

Interface Document – SAP Business Warehouse Staging Scenario



Applies to:

SAP NetWeaver 7.3

Summary

This document describes the integration of metadata and data transfer from a third party tool with the SAP Business Warehouse. The integration is based on the “new” DataSource (object type R3TR RSDS) and re-uses the already known APIs which existed for the 3.x integration scenarios like Business Objects, etc.

Authors: Oliver Thomsen

Company: SAP AG

Created on: 30 May 2011



SAP
INTEGRATION AND CERTIFICATION CENTER

**APIs for Staging scenario
based on the new
DataSource in
SAP NetWeaver Business
Warehouse 7.3**

Version 1
November 30, 2011

Table of Contents

Flat structures	5
Hierarchies.....	6
Very simple customer hierarchy.....	7
Simple customer hierarchy	8
Customer hierarchy with intervals.....	8
Hierarchy with multiple parents for one node	9
Design Scenario	10
Extraction Scenario.....	11
Limitations	12
Data types and formats.....	13
Data Transfer in SAP internal format.....	14
Data Transfer in External format.....	14
Certification.....	15
Technical Prerequisite	15
Organizational Prerequisites.....	15
Metadata Objects for Staging	16
DataSource with flat structure.....	16
APIs	16
DataSource Hierarchy	20
APIs	20
Remote Data Provider (in 3 rd Party System)	22
APIs in detail	22
InfoPackage.....	24
Source System	24
ABAP Dictionary Structures.....	25
BAPIDSHEADER	25
BAPIDSHEADERATTR.....	25
BAPIDSHIER	25
BAPIDSFIELD.....	25
BAPIDSLIST	26
BAPIDSSEL	26
BAPIDSTYPESEL.....	26
BAPI6100DARAW.....	27
BAPI6100DA	27
BAPI6100SL.....	27
BAPI6101SL.....	27
BAPI6104	27
BAPI6107DP	28
Copyright.....	29

Introduction

With SAP NetWeaver 7.0 SAP NetWeaver Business Warehouse (SAP NetWeaver BW) introduced the (new) DataSource together with the Transformation and Data Transfer Process. As a consequence APIs are provided here which can be used to read and write metadata of the DataSource in “BAPI Source systems”.

Looking at the InfoPackage, source system and external data provider the new scenario makes use of the already existing APIs which are already known as “Staging BAPIs” [please contact ssstp@sap.com for the document]. Although the existing APIs can also be found as Business Objects in the Business Object Repository, in this document they are described on the API level, i.e. as RFC enabled function modules.

By using these APIs, customers and providers of 3rd Party tools can connect their metadata repositories and their extraction engines to SAP NetWeaver BW.

The APIs (or Business Objects) for updating and retrieving metadata of InfoObjects, DataStore objects (formerly known as ODS-Objects), InfoCubes and InfoObject Catalogs as well as the definition of InfoPackages are not described in this document. Please refer to the existing documentation.

The basic concept for the staging engine of SAP NetWeaver BW is the new *DataSource*¹. DataSources can be defined using flat structures for transaction data, master data attributes and master data texts. DataSources for hierarchies use a flat structure in order to transfer data which is split into different segments within SAP NetWeaver BW. Using the new DataSource APIs, instead of the existing BAPIs for the maintenance of InfoSources, in the 3rd party system, it is (on the technical API level) no longer required to distinguish between the data types master data attributes, master data texts and transaction data. However, the data type still exists as an attribute of the DataSource for semantically reasons.

The new DataSource includes the Transfer Structure and persistence called Persistent Staging Area (PSA). It is based on field descriptions and does not refer to InfoObjects – therefore APIs in order to maintain InfoObjects do not have to be used necessarily in conjunction with the new DataSource.

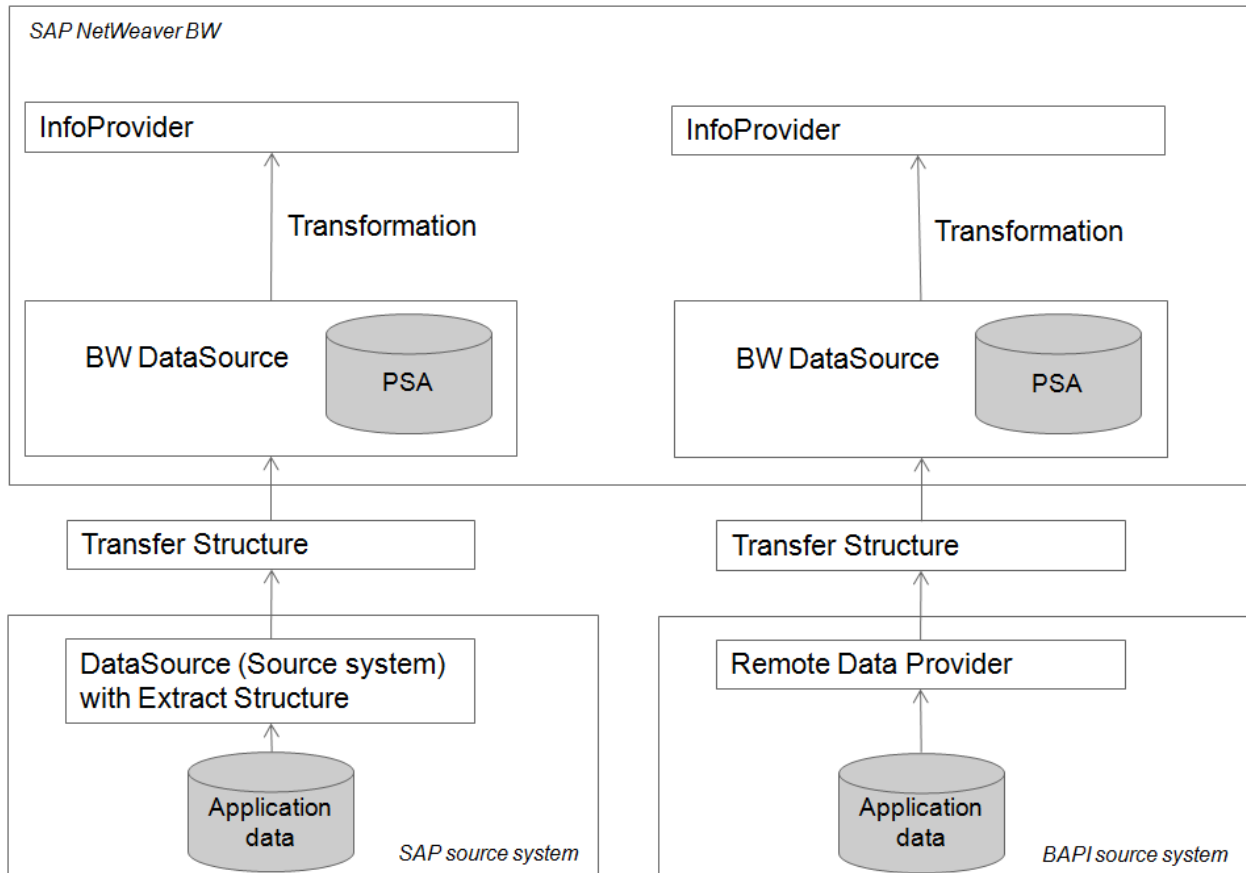
The following diagrams visualize the DataSource concept for the available types for flat structure or hierarchies.

¹ Object type RSDS. When using the term DataSource in this document, the DataSource with object type RSDS is meant. If we have to distinguish or explicitly talk about the “old” DataSource (object type ISFS), we use the term “3.x DataSource”

Flat structures

In case of a DataSource for flat structures the incoming data records are transferred identically into SAP NetWeaver BW and stored in the PSA. From here the data is transformed using the Transformations before it is written into the data target, e.g. to an InfoCube.

Please note: One DataSource could be used to update several data targets, one InfoCube can be updated by several DataSources (from different or identical source systems)

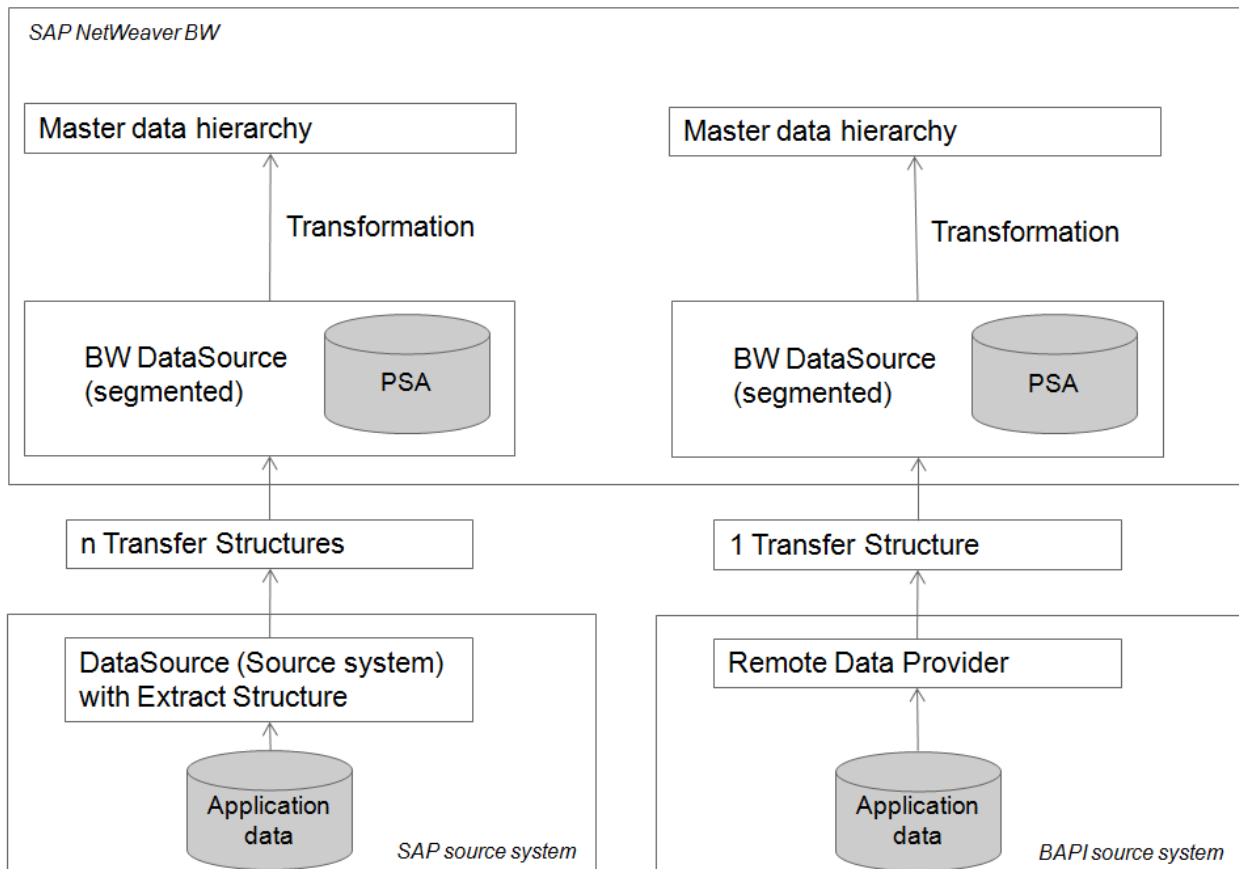


DataSource with flat structure

Hierarchies

SAP NetWeaver BW provides a way of handling hierarchies. The transfer is based on the so-called “IDOC structure” which has a pre-defined structure (i.e. not “flexible”).

Within the Data Warehousing Workbench (DWB) of SAP NetWeaver BW it is also possible to define “flexible” hierarchies meaning that additional compounded fields can be added within the InfoObject maintenance. Hierarchies with flexible structure cannot be maintained via API, but only in User Interface (DWB), as the flexible structure (with field names) is derived from the target InfoObject.



DataSource for master data hierarchies

In case of master data hierarchies, incoming data records are transferred into SAP NetWeaver BW and split into the different segments (header, header texts, nodes, node texts, intervals) and then transformed using the Transformations.

The transfer structure used for sending hierarchy data to SAP NetWeaver BW depends on the characteristics of the specific hierarchy. As a basis for the transfer structure the following template is used:

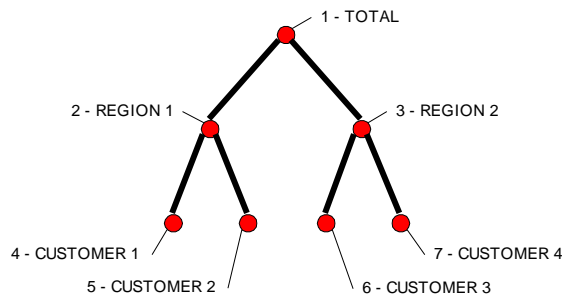
Field name	Description
NODEID	Unique id for the hierarchy node (local to each hierarchy)
INFOOBJECT	Name of the InfoObject this hierarchy node refers to
NODENAME	Name of the key value of this hierarchy node
LINK	Flag: ‚X‘ = link node
PARENTID	Node id of the parent node
CHILDID	Node id of the first child node

NEXTID	Node id of the next sibling
DATETO	Date valid to (only used for hierarchies with time dependent nodes)
DATEFROM	Date valid from (only used for hierarchies with time dependent nodes)
LEAFTO	Key value to (only used for hierarchies with interval nodes)
LEAFFROM	Key value from (only used for hierarchies with interval nodes)
LANGU	Language (only used if there's no text information available in the master data)
TXTSH	Short text (only used if there's no text information available in the master data)
TXTMD	Medium length text (only used if there's no text information available in the master data)
TXTLG	Long text (only used if there's no text information available in the master data)

The following examples are based on the non-flexible ("IDOC") Format.

Very simple customer hierarchy

This may best be illustrated by the sample hierarchy shown below:



This hierarchy will have to be transferred to SAP NetWeaver BW as shown in the following table (Initial value '00000000' for IDs left out for readability):

NODE ID	INFO OBJECT	NODE NAME	LINK	PARENT ID	CHILD ID	NEXT ID	LANGU	TXTSH	TXTMD	TXTLG
1	0HIER_NODE	TOTAL			2		EN	Total	Total	Total
2	0HIER_NODE	REGION 1		1	4	3		Region 1	Region 1	Region 1
3	0HIER_NODE	REGION 2		1	6			Region 2	Region 2	Region 2
4	0CUSTOMER	CUSTOMER 1		2		5				
5	0CUSTOMER	CUSTOMER 2		2						
6	0CUSTOMER	CUSTOMER 3		3		7				
7	0CUSTOMER	CUSTOMER 4		3						

Please note, that in this example we do not have a special InfoObject for the region. In order to be able to display the texts 'Region 1' and 'Region 2' in a report, you will have to send the two nodes as specified above.

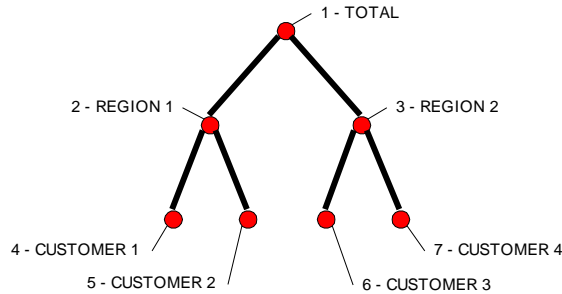
Because the node name is a key value, it is not possible to specify mixed case or lower case values for this field.

The DATETO, DATEFROM fields are not shown here but have to be specified, if the nodes of the hierarchy are defined in the InfoObject definition as time dependent.

The LEAFTO, LEAFFROM fields are described in an example below.

Simple customer hierarchy

This may best be illustrated by the sample hierarchy shown below:



This hierarchy will have to be transferred to SAP NetWeaver BW as shown in the following table (Initial value '00000000' for IDs left out for readability):

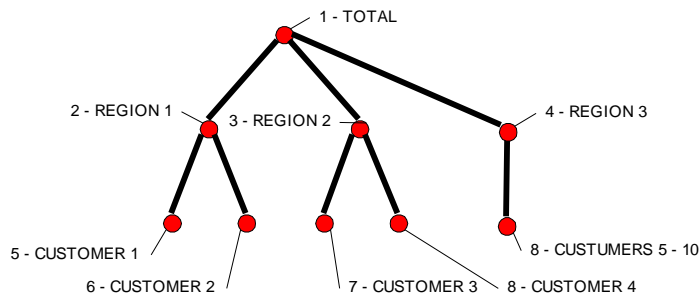
NODE ID	INFO OBJECT	NODE NAME	LINK	PARENT ID	CHILD ID	NEXT ID	LANGU	TXTSH	TXTMD	TXTLG
1	0HIER_NODE	TOTAL			2		EN	Total	Total	Total
2	0REGION	REGION 1		1	4	3				
3	0REGION	REGION 2		1	6					
4	0CUSTOMER	CUSTOMER 1		2		5				
5	0CUSTOMER	CUSTOMER 2		2						
6	0CUSTOMER	CUSTOMER 3		3		7				
7	0CUSTOMER	CUSTOMER 4		3						

The difference between this and the previous example is, that in this case the region has its own InfoObject called *0REGION*. Text information to the region in this case is read from the master data text table of the InfoObject *0REGION*.

The DATETO, DATEFROM fields are not shown here but have to be specified, if the nodes of the hierarchy are defined in the InfoObject definition as time dependent.

The LEAFTO, LEAFFROM fields are described in the next example.

Customer hierarchy with intervals



This hierarchy will have to be transferred to SAP NetWeaver BW as shown in the following table:

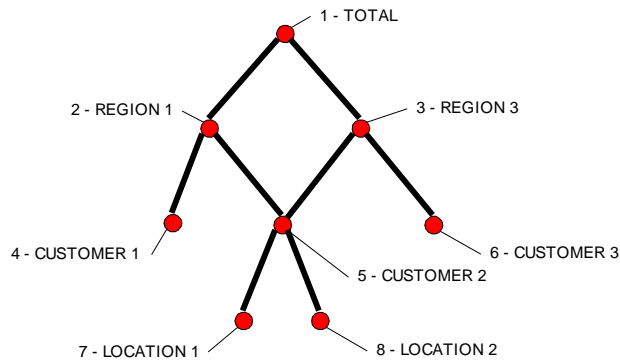
NODE ID	INFO OBJECT	NODE NAME	LINK	PARENT ID	CHILD ID	NEXT ID	LEAFFROM	LEAFTO
1	0HIER_NODE	TOTAL			2			
2	0REGION	REGION 1		1	5	3		
3	0REGION	REGION 2		1	7	4		
4	0REGION	REGION 3		1	9			

5	0CUSTOMER	CUSTOMER 1		2		6		
6	0CUSTOMER	CUSTOMER 2		2		7		
7	0CUSTOMER	CUSTOMER 3		3		8		
8	0CUSTOMER	CUSTOMER 4		3		9		
9	0CUSTOMER	INTERVAL1		4			Customer 5	Customer 10

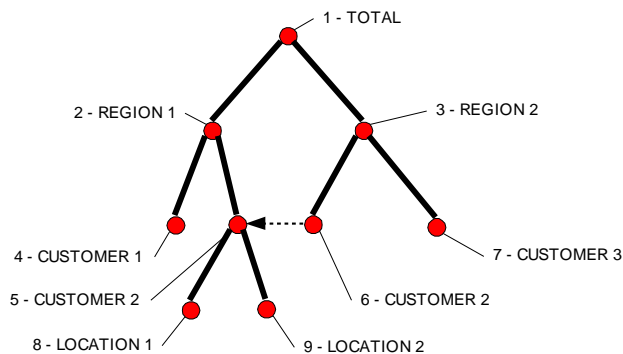
The text fields have been left out to reduce the size of the table.

Hierarchy with multiple parents for one node

SAP NetWeaver BW also supports hierarchies with loops as shown in the next example



To understand how SAP NetWeaver BW deals with this kind of hierarchies, have a look at the following alternative graphical representation of the hierarchy:

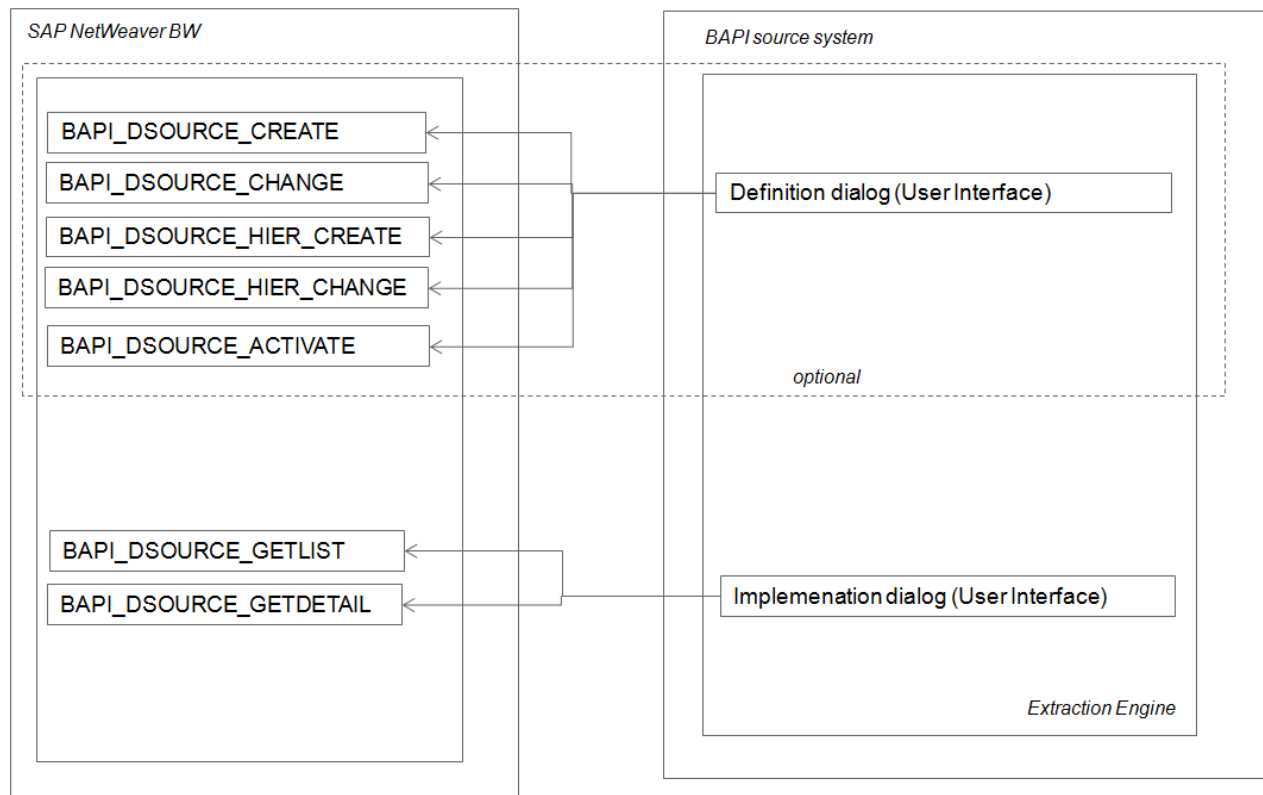


NODE ID	INFO OBJECT	NODE NAME	LINK	PARENT ID	CHILD ID	NEXT ID
1	0HIER_NODE	TOTAL			2	
2	0REGION	REGION 1		1	4	3
3	0REGION	REGION 2		1	6	
4	0CUSTOMER	CUSTOMER 1		2		5
5	0CUSTOMER	CUSTOMER 2		2	8	
6	0CUSTOMER	CUSTOMER 2	X	3	8	7
7	0CUSTOMER	CUSTOMER 3		3		
8	0LOCATION	LOCATION 1		5		9
9	0LOCATION	LOCATION 2		5		

Design Scenario

This chapter contains a basic scenario considered typical for the use of the Staging APIs. However, more complex (metadata update and upload) scenarios can easily be implemented based by means of these APIs.

The first thing to do in this basic scenario is to map the extracted data against the transfer structure of a DataSource defined in SAP NetWeaver BW. This scenario applies to DataSources with a flat structure as well as to DataSource for hierarchies:



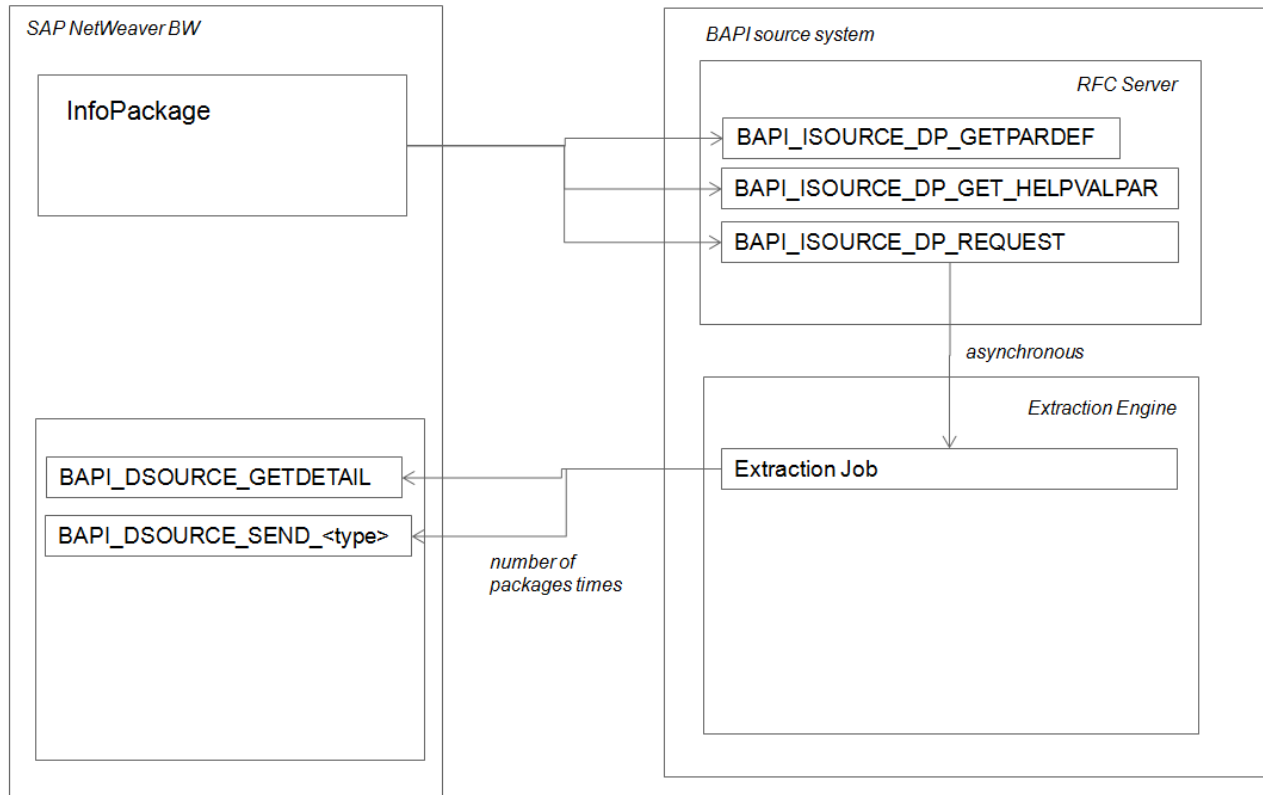
Design Scenario

1. The source system calls the API `BAPI_DSOURCE_GETLIST` of SAP NetWeaver BW. It may then present this list in a dialog box and have the user select a DataSource for mapping it with the structure of the extracted data.
2. The mapping process is based on the layout of the transfer structure which can be retrieved from SAP NetWeaver BW by calling the API `BAPI_DSOURCE_GETDETAIL` of the DataSource.

Extraction Scenario

The next step is to actually extract data from the data sources, perform the mapping defined in step one and then send the extracted data to SAP NetWeaver BW. This scenario – with minor modifications – applies to every type of DataSource supported by SAP NetWeaver BW.

A typical scenario to request data form a 3rd Party system is the following:



Extraction scenario

When defining a data request in SAP NetWeaver BW the InfoPackage requests a list of parameters needed to start the extraction process in the source system by calling the API **BAPI_ISOURCE_DP_GETPARDEF** of the source system. Examples for this kind of parameters are the *user name* and the *password* needed to log on to the source data base system.

If the user wants to see help values from the source system for the selection fields the InfoPackage calls the API **BAPI_ISOURCE_DP_GET_HELPVALPAR** and shows the results of this as a popup so that the user can choose one of the values as a selection criteria for the 3rd Party selections.

Then the InfoPackage sends a data request by calling the API **BAPI_ISOURCE_DP_REQUEST**. Parts of the request are the values of the requested parameters, selection criteria and request details such as a unique request ID, the name of the corresponding DataSource (note: on API level, the field name is still InfoSource here, as these APIs already existed before as Business Objects).

In order to ensure that the extract structure exactly matches the current transfer structure the 3rd party system has to call the API **BAPI_DSOURCE_GETDETAIL** for the DataSource data is requested for. This is necessary as this defines the current structure ("Transfer Structure") the data has to be supplied.

The extracted data is then sent to SAP NetWeaver BW by calling one of the following API's:

BAPI_DSOURCE_SEND_CHAR	If the sending system is implementing the old RFC library and is a Unicode system
BAPI_DSOURCE_SEND_CHARSTRING	If the sending system is implementing the new RFC library and is a Unicode system
BAPI_DSOURCE_SEND_RAW	If the sending system is implementing the old RFC library and is a non-Unicode system
BAPI_DSOURCE_SEND_RAWSTRING	If the sending system is implementing the new RFC library and is a non-Unicode system

The use of function BAPI_DSOURCE_SEND_CHARSTRING is preferred, since it requires the least transformations of the data.

In general, this API supports sending any number of records per call (each call defining a data packet). However, the API *BAPI_ISOURCE_DP_REQUEST* has transferred the desired PACKAGESIZE. This should normally be between 10.000 to 100.000 records.

Note: The loading scenario has been designed to work asynchronous. This means that after receiving (and checking) the request the 3rd Party system has to split up the extraction job in a new self-contained process and return the process of the request sending to SAP NetWeaver BW.

Limitations

External APIs for the definition of Transformations and Data Transfer Processes are currently not provided by SAP NetWeaver BW, however the maintenance is possible within the Data Warehousing Workbench of SAP NetWeaver BW.

Data types and formats

Note: the data is transferred according to the settings which are obtained with the API BAPI_DSOURCE_GETDETAIL. Here the attribute OUTPUTLEN is relevant for the length of the field filled in the transfer structure.

Example: the currency field AMOUNT with the data type "DEC" has the attributes LENG value "17", DECIMALS value "2". However OUTPUTLEN is "23". This is the relevant length to be considered for the data transfer, i.e. the space in the structure.

In general, regarding the information about internal vs. external format within the DataSource fields, please refer to the standard documentation here

Data type	Description	Example	Internal Format
ACCP	Posting period YYYYMM	199408	yyyymm
CHAR	Character strings	Hugo	any character except non-printable characters
CLNT	Client	003	3 characters
CUKY	Currency key, referenced by CURR fields	EUR	3 characters
CURR	Currency field, stored as DEC	19.50-	up to <length> numbers / point / 2 numbers / sign
D34R	Decimal Floating Point, 34 Digits, RAW on Database	-19.349057238583465693686	sign / up to 16 numbers / point / up to <decimal> numbers
DATS	Date field (YYYYMMDD) stored as char(8)	19991231	yyyymmdd
DEC	Counter or amount field with comma and sign	19.546-	up to <length> numbers / point / <decimal> numbers / sign
FLTP	Floating point number, accurate to 8 bytes	-2.0000000000000000E-03	sign / one number / dot / 16 numbers / E / sign / two numbers
INT1	1-byte integer, integer number <= 254	200-	3 number < 255 / sign
INT2	2-byte integer, only for length field before VARC or RAW	20000-	five numbers < 65536 / sign
INT4	4-byte integer, integer number with sign	2000000-	ten numbers < 4294967296 / sign
LANG	Language key	D	one character as listed in table T002
NUMC	Character field with only digits	12345	up to <length> numbers (only positive!)
QUAN	Quantity field, points to a unit field with format UNIT	19.546-	up to <length> numbers / point / <decimal> numbers / sign
RAW	Uninterpreted sequence of bytes		any character including non-printable characters
TIMS	Time field (hhmmss), stored as char(6)	162019	hhmmss
UNIT	Unit key for QUAN fields	STK	5 characters

The attribute CONVTYPE of the structure BAPIDSFIELD specifies for each field whether the data is supplied in internal or external format. Source data that is transferred as text can differ from the internal, SAP character representation (the internal format).

The possible options are:

- Internal, value 'I'
- External, value 'E'
- Check:, value 'C'

Example: Where z is a digit and v a sign, then a number with two decimal places can be represented in text as zzzz.zzv (internal, SAP representation), or, for example, vz.zzz,zz (a possible external representation).

Data Transfer in SAP internal format

Definition: The DataSource expects the data in the internal format regarding a certain field. (The attribute CONVTYPE of the structure BAPIDSFIELD is set to the value ' ').

If this option is chosen in the DataSource field the external system must return the data in the correct internal format. In this case, no checks and conversions are performed. If the data is not provided in internal format, there could be data inconsistencies or errors in the loading process.

Example: A date field (data type DATS) can always be provided with data in format yyyymmdd.

Special case: Data Transfer in Internal Format with Check

Definition: The data is expected in the DataSource as internal format. However, this will be checked in the SAP system before being passed on to the transformation. In case of errors during the check, messages are generated during the loading process.

Data Transfer in External format

Definition: The DataSource expects the data to be delivered in external format (The attribute CONVTYPE of the structure BAPIDSFIELD is set to the value 'E').

The data will be converted to internal format in the SAP system before being passed on to the Transformations.

This option shall be chosen for currency conversion or if there is a generic conversion exit available in SAP NetWeaver BW (or if the data cannot be sent in a valid internal format).

Example 1 (currency conversion):

The data for an amount field of the data type CURR is transferred together with the currency field (data type CUKY). The data has to be stored in internal format in the DataSource in order to be processed properly in the BW system. Especially regarding currency conversion it is not trivial to achieve from the external system. Thus the data is transferred in the external format.

So the DataSource field has the attribute CONVTYPE set to 'E' and CONVEXIT is ' '. The data is sent in external format and converted into internal format according to the currency reference when it is received in the BW system.

Note: Regarding loading of currencies, please refer to SAP note 1176399.

Example 2 (generic conversion exit available):

The Language code shall be transferred in external format (ISO Language code). Then the DataSource field has the attribute CONVTYPE set to 'E' and CONVEXIT is 'ISOLA'. Data is transferred as 'EN' or 'DE' instead of internal format 'E' or 'D'.

Certification

In order to get certified by SAP, a 3rd party tool has to support at least the data extraction scenario and the read only version of the design scenario.

For further information – especially about the certification process for the 3.x BAPIs - please contact your CSP manager.

Technical Prerequisite

As a technical prerequisite in order to enable the new DataSource within the SAP NetWeaver BW system it is necessary to implement the function module BAPI_DSOURCE_IS_SUPPORTED in the 3rd party system. Otherwise the BW system will not consider the 3rd party system to support the new DataSource.

Organizational Prerequisites

Prerequisites for the use of the Staging APIs of SAP NetWeaver BW are:

- Knowledge of the concept of the Staging Engine of SAP NetWeaver BW. Developers should be familiar with the terms DataSource and InfoPackage. Beyond this, the developers should have
- Knowledge in the use of APIs based on the RFC engine in general and in the selected development environment (VisualBasic, Java,).

Currently there's no C++ library available for the Staging BAPIs. In order to use the APIs you can call the corresponding RFC function modules defined below (for the existing APIs which were used already in the 3.x Staging scenario, you may use the general BAPI OCX control as well. This however does not include the DataSource specific APIs BAPI_DSOURCE* !).

You can generate a template for calling the function modules by executing the following commands in the SAP NetWeaver BW system:

1. Select *Tools* → *ABAP Workbench* → *Function Builder* from the SAP Easy Access menu.
2. Enter the name of the desired function module, e.g. BAPI_DSOURCE_GETDETAIL
3. Select menu item *Utilities* -> *RFC interface* -> *Generate*.

Please note, that for the following function modules you will have to generate the server sample code instead of the client sample code:

```
BAPI_DSOURCE_IS_SUPPORTED,
BAPI_ISOURCE_DP_REQUEST,
BAPI_ISOURCE_DP_GETPARDEF,
BAPI_ISOURCE_DP_GET_HELPVALPAR
BAPI_ISREQUEST_SEND_LOG
```

The Staging APIs which existed already in the 3.x scenario, e.g. regarding the InfoPackage, could be viewed also inside the BW system with the BAPI Explorer (Transaction BAPI). Note: the filter has to be set to all to see all Staging Interfaces.

Metadata Objects for Staging

The following metadata objects for deploying a staging scenario are available in Release 7.30 and following of SAP NetWeaver BW (not available as Business Objects):

DataSource

The DataSource encapsulates the inbound layer of the BW. It includes a field list which is not based on InfoObjects but only on ABAP/DDIC field descriptions. The Transfer Structure and Persistent Staging Area is a part of the DataSource. Here, these kinds of DataSources are distinguished:

- **DataSource with flat structure** using the data types: Transactional data, Master data Attributes, Master data Texts
- **DataSource for hierarchies** for the data type: Hierarchy

The following metadata objects for deploying a staging scenario existed already in Release 3.0 of the SAP NetWeaver BW as Business Objects:

Remote Data Provider (“Remote InfoSourceProvider”) This object is implemented on the side of the 3rd party system and encapsulates the properties and methods of a data request (*Request*) to a source system.

InfoPackage

The InfoPackage encapsulates the creation of requests in the BW. There are APIs for creating, changing and InfoPackages as well as APIs for starting and stopping requests.

Source System

The source system encapsulates the properties and methods of a source system.

DataSource with flat structure

The APIs for the flat DataSource representing the data types Transactional data, Master data Attributes and Master data Texts.

The APIs provide methods for creating and changing DataSources, for retrieving detail and status information, retrieving lists of existing DataSources.

Also there are APIs to send data from the 3rd party system to the DataSource in SAP NetWeaver BW in character and binary format.

APIs

BAPI_DSOURCE_CREATE

The API BAPI_DSOURCE_CREATE creates a DataSource in the BW system. The data type has to be set in the field DATATYPE of the parameter DSHEADERATTR (BAPIDSHEADERATTR), D: for transactional data, M for Master data attributes, T for Master data Texts.

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
DataSource Header Attributes	Attributes based on structure DSHEADERATTR
Tables parameters	
DataSource Fields	Field definitions for the DataSource (structure DATASOURCEFIELDS)
Export parameters	
RETURN	Standard BAPI result parameters

BAPI_DSOURCE_CHANGE

The API *BAPI_DSOURCE_CHANGE* changes the DataSource (in a specific a source system).

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
DataSource Header Attributes	Attributes based on structure DSHEADERATTR
Tables parameters	
DataSource Fields	Field definitions for the DataSource (structure DATASOURCEFIELDS)
Export parameters	
RETURN	Standard BAPI result parameters

BAPI_DSOURCE_ACTIVATE

The API *BAPI_DSOURCE_ACTIVATE* activates the DataSource (this includes the creation / update of the Transfer Structure and PSA table according the current settings of the DataSource).

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
Export parameters	
RETURN	Standard BAPI result parameters

BAPI_DSOURCE_EXISTS

The API *BAPI_DSOURCE_EXISTS* checks the existence of a DataSource.

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
Export parameters	
RETURN	Standard BAPI result parameters

BAPI_DSOURCE_GETSTATUS

The API *BAPI_DSOURCE_GETSTATUS* delivers the status of the DataSource.

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
Export parameters	
Active	Flag: DataSource is active
RETURN	Standard BAPI result parameters

BAPI_DSOURCE_GETDETAIL

The API *BAPI_DSOURCE_GETDETAIL* delivers detailed information about the DataSource. This contains a list of the DataSource fields which are sorted by position number.

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
Export parameters	
DataSource Header Attributes	Attributes based on structure DSHEADERATTR
DataSource Hierarchy Attributes	Obsolete, if not a Hierarchy DataSource
Customer defined	Flag: DataSource is customer defined
RETURN	Standard BAPI result parameters
Tables parameters	
DataSource Fields	Field definitions for the DataSource (structure DATASOURCEFIELDS)

BAPI_DSOURCE_GETLIST

The API *BAPI_DSOURCE_GETLIST* method delivers a list of DataSources according to the given selection criteria.

Import parameters	
SystemType	Type of the source system
Export parameters	
Return	Standard BAPI result parameters
Tables	
SelDataSource	Selection criteria for the name of the DataSource, structure BAPIDSSEL – provided by caller
SelTextLong	Selection criteria for the long text description of the DataSource, structure BAPI6100SL – provided by caller
SelSourceSystem	Selection criteria for the name of the Source System, structure BAPI6101SL – provided by caller
DataSourceList	Table DataSources (according to the selection criteria), structure BAPIDSLIST – generated by the function module

BAPI_DSOURCE_SEND_CHAR

The API *BAPI_DSOURCE_SEND_CHAR* sends data to the DataSource of the SAP NetWeaver BW in character format.

The SAP NetWeaver BW provides complete monitoring capabilities within the Data Warehousing Workbench about the loaded data.

If you send long data records each record has to be divided into fragments of exactly 250 bytes using the continuation flag in structure BAPI6100DA. This is due to a limitation in the old RFC library. The continuation flag signals that the fragment for which it is set is the continuation of the fragment that was sent before.

You could customize in the SAP NetWeaver BW a default size for data package or define the package size in the InfoPackage. This size is provided by parameter PackageSize of API *BAPI_ISOURCE_DP_REQUEST* which is described below.

It is recommended to use this number of records for creating the data packages (relevant for certification). If this parameter is not used we do recommend to send data packages with 10.000 to 100.000 records.

Parallel sending of data packages using unique package numbers is supported in order to achieve a better loading performance.

Please note: The API *BAPI_DSOURCE_SEND_CHAR* has limitations in sending data from a non-unicode to a unicode system. In this case, use the API *BAPI_DSOURCE_SEND_RAW* instead (see below).

Import parameters	
RequestID	Valid request number (see Business Object <i>DataProvider</i>)
PacketNumber	Data packet counter (1 st packet should have number 1)
LastPacket	Flag: The current data packet is the last data packet for the entered data request.
SelectionDate	Date on which the data was selected
SelectionTime	Time at which the data was selected
ExtractionError	Extraction error messages provided by the source system according to standard BAPI result parameters. For available standard error messages please refer to the online documentation.
Export parameters	
Return	Standard BAPI result parameters
Tables parameters	
Data	Table of the data records in the format of the transfer structure (max. 250 Bytes per data record, continuation records allowed, structure BAPI6100DA)

BAPI_DSOURCE_SEND_RAW

The API *BAPI_DSOURCE_SEND_RAW* sends data to the DataSource of the SAP NetWeaver BW similar to *BAPI_DSOURCE_SEND_CHAR*. This method has the only difference that the format of the data table is typed as binary data format instead of character format.

This API shall be used in case of a data transfer from a non-unicode SourceSystem to a unicode BW system.

Import parameters	
RequestID	Valid request number (see Business Object <i>DataProvider</i>)
PacketNumber	Data packet counter (1 st packet should have number 1)
LastPacket	Flag: The current data packet is the last data packet for the entered data request.
SelectionDate	Date on which the data was selected
SelectionTime	Time at which the data was selected
ExtractionError	Extraction error messages provided by the source system according to standard BAPI result parameters. For available standard error messages please refer to the online documentation.
ENCODING	
Export parameters	
Return	Standard BAPI result parameters
Tables parameters	
Data	Table of the data records in the format of the transfer structure (max. 250 Bytes per data record, continuation records allowed, structure BAPI6100DARAW)

BAPI_DSOURCE_SEND_CHARSTRING

The API *BAPI_DSOURCE_SEND_CHARSTRING* sends data to the DataSource of the SAP NetWeaver BW without splitting the data into chunks of 250 Characters, i.e. as flexible length character fields. This API is to be preferred, but cannot be used unless the data provider uses the new RFC library in its implementation (which is recommended).

Please note: The API *BAPI_DSOURCE_SEND_CHARSTRING* has limitations by sending data from a non-unicode to a unicode system. Use in this case the API *BAPI_DSOURCE_SEND_RAWSTRING* instead (see below).

Import parameters	
RequestID	Valid request number (see Business Object <i>DataProvider</i>)
PacketNumber	Data packet counter (1 st packet should have number 1)
LastPacket	Flag: The current data packet is the last data packet for the entered data request.
SelectionDate	Date on which the data was selected
SelectionTime	Time at which the data was selected
ExtractionError	Extraction error messages provided by the source system according to standard BAPI result parameters. For available standard error messages please refer to the online documentation.
ENCODING	
Export parameters	
Return	Standard BAPI result parameters
Tables parameters	
Data	Table of the data records in the format of the transfer structure (structure BAPIDSCHARSTRING)

BAPI_DSOURCE_SEND_RAWSTRING

The API *BAPI_DSOURCE_SEND_RAWSTRING* sends data to the DataSource of the SAP NetWeaver BW similar to *BAPI_DSOURCE_SEND_CHARSTRING*. This method has the only difference that the format of the data table is typed as binary data format instead of character format.

This API shall be used in case of a data transfer from a non-unicode SourceSystem to a unicode BW system.

Import parameters	
RequestID	Valid request number (see Business Object <i>DataProvider</i>)
PacketNumber	Data packet counter (1 st packet should have number 1)
LastPacket	Flag: The current data packet is the last data packet for the entered data request.
SelectionDate	Date on which the data was selected
SelectionTime	Time at which the data was selected
ExtractionError	Extraction error messages provided by the source system according to standard BAPI result parameters. For available standard error messages please refer to the online documentation.
ENCODING	
Export parameters	
Return	Standard BAPI result parameters
Tables parameters	
Data	Table of the data records in the format of the transfer structure (structure BAPIDRAWSTRING)

DataSource Hierarchy

Different to the APIs mentioned above for DataSource hierarchies there are some special APIs.

Changes to the transfer structure for hierarchies are only possible implicitly by changing the available hierarchy flags. The currently valid Transfer Structure can be obtained then using the API *BAPI_DSOURCE_GETDETAIL*.

APIs

BAPI_DSOURCE_HIER_CREATE

The API for creating a new Hierarchy DataSource for a source system is implemented in function module *BAPI_DSOURCE_HIER_CREATE*.

Import parameters	
DataSource	DataSource name
SourceSystem	Source system name
Versioned	Hierarchy is version dependent

Sorted	Hierarchy nodes are sorted
Interval	Hierarchy with interval leaves
TimeDependent	Hierarchy is time-dependent
Export parameters	
Return	Standard BAPI result parameters

BAPI_DSOURCE_HIER_CHANGE

The API *BAPI_DSOURCE_HIER_CHANGE* is used for changing a hierarchy DataSource.

Import parameters	
DataSource	DataSource name
SourceSystem	Source system name
Versioned	Hierarchy is version dependent
Sorted	Hierarchy nodes are sorted
Interval	Hierarchy with interval leaves
TimeDependent	Hierarchy is time-dependent
Export parameters	
Return	Standard BAPI result parameters

BAPI_DSOURCE_EXISTS

BAPI_DSOURCE_EXISTS, [refer APIs section](#).

BAPI_DSOURCE_GETDETAIL

BAPI_DSOURCE_GETDETAIL, [refer APIs section](#) as well.

Import parameters	
DataSource	DataSource Name
SourceSystem	Source system name
Export parameters	
DataSource Header Attributes	Attributes based on structure DSHEADERATTR
DataSource Hierarchy Attributes	Hierarchy Attributes of the DataSource
Customer defined	Flag: DataSource is customer defined
RETURN	Standard BAPI result parameters
Tables parameters	
DataSource Fields	Field definitions for the DataSource (structure DATASOURCEFIELDS)

BAPI_DSOURCE_GETLIST

BAPI_DSOURCE_GETLIST, [refer APIs section](#).

BAPI_DSOURCE_SEND_CHAR

The API for sending hierarchies to the SAP NetWeaver BW is also *BAPI_DSOURCE_SEND_CHAR*.

The transfer of delta information regarding hierarchies is currently not supported by the SAP NetWeaver BW. A hierarchy has always to be transferred completely by a single call of this method.

BAPI_DSOURCE_SEND_CHAR, [refer APIs section](#).

BAPI_DSOURCE_SEND_RAW

The API *BAPI_DSOURCE_SEND_RAW* sends hierarchies to the DataSource in the SAP NetWeaver BW in binary format, [refer APIs section](#).

BAPI_DSOURCE_SEND_CHARSTRING

The API *BAPI_DSOURCE_SEND_CHARSTRING* sends hierarchies to the DataSource in the SAP NetWeaver BW in flexible length character format, [refer APIs section](#).

BAPI_DSOURCE_SEND_RAWSTRING

The API *BAPI_DSOURCE_SEND_RAWSTRING* sends hierarchies to the DataSource in the SAP NetWeaver BW in binary format, [refer APIs section](#).

Remote Data Provider (in 3rd Party System)

The Implementation of APIs in order to build the 3rd party provider takes place in the source system, either via a 3rd Party provider or in one of the customer's own developments.

These APIs provide methods for the transmission of data requests to the source system as well as a method to query names and descriptions of application-defined additional parameters. They also provide APIs to retrieve parameters and help values from the source system to support the selection fields in the scheduler. In addition, they enable the source system to send error logs to the request monitor in the SAP NetWeaver BW on demand.

When creating a data request in the InfoPackage of the SAP NetWeaver BW additional parameters, such as *username* and *password*, can be obtained from the end user. Further examples of additional parameters are the *library name* and the *name* of a selection definition in a 3rd Party Tool. The entered values are transmitted to the source system when sending the data request (see *BAPI_ISOURCE_DP_REQUEST*).

Templates for the implementation of the APIs are available in the SAP NetWeaver BW. These are the function modules which can be used as templates for the implementation:

- *BAPI_ISOURCE_DP_REQUEST* (known as Business Object RemoteInfoSourceProvider , method *RequestInfoSourceData*)
- *BAPI_ISOURCE_DP_GETPARDEF* (known as Business Object RemoteInfoSourceProvider, method *GetParameterDefinition*)
- *BAPI_ISOURCE_DP_GET_HELPVALPAR* (known as Business Object RemoteInfoSourceProvider, method *GetParameterHelpValues*)
- *BAPI_ISREQUEST_SEND_LOG*.

APIs in detail

BAPI_ISOURCE_DP_REQUEST

The API *BAPI_ISOURCE_DP_REQUEST* sends a data request to the source system. An important attribute of this data request is the data request id that is absolutely necessary for the processing of the data in the SAP NetWeaver BW.

Note: These function module is called from the scheduler in a batch process. After receiving the request data on 3rd Party Tool side and splitting an asynchronous job here, the control of the request process (RFC call) has be returned back to the SAP system. If the 3rd Party Tool hold this RFC call open until the total process is finished, a work process of the application server in the SAP NetWeaver BW is blocked for the whole period or runs into a timeout.

Import parameters	
RequestID	Unique ID of the data request
InfoSource	InfoSource name
SourceSystem	Source system name
Type	Type of the requested data
Date	Request date
Time	Request time
UserName	Name of user making request
UpdateMode	Delta, Full, Initial
Hierarchy	Selection based on a hierarchy request (only necessary for hierarchy requests, structure BAPI6107HI)
PackageSize	Expected PackageSize of a package
Export parameters	
Return	Standard BAPI return parameter. For available standard error messages please refer to the online documentation.
Tables parameters	
SelData	Selection criteria for transaction data and master data attributes (only for requests for transaction or master data, structure BAPI6107DA)
SelLanguages	Selection criteria for texts for master data (Language) (only for requests for texts, structure BAPI6107LA)
Parameters	Values of the application-based parameters (only where necessary, structure BAPI6107PA)

BAPI_ISOURCE_DP_GETPARDEF

The API *BAPI_ISOURCE_DP_GETPARDEF* is implemented in the 3rd Party system and retrieves a table of parameters needed by the source system with parameter names, descriptions, data type, maximum length and the information as to which of the parameters are mandatory.

The API is called during creating of an InfoPackage or when the user pushes the refresh button in the tab strip "3rd selections".

Export parameters	
Return	Standard BAPI return parameter. For available standard error messages please refer to the online documentation.
Tables parameters	
ParameterDefinition	Definition of the application-based parameters (structure BAPI6107PD)

BAPI_ISOURCE_DP_GET_HELPVALPAR

The API *BAPI_ISOURCE_DP_GET_HELPVALPAR* retrieves from the 3rd party system a table of help values for parameters requested by the API *BAPI_ISOURCE_DP_GETPARDEF*, described above. These fields are displayed in the BW InfoPackage in the tab strip "3rd selections". The API is called when the user executes the value help (F4) button in the parameter field.

Import parameters	
Keyname	3rd party tool specific key name
Langu	Language for which the help values should be delivered
Export parameters	
Return	Standard BAPI return parameter. For available standard error messages please refer to the online documentation.
Tables parameters	
PreviousSelList	List of the already selected values for other fields (structure BAPI6107PA)
HelpValuesList	List of possible values for the selection field (structure BAPI6107HV)

BAPI_ISREQUEST_SEND_LOG

The API *BAPI_ISREQUEST_SEND_LOG* retrieves a table with messages from the 3rd party system. With the parameter *message_type* the desired message type(s) of messages are defined (errors, errors and warning, all).

Import parameters	
RequestID	Unique ID of the data request
FieldNm	Name of the selection field
Langu	Language for which the help values should be delivered
Export parameters	
Return	Standard BAPI return parameter. For available standard error messages please refer to the online documentation.
Tables parameters	
Parameters	Values of the application-based parameters (only where necessary, structure BAPI6107PA)
Messages	List of messages occurring during the extraction / sending of data (structure BAPI6114MSG)

After a data request was sent to the 3rd party system information about it can be obtained from the BW.

BAPI_ISREQUEST_GETSTATUS

The API *BAPI_ISREQUEST_GETSTATUS* determines the current status of a Request, it is implemented in the BW system.

Import parameters	
RequestID	ID of the request
Export parameters	
TechStatus	Technical status of the request (G, Y, R)
TechInfo	Information to the technical status (plain text)
QualStatus	Manual set status of the request (G, Y, R, initial)
QualInfo	Information to the manual set status (plain text)
Return	Standard BAPI return parameter (here as a table)

InfoPackage

An InfoPackage is the entry point for the loading processes of a specific DataSource within a specific source system. APIs for creating and change of an InfoPackage exist as well as for starting and stopping of load processes.

For the details about the APIs around the InfoPackage (e.g. creation, change, start, stop), please refer to the 3.x certification paper: "Staging BAPIs" [please contact ssstp@sap.com for the document].

Source System

Source systems contain DataSources belonging logically together. It is possible to create source systems and obtain its properties with this set of APIs.

For the details about the APIs around the Source System (e.g. creation, get detail), please refer to the 3.x certification paper: "Staging BAPIs" [please contact ssstp@sap.com for the document].

ABAP Dictionary Structures

This section contains a technical description of the ABAP Dictionary structures used and their components. For more details on single data fields, especially with respect to possible values, we refer to the ABAP dictionary on your SAP NetWeaver BW system.

BAPIDSHEADER

The Dictionary structure BAPIDSHEADER is used for reading and writing DataSources.

Field name	Type	Length	Description
DATASOURCE	CHAR	30	DataSource name
SOURCESYSTEM	CHAR	10	Logsys name

BAPIDSHEADERATTR

The Dictionary structure BAPIDSHEADERATTR is used for transferring header attributes from and to the BW.

Field name	Type	Length	Description
TEXTSHORT	CHAR	20	Short description
TEXTMEDIUM	CHAR	40	Medium description
TEXTLONG	CHAR	60	Long description
DATATYPE	CHAR	1	Data Type of a DataSource
DELTAMETHOD	CHAR	4	DataSource: Delta Process
PACKGUPD	CHAR	1	Repeated request of a data packet supported
LANGUFIELD	CHAR	30	Field Name: Language Key
DATEFROMFIELD	CHAR	30	Field Name: Lower Limit of Validity
DATETOFIELD	CHAR	30	Field Name: Upper Limit of Validity
MODIFICATIONDATE	DATS	8	Last changed on
MODIFICATIONTIME	TIMS	6	Time of Last Change
CHAR_PSA	CHAR	1	PSA for All Segments in CHAR Format
BULK_ENABLED	CHAR	1	Checkbox

BAPIDSHIER

The Dictionary structure BAPIDSHIER is used for transferring hierarchy information from and to the BW.

Field name	Type	Length	Description
TIMEDEPENDENCY	CHAR	1	Time Hierarchy Flag
VERSIONS_ALLOWED	CHAR	1	Hierarchies can have several versions
INTERVALS_ALLOWED	CHAR	1	Intervals permitted in hierarchy
SIGNS_ALLOWED	CHAR	1	Reverse Sign for Hierarchy Nodes
SORTED_HIERARCHY	CHAR	1	Hierarchy transfer with sorted node(s)
FLEXIBLE_HIERARCHY	CHAR	1	Flexible Hierarchy Flag
INFOOBJECT	CHAR	30	InfoObject

BAPIDSFIELD

The Dictionary structure BAPIDSFIELD is used for transferring field information from and to the BW.

Field name	Type	Length	Description
POSITION	NUMC	4	Position of the field in the table
TRANSFER	CHAR	1	Field Will Be Transferred (In PSA and Structures)

SELECTABLE	CHAR	1	Selection
INFOBJECT	CHAR	30	InfoObject
FIELDNM	CHAR	30	Field name
DATATYPE	CHAR	4	Data type
LENG	NUMC	6	Length (No. of Characters)
DECIMALS	NUMC	6	Number of Decimal Places
OUTPUTLEN	NUMC	6	Output Length
INTTYPE	CHAR	1	ABAP data type (C,D,N,...)
INTLEN	NUMC	6	Internal Length in Bytes
LOWERCASE	CHAR	1	Lowercase letters allowed/not allowed
KEYFIELD	CHAR	1	Key Field
CONVEXIT	CHAR	5	Conversion Routine
UNIFIELDNM	CHAR	30	Name of the relevant unit field
TEXTSHORT	CHAR	20	Short description
TEXTMEDIUM	CHAR	40	Medium description
TEXTLONG	CHAR	60	Long description
CONVTYPE	CHAR	1	Data Input Format (Internal/External)

BAPIDSLIST

The Dictionary structure BAPIDSLIST is used for passing a list of DataSource names with descriptions to the 3rd Party System.

Field name	Type	Length	Description
DATASOURCE	CHAR	30	DataSource
SOURCESYSTEM	CHAR	10	Source system
TYPE	CHAR	1	Data Type of a DataSource
TEXTSHORT	CHAR	20	Short description
TEXTMEDIUM	CHAR	40	Medium description
TEXTLONG	CHAR	60	Long description

BAPIDSSSEL

The Dictionary structure BAPIDSSSEL is used for passing selection criteria of the DataSource name when obtaining DataSources.

Field name	Type	Length	Description
SIGN	CHAR	1	Inclusion/exclusion criterion SIGN for range tables
OPTION	CHAR	2	Selection operator OPTION for range tables
DATASOURCELOW	CHAR	30	DataSource
DATASOURCEHIGH	CHAR	30	DataSource

BAPIDSTYPESEL

The Dictionary structure BAPIDSTYPESEL is used for passing selections criteria of the data type when obtaining DataSources.

Field name	Type	Length	Description
SIGN	CHAR	1	Inclusion/exclusion criterion SIGN for range tables
OPTION	CHAR	2	Selection operator OPTION for range tables
TYPELOW	CHAR	1	Data Type of a DataSource
TYPEHIGH	CHAR	1	Data Type of a DataSource

BAPI6100DARAW

The Dictionary structure *BAPI6100DARAW* is used for sending data to the SAP NetWeaver BW Server in a binary format. Due to limitations in the BAPI interface, data records have to be fragmented into pieces of 250 bytes. Continuation records are identified by the continuation flag.

Field name	Type	Length	Description
CONTINUATION	CHAR	1	Flag: Continuation record
DATA	RAW	250	Data record

BAPI6100DA

The Dictionary structure *BAPI6100DA* is used for sending data to the SAP NetWeaver BW Server. Due to limitations in the BAPI interface, data records have to be fragmented into pieces of 250 bytes. Continuation records are identified by the continuation flag.

Field name	Type	Length	Description
CONTINUATION	CHAR	1	Flag: Continuation record
DATA	CHAR	250	Data record

BAPI6100SL

The dictionary structure *BAPI6100SL* is used for the specification of selection criteria for long text descriptions.

Field name	Type	Length	Description
SIGN	CHAR	1	Selection operator (see ranges tables)
OPTION	CHAR	2	Selection operator (see ranges tables)
TEXTLONGLOW	CHAR	60	Description (lower bound)
TEXTLONGHIGH	CHAR	60	Description (upper bound)

BAPI6101SL

The dictionary structure *BAPI6101SL* is used for the specification of selection criteria for source systems.

Field name	Type	Length	Description
SIGN	CHAR	1	Selection operator (see ranges tables)
OPTION	CHAR	2	Selection operator (see ranges tables)
SOURCESYSTEMLOW	CHAR	10	Source system name (lower bound)
SOURCESYSTEMHIGH	CHAR	10	Source system name (upper bound)

BAPI6104

The dictionary structure *BAPI6104* is used for details about master (data transfer) DataSources (InfoSources).

Field name	Type	Length	Description
INFOSOURCE	CHAR	30	Name of the InfoSource
SOURCESYSTEM	CHAR	10	Name of the source system
TEXTSHORT	CHAR	20	Short description
TEXTMEDIUM	CHAR	40	Medium length description
TEXTLONG	CHAR	60	Long description
MSDFLAG	CHAR	1	Flag: InfoSource provides master data attributes

TXTFLAG	CHAR	1	Flag: InfoSource supports text information
HIEFLAG	CHAR	1	Flag: InfoSource supports hierarchies
TIMDEPFL	CHAR	1	Flag: InfoSource supports time dependent attributes
DELTAUPD	CHAR	1	Flag: InfoSource supports delta updates
PACKGUPD	CHAR	1	Flag: InfoSource supports reloading data requests
CUSTOMERDEFINED	CHAR	1	Flag: InfoSource is defined by the customer
ACTIVE	CHAR	1	Flag: InfoSource is active (data transmission is possible)
TRANSTRUCTURE	CHAR	27	Name of the transfer structure
MODIFICATIONDATE	DATS	8	Modification date of transfer structure
MODIFICATIONTIME	TIMS	6	Modification time of transfer structure

BAPI6107DP

The dictionary structure *BAPI6107DP* is used for general data request parameters.

Field name	Type	Length	Description
REQUEST	CHAR	30	Request number
PACKETNUMBER	NUMC	6	Current data packet number
LASTPACKET	CHAR	1	Flag: Current packet is the last data packet
SELECTIONDATE	DATS	8	Selection date
SELECTIONTIME	TIMS	6	Selection time

Copyright

© Copyright 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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.