

Using Timestamps in Generic Delta Extraction by Function Module



Applies to:

SAP BW3.5, BI7.0

For more information, visit the [Business Intelligence homepage](#).

Summary

Timestamps are often required as delta criteria for generic delta extraction. However, in many tables such the timestamp field is not available; instead the creation/change date and time are available. Generic delta needs to function on one field. This article explains how to provide delta for these tables using timestamps by a workaround.

Author: Suhas Karnik

Company: Wipro Technologies

Created on: 5 Jun 2009

Author Bio

Suhas Karnik is an SAP BW consultant with Wipro Technologies, Bangalore. He has been working on BW for two years now.

Table of Contents

The Scenario	3
Creating the Extract Structure	4
Reason for Addition of a Timestamp Field in Extract Structure	5
Creating the function module.....	6
Explanation of the Code.....	9
Putting it All Together: Create the Datasource	10
How it Works.....	15
Related Content.....	16
Disclaimer and Liability Notice.....	17

The Scenario

The data on orders is being extracted from SAP daily into BW on the basis of order creations and changes on the previous day. Currently this is being done by a nightly job which runs once every 24 hours. It occurs using a pseudo-delta i.e. there are date-based selection conditions specified in the info package level.

The users need this data for some reports which are critical for operating the business. As a result, there is a requirement to provide this using delta functionality so that the extractor can be run multiple times in a day. This therefore requires the extraction to be dependent on the timestamp for creation/change instead of only the date of creation/change.

In the orders table (AUFK), however, the timestamp for the creation or change is not stored, though the date and time for creation/change are stored in the following fields:

ERDAT: Creation Date

ERFZEIT: Creation Time

AEDAT: Last Change Date

AEZEIT: Last Change Time

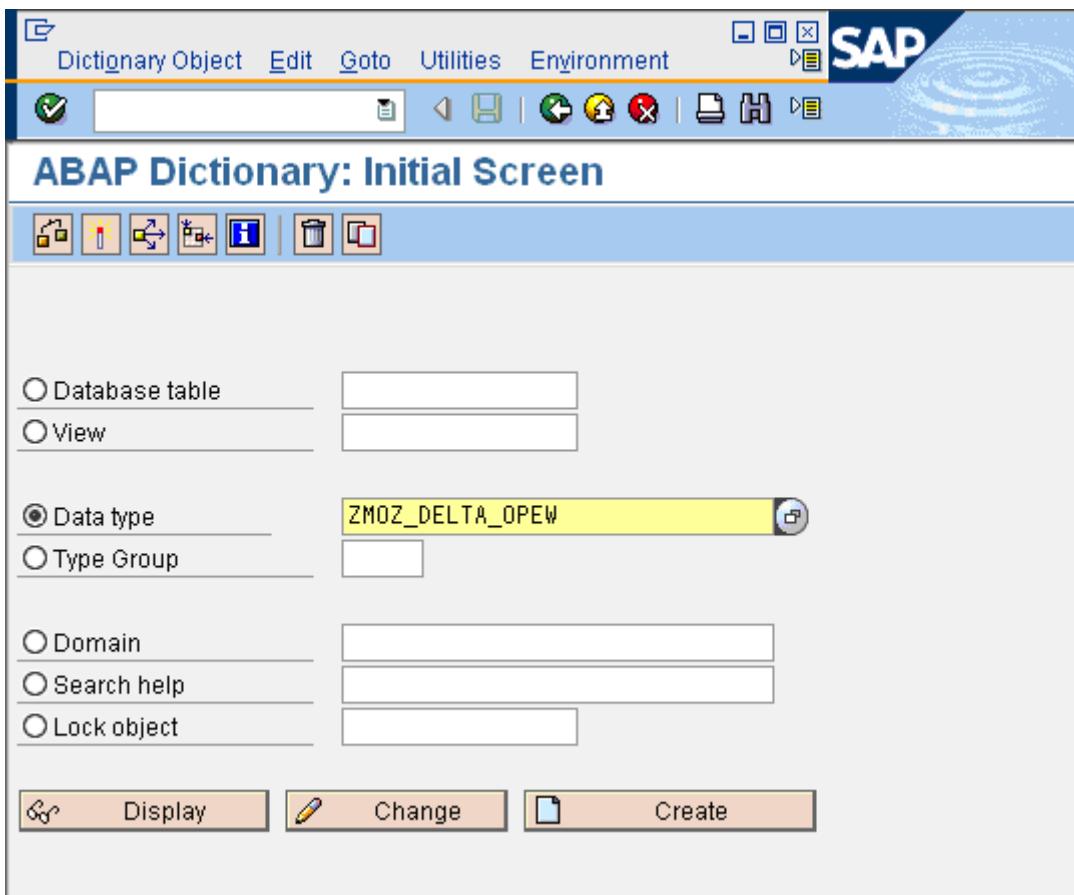
This document describes a workaround to obtaining the timestamp delta functionality using a generic extractor.

The key to understanding the workaround is to look into how the timestamp selection is normally conveyed to the extractor. When BW requests a delta, the timestamp is sent to the function module just as normal selection criteria are sent. This can be observed easily by writing to an ad-hoc log table while the extraction is running. This will be covered later as we look into the code.

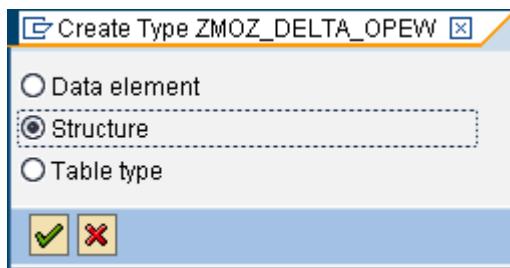
The basics on the creation of Generic extractors using function modules are covered in [this document](#).

Creating the Extract Structure

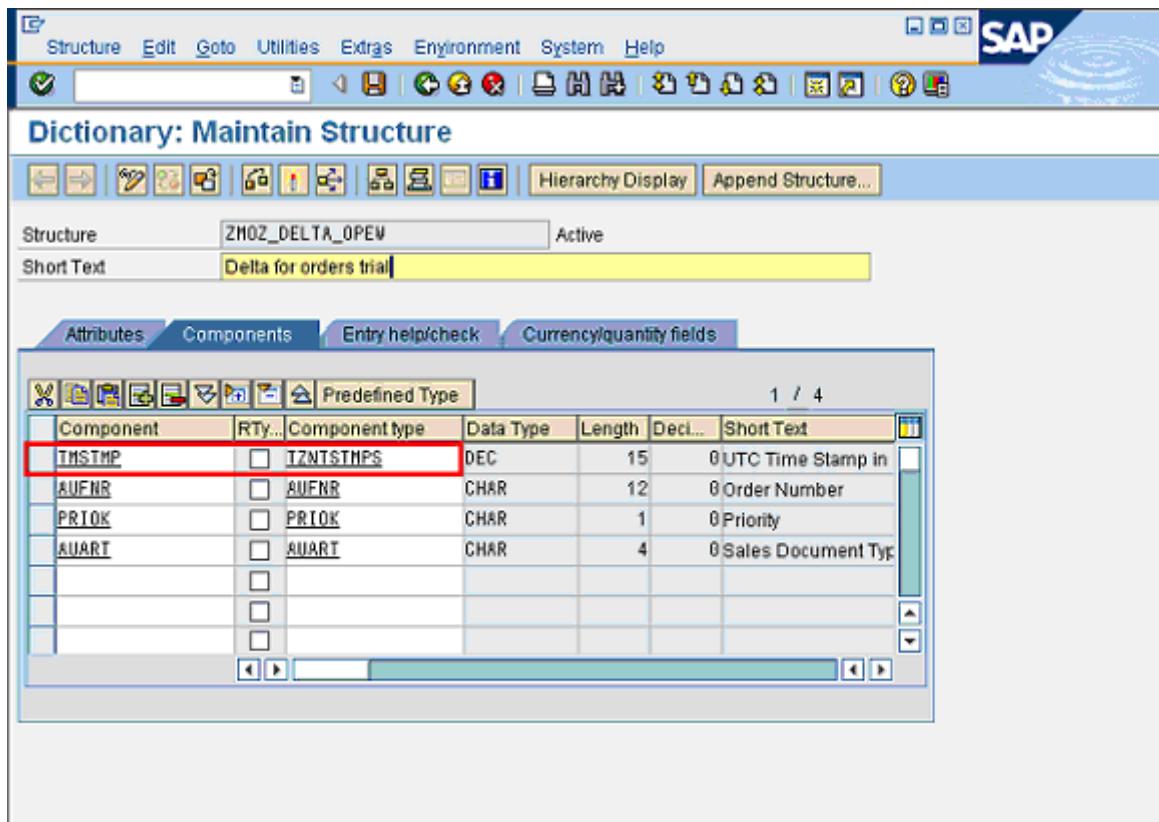
Go to transaction SE11 and try to create a new structure. Select the *Data type* radio button, enter the name of your extract structure and click *Create*.



Select *Structure* and click *OK*.



Add the fields you want to pick up using the extractor. Also add a field of the type TZNTSTMP. This is the field that will hold the timestamp.



Reason for Addition of a Timestamp Field in Extract Structure

While configuring delta in the RSO2 screen, the field on which the delta is requested must be a field present in the extract structure. To allow the extractor to provide delta on timestamp, there must be a timestamp field in the extract structure. Hence the timestamp field is added here – it is merely a dummy field created to allow us to use the extractor for delta purposes, as will become clear later.

Creating the function module

Go to SE80 and copy the function group *RSAX* and the function module *RSAX_BIW_GET_DATA_SIMPLE* into your own function group and module. Name the function module as *ZMOZ_DELTA_OPEW*. The name of the function module is the same as the name of the extract structure, but this is purely for convenience and is not mandatory; the FM can be given a different name if required. The code to be changed is highlighted below.

```

FUNCTION ZMOZ_DELTA_OPEW.
* -----
* ** Local Interface:
*   IMPORTING
*     VALUE(I_REQUNR) TYPE SRSC_S_IF_SIMPLE-REQUNR
*     VALUE(I_DSOURCE) TYPE SRSC_S_IF_SIMPLE-DSOURCE OPTIONAL
*     VALUE(I_MAXSIZE) TYPE SRSC_S_IF_SIMPLE-MAXSIZE OPTIONAL
*     VALUE(I_INITFLAG) TYPE SRSC_S_IF_SIMPLE-INITFLAG OPTIONAL
*     VALUE(I_READ_ONLY) TYPE SRSC_S_IF_SIMPLE-READONLY OPTIONAL
*     VALUE(I_REMOTE_CALL) TYPE SBIWA_FLAG DEFAULT SBIWA_C_FLAG_OFF
*   TABLES
*     I_T_SELECT TYPE SRSC_S_IF_SIMPLE-T_SELECT OPTIONAL
*     I_T_FIELDS TYPE SRSC_S_IF_SIMPLE-T_FIELDS OPTIONAL
*     E_T_DATA STRUCTURE ZMOZ_DELTA_OPEW OPTIONAL
*   EXCEPTIONS
*     NO_MORE_DATA
*     ERROR_PASSED_TO_MESS_HANDLER
*   -----
*
* Auxiliary Selection criteria structure
DATA: l_s_select TYPE srsc_s_select.

* Maximum number of lines for DB table
STATIC: s_s_if TYPE srsc_s_if_simple,

* counter
      s_counter_datapakid LIKE sy-tabix,

* cursor
      s_cursor TYPE cursor.
* Select ranges
      RANGES: l_r_aufnr FOR aufk-aufnr,
              l_r_tmstmp FOR ZMOZ_DELTA_OPEW-TMSTMP.

      data: startDate like sy-datum,
            startTime like sy-uzeit,
            endDate like sy-datum,
            endTime like sy-uzeit,
            tstamp like tzonref-tstamps.

      ranges: l_r_erdat for aufk-erdat,
              l_r_erfzeit for aufk-erfzeit.
* Initialization mode (first call by SAPI) or data transfer mode
* (following calls) ?
IF i_initflag = sbiwa_c_flag_on.

*****
* Initialization: check input parameters

```

```

*          buffer input parameters
*          prepare data selection
*****



* Check DataSource validity
CASE i_dsource.
  WHEN 'ZMOZ_DELTA_OPEW'.
  WHEN OTHERS.
    IF 1 = 2. MESSAGE e009(r3). ENDIF.
* this is a typical log call. Please write every error message like this
  log_write 'E'           "message type
             'R3'          "message class
             '009'          "message number
             i_dsource     "message variable 1
             ' '            "message variable 2
  RAISE error_passed_to_mess_handler.
ENDCASE.

APPEND LINES OF i_t_select TO s_s_if-t_select.

* Fill parameter buffer for data extraction calls
  s_s_if-requnr   = i_requnr.
  s_s_if-dsource = i_dsource.
  s_s_if-maxsize = i_maxsize.

* Fill field list table for an optimized select statement
* (in case that there is no 1:1 relation between InfoSource fields
* and database table fields this may be far from being trivial)
  APPEND LINES OF i_t_fields TO s_s_if-t_fields.

ELSE.           "Initialization mode or data extraction ?

*****



* Data transfer: First Call      OPEN CURSOR + FETCH
*                 Following Calls FETCH only
*****



* First data package -> OPEN CURSOR
  IF s_counter_datapakid = 0.

* Fill range tables BW will only pass down simple selection criteria
* of the type SIGN = 'I' and OPTION = 'EQ' or OPTION = 'BT'.
  LOOP AT s_s_if-t_select INTO l_s_select WHERE fieldnm = 'AUFNR'.
    MOVE-CORRESPONDING l_s_select TO l_r_aufnr.
    APPEND l_r_aufnr.
  ENDLOOP.

* Timestamp is delivered as a selection criterion.
* Split the timestamp into date and time
  LOOP AT s_s_if-t_select INTO l_s_select WHERE fieldnm = 'TMSTMP'.
    tstamp = l_s_select-low.
    convert TIME STAMP tstamp TIME ZONE sy-zonlo
               into DATE startDate TIME startTime.

    tstamp = l_s_select-high.

```

```

        convert TIME STAMP tstamp TIME ZONE sy-zonlo
        into DATE endDate TIME endTime.

        l_r_erd़at-low = startDate.
        l_r_erd़at-sign = l_s_select-sign.
        l_r_erd़at-option = l_s_select-option.
        l_r_erd़at-high = endDate.
        append l_r_erd़at.

        l_r_efzeit-low = startTime.
        l_r_efzeit-sign = l_s_select-sign.
        l_r_efzeit-option = l_s_select-option.
        l_r_efzeit-high = endTime.
        append l_r_efzeit.

    ENDLOOP.

* Determine number of database records to be read per FETCH statement
* from input parameter I_MAXSIZE. If there is a one to one relation
* between DataSource table lines and database entries, this is trivial.
* In other cases, it may be impossible and some estimated value has to
* be determined.

OPEN CURSOR WITH HOLD s_cursor FOR

* Use the l_r_erd़at and l_r_efzeit for both creation and change selections
* This way we can pick up both the creations and changes in a given time period.

        SELECT aufnr FROM aufk
        WHERE aufnr IN l_r_aufnr
        AND (
                ( erdat >= startdate and ( erfzeit >= starttime OR ( erdat <=
enddate and erfzeit <= endtime ) ) OR ( aedat >= startdate and ( aezeit >=
starttime OR ( aedat <= enddate and aezeit <= endtime ) )
                ).
ENDIF.                                "First data package ?

* Fetch records into interface table.
* named E_T_'Name of extract structure'.

FETCH NEXT CURSOR s_cursor
        APPENDING CORRESPONDING FIELDS
        OF TABLE e_t_data
        PACKAGE SIZE s_s_if-maxsize.

IF sy-subrc <> 0.
    CLOSE CURSOR s_cursor.
    RAISE no_more_data.
ENDIF.

s_counter_datapakid = s_counter_datapakid + 1.

ENDIF.                                "Initialization mode or data extraction ?

ENDFUNCTION.

```

Explanation of the Code

```
RANGES: l_r_aufnr FOR aufk-aufnr,
        l_r_tmstmp FOR ZMOZ_DELTA_OPEW-TMSTMP.
```

and

```
ranges: l_r_erdat for aufk-erdat,
        l_r_erfzeit for aufk-erfzeit.
```

The *l_r_tmstmp* range is created for the timestamp. The selection criteria for the extractor will be filled up into this range. The values in this range are later split and filled into *l_r_erdat* and *l_r_erfzeit* ranges.

```
LOOP AT s_s_if-t_select INTO l_s_select WHERE fieldnm = 'TMSTMP'.
  tstamp = l_s_select-low.
  convert TIME STAMP tstamp TIME ZONE sy-zonlo
    into DATE startDate TIME startTime.
  ...

```

This part of the code splits the received timestamp and converts it into the date and time. The *low* field of the range yields the starting date and time, while the *high* field gives the ending date and time.

```
l_r_erdat-low = startDate.
l_r_erdat-sign = l_s_select-sign.
l_r_erdat-option = l_s_select-option.
l_r_erdat-high = endDate.
append l_r_erdat.
```

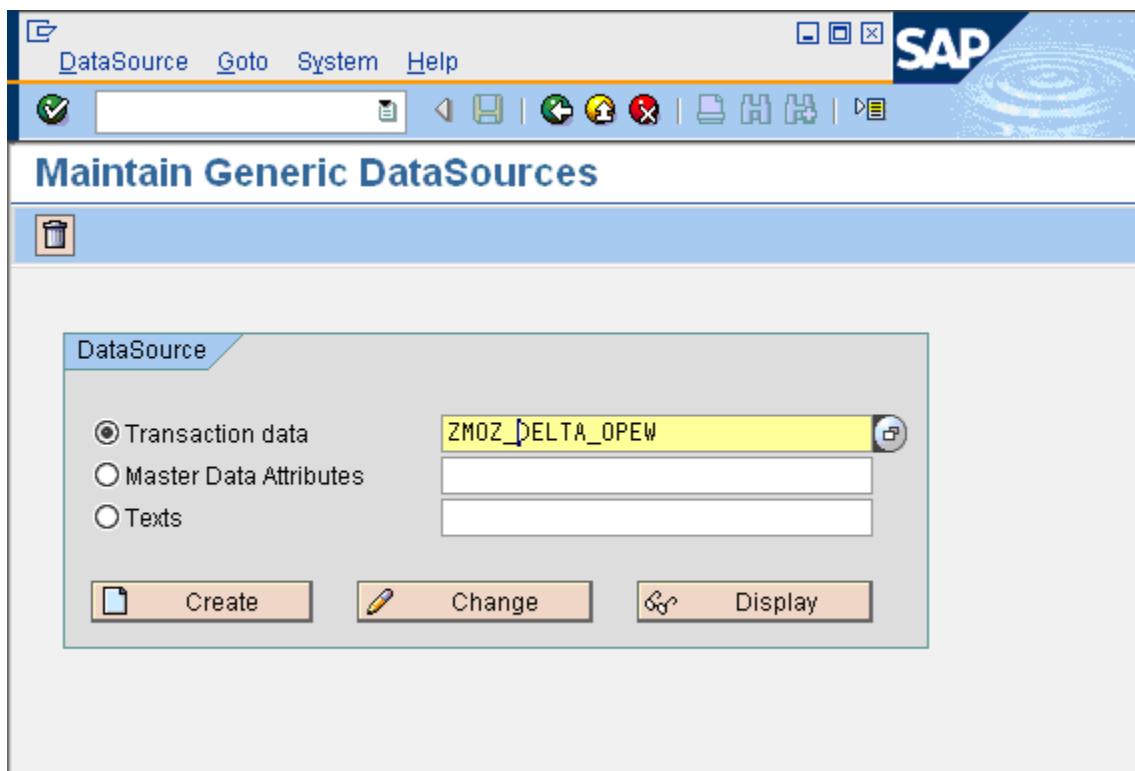
The *l_r_erdat* range is filled with *startDate* and *endDate* values obtained by parsing the timestamp. A similar process occurs with the *startTime* and the *endTime*.

```
SELECT  aufnr FROM aufk
        WHERE aufnr IN  l_r_aufnr
        AND (
          ( erdat >= startdate and ( erfzeit >= starttime OR ( erdat <=
enddate and erfzeit <= endtime ) ) ) OR ( aedat >= startdate and ( aezeit >=
starttime OR ( aedat <= enddate and aezeit <= endtime ) ) )
        ).
```

The ranges obtained by splitting the timestamp are used for selecting data based on both creation and change of the order. Before implementing this statement, please consider the indexing that is present on the table, the number of records in production etc.

Putting it All Together: Create the Datasource

Go to transaction RSO2 and create a new datasource.



Enter the short text and click the *Extraction by FM* option.

Create DataSource for Transactn data: ZMOZ_DELTA_OPEW

Generic Delta

DataSource	ZMOZ_DELTA_OPEW	Extraction from View
Appl. Component	PM	Extraction from Query
Reconciliation	<input type="checkbox"/>	Extraction by FM
Obj. status	New	

Texts

Short description	Delta Trial
Medium description	Delta Trial
Long description	Delta Trial

Extraction from DB View

View/Table	
ExtractStruct.	

Extraction from SAP Query

InfoSet	
---------	--

Extraction by Function Module

Function Module	
Extract.Struct.	

Enter the name of the function module and the extract structure and click *Generic Delta*.

Create DataSource for Transactn data: ZMOZ_DELTA_OPEW

Generic Delta

DataSource	ZMOZ_DELTA_OPEW	Extraction from View
Applc. Component	PM	Extraction from Query
Reconciliation	<input type="checkbox"/>	Extraction by FM
Obj. status	New	

Texts

Short description	Delta Trial
Medium description	Delta Trial
Long description	Delta Trial

Extraction from DB View

View/Table	
ExtractStruct.	

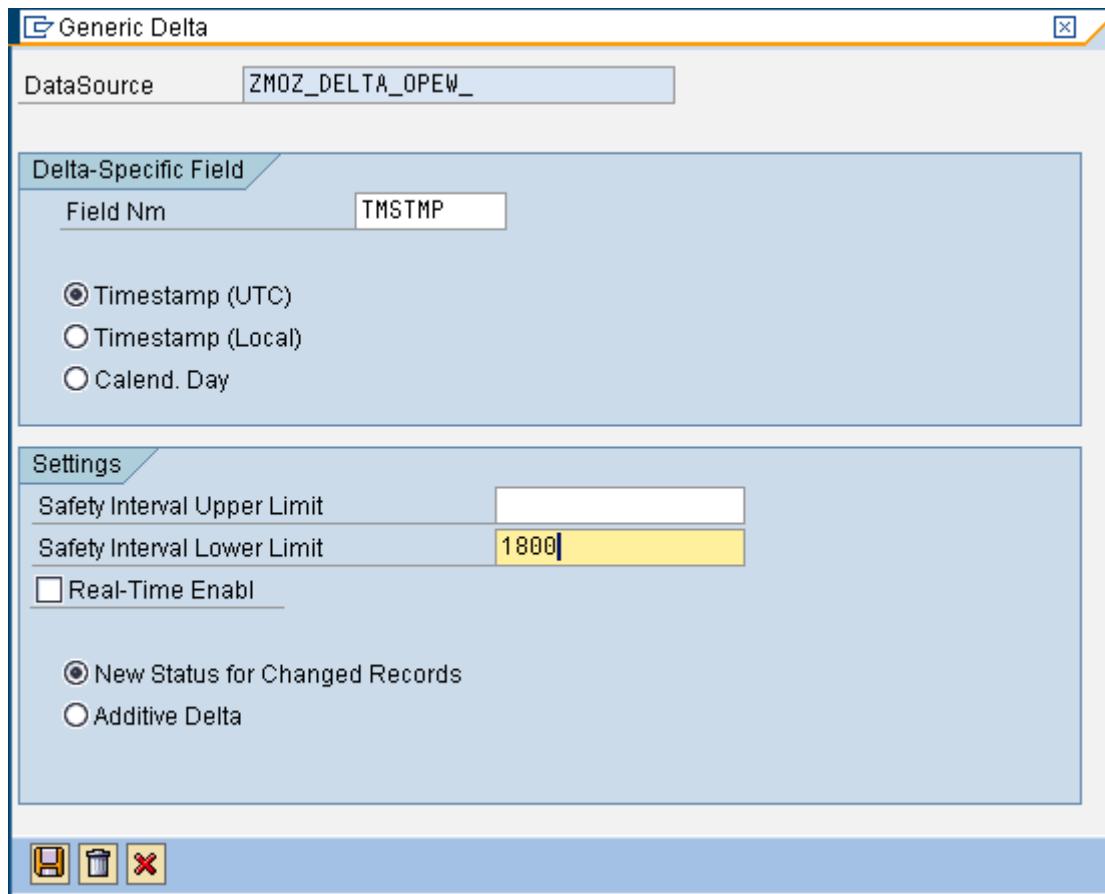
Extraction frm SAP Query

InfoSet	
---------	--

Extraction by Function Module

Function Module	ZMOZ_DELTA_OPEW
Extract.Struct.	ZMOZ_DELTA_OPEW 

Select the option *Timestamp (UTC)* and select the timestamp field you had added in your extract structure. Also set the safety lower interval to 1800.



Click Save to go back to the previous screen and click Save again. The following screen comes up.

DataSource: Customer version Edit

Header Data																																													
DataSource:	ZMOZ_DELTA_OPEW	Package																																											
Description	Delta Trial																																												
Extraction																																													
ExtractStruct:	ZMOZ_DELTA_OPEW	DataSource for Reconciliation	<input type="checkbox"/>																																										
Direct Access	1																																												
Delta Update	<input checked="" type="checkbox"/>																																												
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Short text</th> <th>Selection</th> <th>Hide field</th> <th>Inversion</th> <th>Field only</th> <th></th> </tr> </thead> <tbody> <tr> <td>TMSTMP</td> <td>UTC Time Stamp in Short Form (YYYYMMDDHHMMSS)</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>AUFNR</td> <td>Order Number</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>PRIOK</td> <td>Priority</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>AUART</td> <td>Sales Document Type</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				Field Name	Short text	Selection	Hide field	Inversion	Field only		TMSTMP	UTC Time Stamp in Short Form (YYYYMMDDHHMMSS)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		AUFNR	Order Number	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		PRIOK	Priority	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		AUART	Sales Document Type	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>								
Field Name	Short text	Selection	Hide field	Inversion	Field only																																								
TMSTMP	UTC Time Stamp in Short Form (YYYYMMDDHHMMSS)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																								
AUFNR	Order Number	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																								
PRIOK	Priority	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																								
AUART	Sales Document Type	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																								

Note that the timestamp field is disabled for selection; this is because the timestamp value will be automatically populated as part of the delta. As a result, it will be unavailable for manual entry in the Infopackage or in RSA3.

Following this step, create the corresponding Infosource, ODS/Cube, Datasource etc in BW side and replicate. These steps are similar to what would be done for a normal generic datasource.

How it Works

To understand how the timestamp field is passed to the extractor, you can add a bit of code to write the values received by the extractor, to a table. This is purely for your own understanding and does not affect the extraction process itself. Later on, looking at the values in the table would give an idea of how the parameter was passed.

For instance, when the above extractor was run, the values it received, and the corresponding values of start and end date and time were as below.

	Date & Time of extraction	LOW	OPT	SIGN	HIGH	Start Date & Time	End Date & Time
Init	04.06.2009 06:23:19		BT	I	20090604052319		04/06/2009 06:23:19
Delta	04.06.2009 06:23:45	20090604045319	BT	I	20090604052345	04/06/2009 05:53:19	04/06/2009 06:23:45

Some observations from the above table:

1. During initialization, the lower limit value is blank. The higher limit is the current time.
2. During delta request, the lower limit value was 30 minutes older than the higher limit value of the previous (i.e. init) delta request. This is due to the fact that we had given a safety lower interval of 1800. As a result, the lower interval was taken as the previous higher interval limit minus 30 minutes (i.e. 1800 seconds)
3. There is a difference of one hour between the Time of extraction column (which shows the system time when the delta request was received in the source system) and the Low and High fields. This is due to the fact that the timestamp was configured as a UTC timestamp, and the time zone of the system is UTC+1 (i.e. one hour ahead of UTC). The same difference exists between the Low and High fields and the Start/End Date & Time, which are the fields obtained when the low and high timestamps were split. The difference in this case is also due to the time zone.

Related Content

[Generic Extraction using a Function Module](#)

[How to create generic delta](#)

[Documentation on Generic Extraction on SAP help](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.