimal
UNIT	2-3	2 or 3 blanks	Unit key of a quantity field	c, length m	NVARCHAR 2-3	abap.unit	String

Geo

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type	ABAP Simple Type	CDS Type
GEOM_EWKB					ST_GEOMETRY	abap.geom_ewkb	hana.ST_GEOMETRY

Obsolete Types

Type	Valid Places m	Initial Value	Meaning	ABAP Type	HANA Type
DF16_SCL	16	0	Decimal floating point number stored in binary format with scaling specified (obsolete)	decfloat16	VARBINARY length 8
DF34_SCL	34	0	Decimal floating point number stored in binary format with scaling specified (obsolete)	decfloat34	VARBINARY length 16
PREC	2	0	Obsolete data type	s	SMALLINT
VARC	1-...	None	Obsolete data type	c, length m	NVARCHAR

5 USING THE ABAP SDI ADAPTER FOR REMOTE TABLE FEDERATION AND REPLICATION

A general overview of supported features for the ABAP Adapter can be found in the SDI administration guide: https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SPS05/en-US/c911a7b06cf14c2999b78fe8ae2b96a8.html

If accessing from Data Warehouse Cloud to the ABAP layer via the ABAP adapter, the only SQL functionalities that are pushed down to the source are: SELECT, WHERE, TOP, or LIMIT. This excludes group by, sort, join and other functionalities that may be defined in a virtual model on top. Please note that missing push down capabilities can lead to performance degradation in virtual access as a higher amount of data is transferred to the Data Warehouse Cloud / HANA Cloud target system where the specific operation is then executed.

The following section will add specific considerations for accessing ODP Providers contexts such as S-API (Extractors), CDS (ABAP CDS Views) or BW (InfoProviders) using the HANA SDI ABAP Adapter.

5.1 Setting Up Source Tables

Navigating through source systems to find the right source can be tricky. DWC offers what it gets from its source. The metadata is broken down alphabetically to limit the display for performance reasons in visualization.

First let’s start with the source system types. Other than in database connections, where the source is possibly broken down to schemas or table groups, ABAP reads the application component tree. Depending on which type of extraction case is used, the display is different. As displayed before it is possible to consume data from ABAP tables or extractors, and extractors are categorized in contexts such as SAPI, CDS or BW. It is therefore good to know what sources are dealt with.

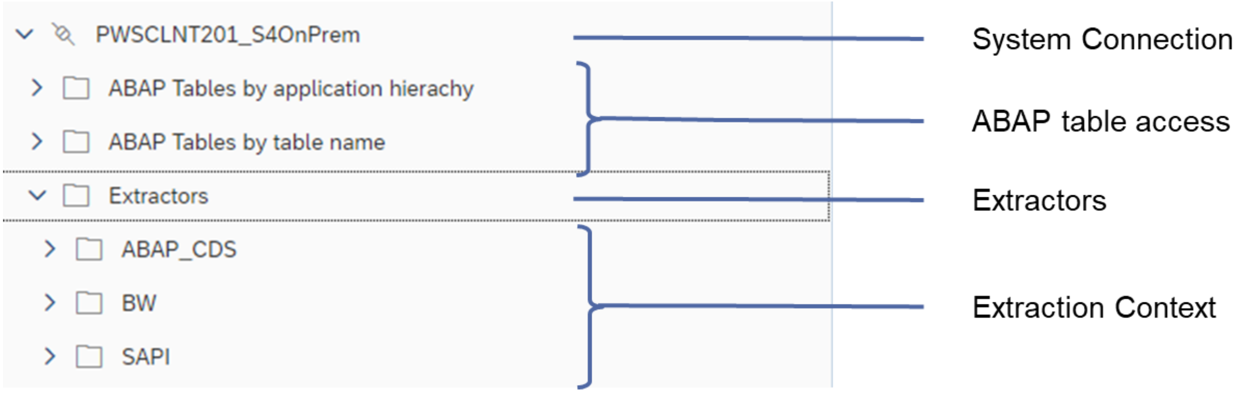


Figure 5: Setting up source tables

If the source is an S/4HANA or an ERP system, the main source objects are extractors such as SAPI and CDS. Tables can also be used with the limitations mentioned in 5.3 ABAP Table Reader. In case of a BW or BW/4HANA system, it is possible to access InfoProviders (DSOs, CompositeProviders, Queries – type InfoProvider, InfoObjects – type InfoProvider). As soon as you have determined the source object type, it is necessary to find the object itself. Since these are separated by the first letter, the following points need to be considered:

BW case: when using a customer namespace in the source or if accessing BPC embedded models, the first letter will be “/”, therefore this area needs to be looked at primarily. The classification is done by the technical name of the InfoProvider not the displayed text.

CDS Views: The referenced technical name of the CDS view is its SQL View name not its data definition. For example, the CDS view I_GLACCOUNTLINEITEMRAWDATA has the following header annotation:

```
@EndUserText.label: 'Raw Data of G/L Account Line Item'

@AbapCatalog: { sqlViewName: 'IFIGLACCTLIR',
                compiler.compareFilter: true,
                preserveKey: true,
                buffering.status: #NOT_ALLOWED }
```

The respective extractor to look for in DWC is therefore 'IFIGLACCTLIR' under the folder path: Extractors > ABAP_CDS > I…

SAPI Extractors: the most common extractors start with “0” (LoB extractors) or “2” (Logistics Cockpit extractors), therefore all relevant extraction scenarios are grouped here.

For Data Flows only ODP extraction contexts will be displayed.

5.2 ODP

For more information on Operational Data Provisioning see the FAQ document on https://blogs.sap.com/2017/07/20/operational-data-provisioning-odp-faq/

5.2.1 S-API DataSources / Extractors

S-API extractors are mostly regarded as acquisition of data for legacy ERP data. The extractors are mostly based on old ECC tables, that have been replaced by compatibility views in S/4HANA. Acquisition of data for Data Warehouse Cloud therefore already bears some complexity.

5.2.1.1 Virtualization

Virtualization via S-API extractors is generally not recommended. Of course, in specific, very restrictive scenarios, these sources can also be used for federation, but features such as join relocation, filters, group by clause or other database specific features are not supported.

The background to this is, that most S-API extractors are based on function modules and these only account for a specific extraction method, only provide filtering on specific columns and will preserve the order of data.

To find out more on the extractor capabilities, help articles on each extractor give a detailed description of their capabilities. It is also possible to search for additional information in the source system itself.
Whether a S-API Datasource supports virtual access can be found in the table ROOSOURCE. If the field VIRTCUBE has either entries 1 (without preaggregation) or 2 (with preaggregation), virtual access is generally available.
For more information please see SAP note 326482 – SAP RemoteCube: No source system is provided: https://launchpad.support.sap.com/#/notes/326482

To find more information on the targeted extractor all settings can be viewed in RSA2 transaction of the source system. Field attributes, extraction method and other important information can be found there. If a general, list-based analysis of the extractor fields is needed, the table ROOSFIELD will give all information on the specific fields.

Projections will generally apply to the source, the fields that were not requested will be left empty in the source queue and will not be transferred. A defined projection does not imply an aggregation and therefore not enforce a group by criteria, the data will still be sent in the source’s original granularity.

5.2.1.2 Extraction / Data Replication

For data integration to Data Warehouse Cloud, many S-API DataSources are not a suitable source (especially when real-time replication (= CDC / Delta) is used). First and foremost, data replication of deltas in Data Warehouse Cloud relies on upserting data. Necessary attribute of the source for extraction is a primary key. Most S-API DataSources have no primary key definition since they have a different extraction structure from their source tables.

To find the correct key definition and a well-founded estimation of the consequences of a modification of the extractor key, the best possibility is to open an incident on the respective component of the extractor (to be found in ABAP package information => attributes => application component). Either a well-founded answer or a correction can be given for the functional area.

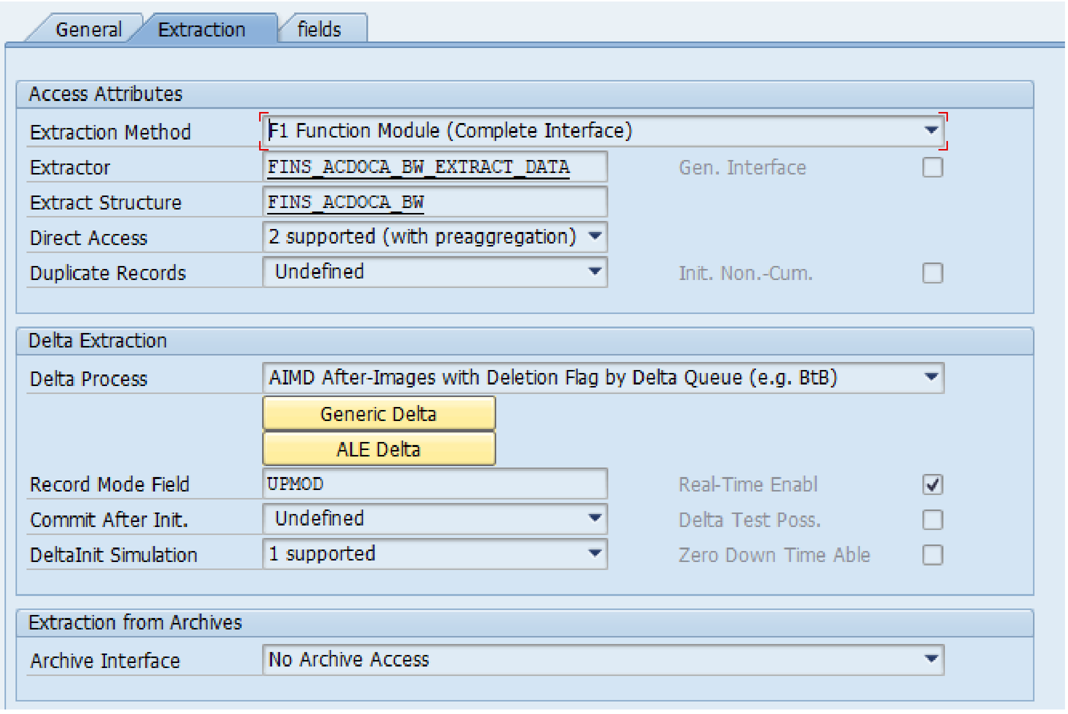
Please note that an uncoordinated modification of keys of standard extractors is not recommended and can have the following risk particularly when using real-time (delta) replication (which is typically the case for large data volumes):

- Real-Time Initialization will abort in case duplicates are sent based on the (custom modified) key definition
- Data Inconsistency will happen in real-time / delta in case UPDATEs on documents are turned into INSERTs because of a modified key
- Already initialized extraction processes to other sources may be affected by the change leading to inconsistencies.
- Other targets (consumers) loading in delta from that S-API DataSource may be impacted

For the highly demanded extractors 0FI_ACDOCA_10 and 0FI_ACDOCA_20 it has been confirmed that a modification to the key fields RLDNR, RBUKRS, PERIV, FISCYEARPER, BELNR, DOCLN is possible.
See also SAP note 2341038 – Usage of datasources 0FI_ACDOCA_10 and 0FI_ACDOCA_20 in sFIN / S4HANA: https://launchpad.support.sap.com/#/notes/2341038.
For G/L line items in higher S/4 releases please refer to the suitable CDS views for extraction.

Additionally, to the delta enablement of the extractor, the extractor must also comply in its delta method with the UPSERT method used in Data Warehouse Cloud to process CDC deltas. This is not the case for delta method ADD and ADDD. It is possible to check for extractors with this logic in ROOSOURCE table or in transaction RSA2.

To find key settings for the extractors (see Figure 6), it is either possible to look them up in RSA2 on the fields tab in the column “Key Field” or in ROOSFIELDS. In these cases, it will be necessary to define the key, if not already defined properly. The key definition depends on the attributes of an extractor such as extraction method, delta process, record mode field and the involved source tables. The determination of the key fields can be very complicated, especially looking at extractors based on function modules.



Access Attributes

Extraction Method	F1 Function Module (Complete Interface)		
Extractor	FINS_ACDOCA_BW_EXTRACT_DATA	Gen. Interface	□
Extract Structure	FINS_ACDOCA_BW		
Direct Access	2 supported (with preaggregation)		
Duplicate Records	Undefined	Init. Non.-Cum.	□

Delta Extraction

Delta Process	AIMD After-Images with Deletion Flag by Delta Queue (e.g. BtB)		
Record Mode Field	UPMOD	Real-Time Enabl	☑
Commit After Init.	Undefined	Delta Test Poss.	□
DeltaInit Simulation	1 supported	Zero Down Time Able	□

Extraction from Archives

Archive Interface	No Archive Access

Figure 6: Extraction Settings Overview

⚠ Please note that certain S-API DataSource / Extractors (e.g. FI-AA Business Content) may send duplicate records during snapshot replication or real-time initialization. As the Remote Table replication runs with an optimized INSERT processing on DataSource key level this will lead to “unique constraint violation” errors. For these cases, we recommend to use instead Data Flows with the Remote Tables as source and the target table set to Mode "APPEND" with the flag "Update Records By Primary Key (UPSERT)” enabled.

5.2.2 ABAP CDS Views

5.2.2.1 General Consideration

Only ABAP CDS views from source systems are visible to the ABAP Adapter that are of type “Extraction” and therefore must include the annotation: `@Analytics.dataExtraction.enabled: true`. Moreover, a data category must be maintained.
See also SAP Help documentation:
https://help.sap.com/viewer/cc0c305d2fab47bd808adcad3ca7ee9d/7.5.9/en-US/c2dd92fb83784c4a87e16e66abeeacbd.html

Please note ABAP CDS Views are the preferred method of integrating S/4HANA (On-Premise / Cloud) with SAP Data Warehouse Cloud. For additional information see:

- SAP note 2991278 - Extracting Delta Data from CDS Views in SAP S/4HANA using Change Data Capture Annotation: https://launchpad.support.sap.com/#/notes/2991278
- SAP note 2674373 - SAP S/4HANA Cloud: Extract data from SAP S/4HANA Cloud to SAP Business Warehouse and SAP BW/4HANA using CDS views (also applicable to SAP Data Warehouse Cloud): https://launchpad.support.sap.com/#/notes/2674373
- https://blogs.sap.com/2021/04/19/part-1-sap-s-4hana-cloud-data-integration-to-power-your-intelligence-enterprise/
- https://blogs.sap.com/2019/12/13/cds-based-data-extraction-part-i-overview/

This implies that the CDS views mostly consist of raw, high granularity data and there is a necessity to preserve the sequence of data by creation/update. This also implies high analytical functions via virtual access are not the target.

Not released CDS-Views and missing data extraction setting
The recommendation is to open a ticket on the relevant component to get this resolved, but this process takes some time. In the meantime, since you are on SAP S/4HANA you can build a custom CDS View 1:1 on top and enable it for extraction on your own.

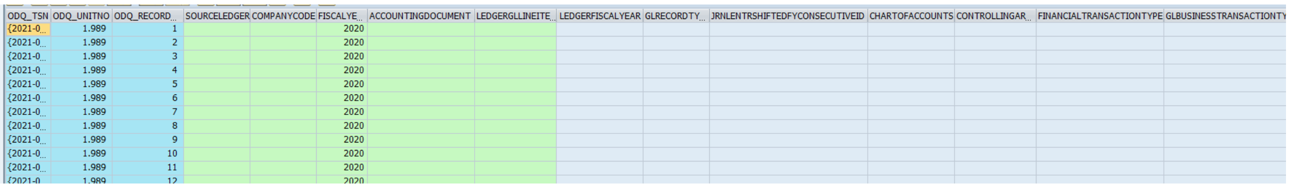
Here are few blogs/notes for more information on the topic:

- Definition of Stability Contracts
- How to deal with “Not released” predefined VDM in S/4HANA
- CDS based data extraction – Part I Overview
- CDS based data extraction – Part II Delta Handling
- SAP Note: 2540701 - Custom CDS Views – Necessary Adoptions after Upgrade

5.2.2.2 Virtualization

Due to the extraction context of the source data, group by statements are either not pushed down by Data Warehouse Cloud or not applied by the extractor. This means, that data will be sent in full granularity from the source S/4 system to Data Warehouse Cloud regardless of the aggregation defined. The projection will hide other columns, but entries will be repeated and not selected distinctly.

Example CDS Extractor IFIGLACCTLIR (DDL view name I_GLACCOUNTLINEITEMRAWDATA line-item extractor on ACDOCA) with projection and aggregation on year and amount in global currency.



ODQ_TSN	ODQ_UNITNO	ODQ_RECORD…	SOURCELEDGER	COMPANYCODE	FISCALYE…	ACCOUNTINGDOCUMENT	LEDGERGLLINEITE…	LEDGERFISCALYEAR	GLRECORDTY…	JRNLENTRSHIFTEDFYCONSECUTIVEID	CHARTOFACCOUNTS	CONTROLLINGAR…	FINANCIALTRANSACTIONTYPE	GLBUSINESSTRANSACTIONTY
{2021-0…	1.989	1			2020									
{2021-0…	1.989	2			2020									
{2021-0…	1.989	3			2020									
{2021-0…	1.989	4			2020									
{2021-0…	1.989	5			2020									
{2021-0…	1.989	6			2020									
{2021-0…	1.989	7			2020									
{2021-0…	1.989	8			2020									
{2021-0…	1.989	9			2020									
{2021-0…	1.989	10			2020									
{2021-0…	1.989	11			2020									
{2021-0	1.989	12			2020									

Figure 7: Year is repeated for each entry.

5.2.2.3 *Extraction / Data Replication*

ABAP CDS “extractors” are more suitable than S-API extraction method for Data Warehouse Cloud since they prepare data for HANA consumption. This makes applying delta easy for SDI change types and does not amount into complications, such as S-API extractors.

As a side note, S/4HANA systems no longer offer some S-API data sources.
For more information see the following SAP note 2500202 - S4TWL - BW Extractors in SAP S/4HANA: https://launchpad.support.sap.com/#/notes/2500202

Therefore, a general best practice is to use ABAP CDS views as far as possible and to revert to S-API DataSources only in case of a gap. You can also raise gaps in ABAP CDS “extractors” compared to S-API DataSources via the SAP Customer Influence Program (https://influence.sap.com/public/ci/help/UserDocumentation_CustomerInfluenceforSAPS4HANACloud.pdf) or by creating a customer case on the respective support component (see respective SAP notes) for the former S-API DataSource (e.g. BW-BCT-MM-PUR for S-API DataSource 2LIS_02_ITM).

SAP Data Warehouse Cloud will only offer real-time, thus delta, replication for CDS-views with change data capture feature enabled. In the newer S/4HANA versions the annotation is:
`@Analytics.dataExtraction.delta.changeDataCapture…`
If SAP Data Warehouse Cloud only offers snap shots on a CDS source, check the definition accordingly.

CDC deltas from ABAP CDS Views always require key definitions and therefore should not pose problems to Data Warehouse Cloud. There are several ways in ABAP CDS to define the required fields for CDC deltas. For more information these can be viewed here: https://help.sap.com/viewer/cc0c305d2fab47bd808adcad3ca7ee9d/202009.000/en-US/c2dd92fb83784c4a87e16e66abeeacbd.html?q=cdc%20delta.

5.2.3 SAP BW and SAP BW/4HANA

SAP BW and SAP BW/4HANA offer a variety of object types to use. In general, all BW objects of type InfoProvider are available to Data Warehouse Cloud.

5.2.3.1 *General Considerations*

There are two cases in the availability of BW objects as InfoProvider.

Default Availability:

- CompositeProvider
- ADSO

Availability by setting:

- Query (in “Runtime Properties” of the Modelling Tools editor)
- InfoObject (in “General” tab of the Modelling Tools editor)

5.2.3.2 *Virtualization*

For virtual access it makes sense to use the analytic runtime objects. These objects are the CompositeProvider and the BW-Query. Still the interface ODP is not necessarily meant for massive virtual reporting. The BW analytic objects do allow ameliorations in comparison to the normal extraction context.

For Queries: Query cache is used. The repeated execution of a query with the same result set or subset results, will be read from query cache and therefore have a better runtime as well as resource consumption in the source.

Queries and CompositeProviders: Both BW analytic objects, queries and CompositeProviders, aggregate data before sending to the ODP queue. Therefore, selecting specific columns will imply a group by on the result set of BW.

Please note that Navigational Attributes are not available in ODP based data extraction.

For connecting to these object types in SAP BW/4HANA we strongly recommended to use the specific connection type "SAP BW/4HANA Model Transfer”. For more details refer to the relevant section of this document.

5.2.3.3 Extraction

For extraction purposes the case breaks down to the normal BW use cases. Any InfoProvider is available for extraction, with the context of the object.

QueryProvider, CompositeProvider: only full extraction mode.

For DataStore Object CDC / real-time replication in SAP Data Warehouse Cloud is always available for the type “Standard” with Change Log. Other types may not meet the requirements for UPSERT replication (see comments above). In any case the Remote Table Monitor in SAP Data Warehouse Cloud will inform you accordingly when “Initialize Real-Time" is selected for source BW InfoProviders.

5.3 ABAP Table Reader

5.3.1 General Considerations

The ABAP Table Reader can be compared to the table display functionality of the ABAP layer. It is mainly considered as a generic display or consumption option for tables via the ABAP stack. This means the user roles are checked during access, there is no analytic privilege applied and data can be read on an ad-hoc basis.

There is no aggregation defined in the table reader, a projection will hide columns, but data will not be aggregated before sending. This makes projection but also filter conditions yet more relevant.

Filtering can be compared to table display in the ABAP layer – complex selection criteria are not supported. Generally, all selections are considered with an ‘OR’ operator.

Please note that reading the data from ABAP Tables using the ABAP (DDIC) Layer only allows for snapshot (full) replication as there is no CDC information available (compared to RDBMS log reader or trigger adapters or compared to using ODP).

5.3.2 Streaming Read

The table reader uses a dialogue process in the ABAP layer for reading data. It collects and sends it at the end of collection. This may result in memory issues on the ABAP backend. One dialogue process has a specific extended memory and heap memory quota. These parameters are defined by the basis administrator.

To limit the amount of resources used for larger requests, streaming read is necessary. This functionality will package read data and send package by package. It is recommended to turn this on by default, if the table reader is intended to be used.

The callback destination for the table reader is the gateway of the source ABAP system. This means, the callback destination host is the same server as the source ABAP system server, the port is the configured gateway port from the ICM framework.

For more information see: https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SPS05/en-US/960a1e7197a94da7ae379caf305bdbc7.html

5.4 SAP BW/4HANA Model Transfer

For an overview of the BW/4HANA model transfer, please refer to the SAP note 2932647 - Supported and unsupported features with SAP BW/4HANA Model Transfer in SAP Data Warehouse Cloud (https://launchpad.support.sap.com/#/notes/2932647) and to the SAP Help Portal: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/1caba954bc604e00bf8e82e383a46368.html

5.5 Table Partitioning

The replication of remote tables with larger data volumes can sometimes be a challenge without having out-of-memory errors or high memory peaks. To avoid such situations partitions can be created per remote table. Hence, enabling a partitioned data transfer of large datasets for supported connection types (all remote table-enabled connections that use a Data Provisioning agent, see administration guide: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/bffbd58c15784a62af0520f171018ded.html).

The partitioning works for both snapshot and real-time enabling replication. From the details screen of the remote table monitor, partitions can be created on columns of the dataset to break it down into smaller and more manageable parts. Therefore, splitting a single transaction into several partitions – each with its own transaction and the option for parallel execution. Up to 5 parallel processes can be defined. By default, the partitions are loaded one after the other to control resource consumption.

5.5.1 Partitioning Columns

When the column for the partitions is selected, it might happen that not all the columns available in your dataset. That may have different reasons:

- Not all data types are supported for partitioning depending on the data source
    - SAP HANA data sources, only the following data types are supported
        - Datetime types: DATE, TIME, TIMESTAMP, SECONDDATE
        - Numeric types: TINYINT, SMALLINT, INT, BIGINT, DECIMAL(p,s)
        - Character string: NVARCHAR
    - For ODP data sources, only the following columns can be selected:
        - Columns which support where clause of the SELECT statement.
        - Columns equal to or less than 45 characters.
- If a non-key column selected to define partitions, it must be ensured that the column value of a record in the source data is not changed during replication. Otherwise, data loading may fail because a record with the same key is replicated in two different partitions.
- Partitions cannot be defined on columns which contain amounts associated with a currency or quantities associated with a unit.

For more information on partitioning limits check the online documentation: https://help.sap.com/viewer/f9c5015e72e04fffa14d7d4f7267d897/2021_01_QRC/en-US/8dd866a688ec4914a074727a2c800142.html

⚠ For ODP data sources, partitions on non-key columns can only be created if the remote tables have been deployed using a DP agent connection from release 2.5.4.1 or newer. Moreover, if the repository entity comprises filter capabilities, the remote table in the Data Builder needs to be refreshed before you can create partitions.

In case you change the partitioning column after you have already loaded data into your partitions, then you are required to delete the existing partitioning before being able to change the partitioning column.

5.5.2 Number of Partitions

The number of partitions is defined by entering a range for each of them, where the ranges must not overlap. Partitioning requires an in-depth knowledge of the values that are used or are valid for the chosen partitioning column. Be aware, that there always will be an “others” partition for all values not fitting the partitioning criteria.

- Negative values are not valid for partition bounds of numeric columns and will be converted to 0 at runtime.
- The low bound must be lower than the high bound. The order is checked according to the type of the partition column. Example: for string-like columns, '90' <= DOC_ID < '100' is not valid, whereas '090' <= DOC_ID < '100' is valid.
- Values entered in the bounds for ABAP ODP remote tables must match the ABAP internal format: for example, enter "20210714" if the date is "July 14, 2021".

Examples for Partitioning

Let's take an example to understand how the partitioning is done. The following partitions for column Calyear, a 4-digit numeric field which can include NULL values, are defined in the UI:

1. Partition 1: 2000 <= Calyear < 2005
2. Partition 2: 2005 <= Calyear < 2010
3. Partition 3: 2010 <= Calyear < 2011

The replica table is partitioned accordingly. An "Others" partition is added for data, which is not covered by the defined partitions. Partitions for Replica Table:

1. Partition 1: 2000 <= Calyear < 2005
2. Partition 2: 2005 <= Calyear < 2010
3. Partition 3: 2010 = Calyear
4. Partition 4: Others

The partition ranges are also used by SDI data load runtime as filters in the SELECT statements sent to the remote source. Usually, the filters are exactly the same as the partition ranges, with the exception of ABAP ODP Extractors. In this case, the filters are as follows:

1. Partition 1: 0000 <= Calyear <= 1999
2. Partition 2: 2000 <= Calyear <= 2004
3. Partition 3: 2005 <= Calyear <= 2009
4. Partition 4: 2010 = Calyear
5. Partition 5: 2011 <= Calyear <= 9999
6. Partition 6: Calyear = NULL

www.sap.com/contactsap

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

THE BEST RUN SAP®