# Deploying the Metadata Extraction ABAP Function Module for the SAP BW Family of Bridges

## Overview

The SAP BW bridge may be used to extract metadata from the SAP Business Warehouse. It does so by making use of an ABAP function module that one places on the SAP server.  This function module is remotely accessible by the bridge to query and download (as a text file) the metadata containing within the SAP server.  When executed, the bridge then either

- Imports metadata from an existing file already downloaded locally from a previous bridge execution.
- Invokes the ABAP module API, waits for the download, and then imports metadata from that file.

This document details the steps required to deploy* this module.

*The following instruction was compiled based on the SAP GUI v.7.2

## Delivery of the Module

The module text itself is delivered as “Z_MITI_BW_DOWNLOAD.txt” and must be requested.

## Deployment of the Module

### Create a Package

This step is optional and an existing package can be used.

1. Login to the SAP BW server and open "Object Navigator" (SE80 transaction).

2. Select option "Package" from the list and enter a name for the new package (e.g. Z_MITI) then press button ‘Display’

4. Press ‘Yes’ in the ‘Create Package’ window. Consequently, a window "Package Builder: Create Package" will be opened. Enter value into ‘Short Description’ field and press the "Continue" icon.

5. Press ‘Own Requests’ in the ‘Prompt for local Workbench request’ window. Select ‘development’ request.

## Create a Function Group

In Object Navigator select "Function Group" from the list and type its name in the input field below (e.g. Z_MITI_BW_FGROUP). Press View icon.

In "Create Object" window press yes to create a new function group.

Specify an appropriate description in the "Short text" field and press button "Save".

Choose a package which was prepared on the previous stage "Create a Package" and click on icon "Save".

Confirm a request by pressing icon "Continue".

Activate this Function Group.

## Create the ABAP Function Module

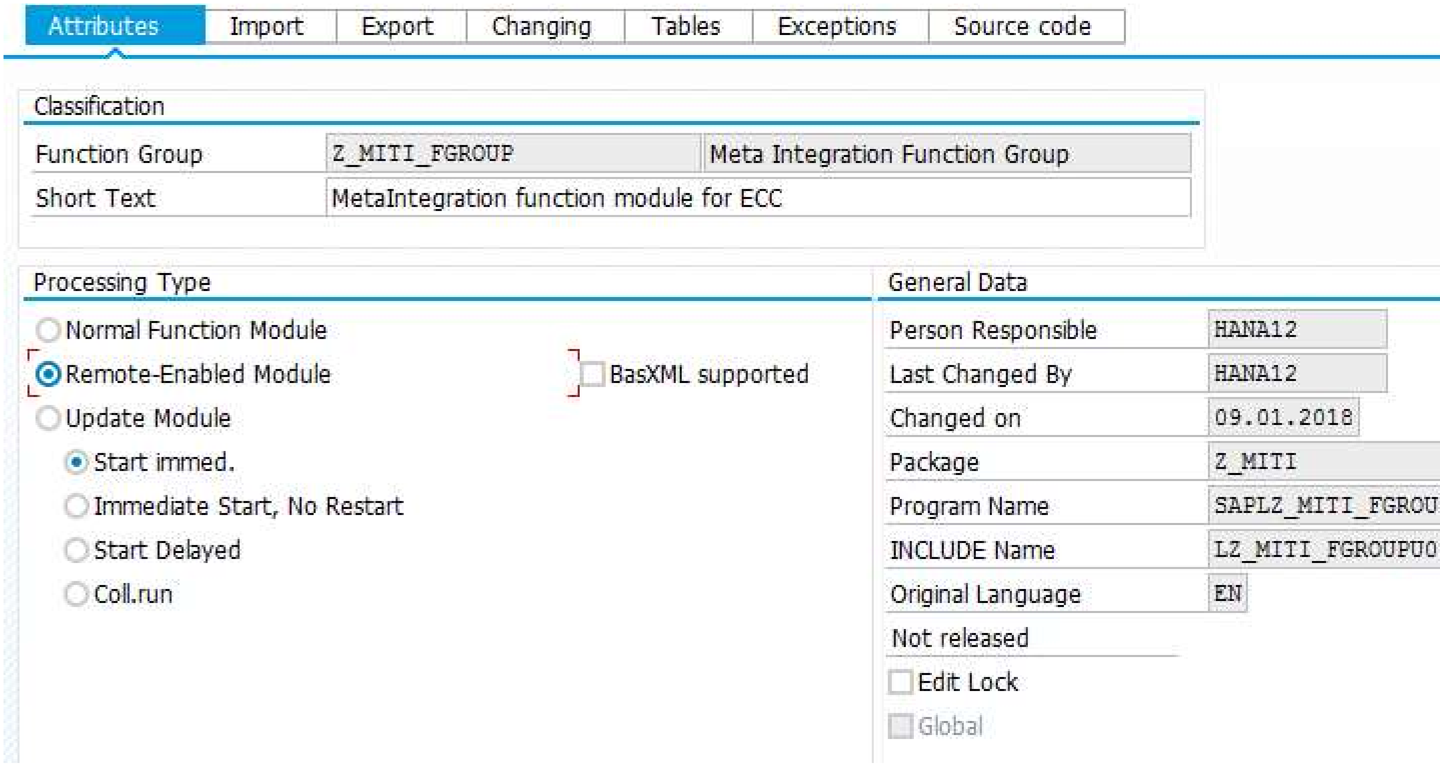
After the function group was created and selected, right mouse click on its name in repository browser and select "Create=>Function Module".

Enter "Z_MITI_BW_DOWNLOAD" into "Function Module" field and populate "Short text" input with proper description.

When the module has been created, specify the following information:

* navigate to the "Attributes" tab. Select

> Processing Type = Remote-Enabled Function Module.



Classification

Function Group: Z_MITI_FGROUP Meta Integration Function Group

Short Text: MetaIntegration function module for ECC

Processing Type

○ Normal Function Module

● Remote-Enabled Module ☐ BasXML supported

○ Update Module

- ● Start immed.
- ○ Immediate Start, No Restart
- ○ Start Delayed
- ○ Coll.run

General Data

Person Responsible: HANA12

Last Changed By: HANA12

Changed on: 09.01.2018

Package: Z_MITI

Program Name: SAPLZ_MITI_FGROU

INCLUDE Name: LZ_MITI_FGROUPU0

Original Language: EN

Not released

☐ Edit Lock

☐ Global

* navigate to the "Source code" tab. There are two ways how to deploy code for the function:

-from the main menu choose Utilities=>More Utilities=>Upload/Download=>Upload. and pick out Z_MITI_BW_DOWNLOAD.txt file.

-alternatively, open the file, copy its content and paste into "Source code" area.

* navigate to the "Import" tab and create the following parameters, pick "Pass Value" for all of them:

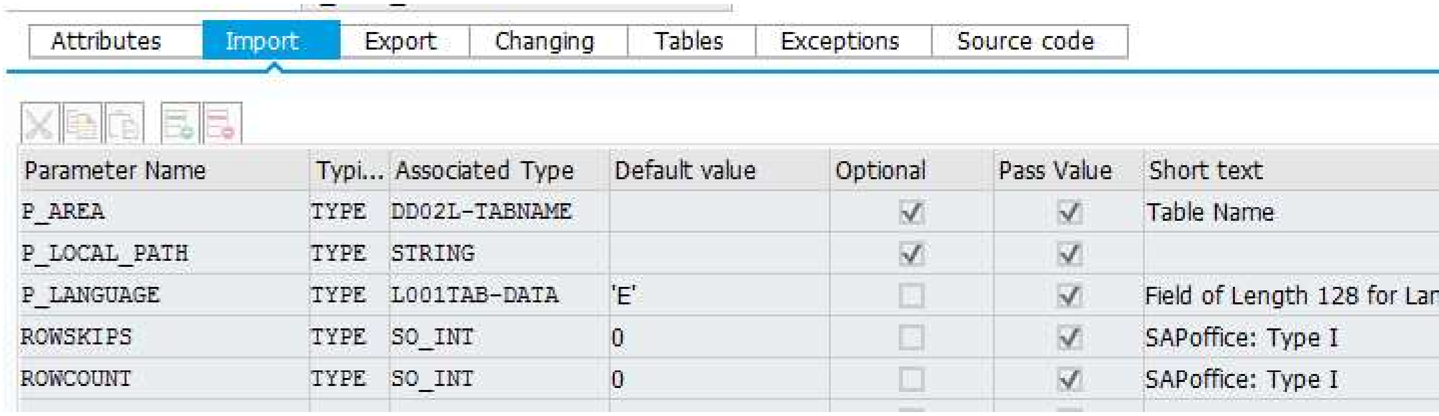
P_AREA TYPE DD02L-TABNAME (Optional = True)

*P_LOCAL_PATH TYPE STRING* (Optional = True)

*P_LANGUAGE TYPE L001TAB-DATA DEFAULT 'E'*

*ROWSKIPS TYPE SO_INT DEFAULT 0*

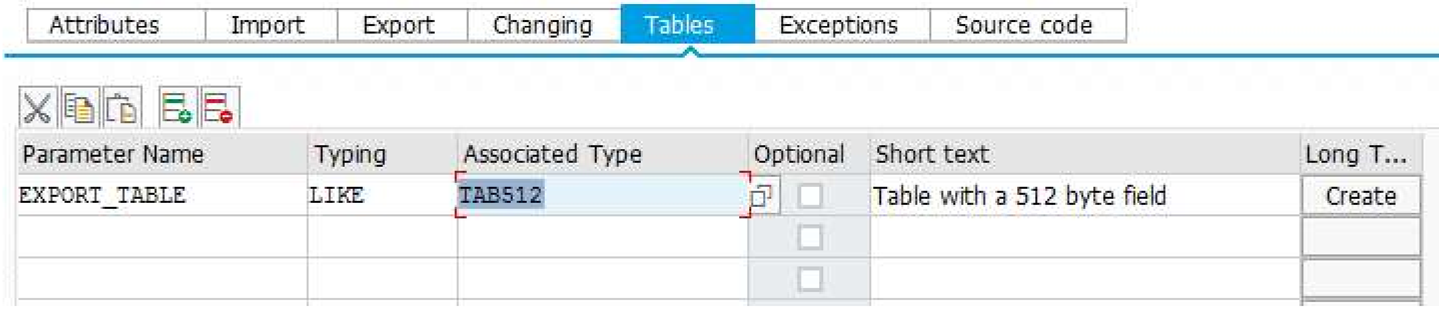
*ROWCOUNT TYPE SO_INT DEFAULT 0*



| Parameter Name | Typi... | Associated Type | Default value | Optional | Pass Value | Short text |
|---|---|---|---|---|---|---|
| P_AREA | TYPE | DD02L-TABNAME | | ☑ | ☑ | Table Name |
| P_LOCAL_PATH | TYPE | STRING | | ☑ | ☑ | |
| P_LANGUAGE | TYPE | L001TAB-DATA | 'E' | ☐ | ☑ | Field of Length 128 for Lar |
| ROWSKIPS | TYPE | SO_INT | 0 | ☐ | ☑ | SAPoffice: Type I |
| ROWCOUNT | TYPE | SO_INT | 0 | ☐ | ☑ | SAPoffice: Type I |

* Navigate to the “Tables” tab and define the following:

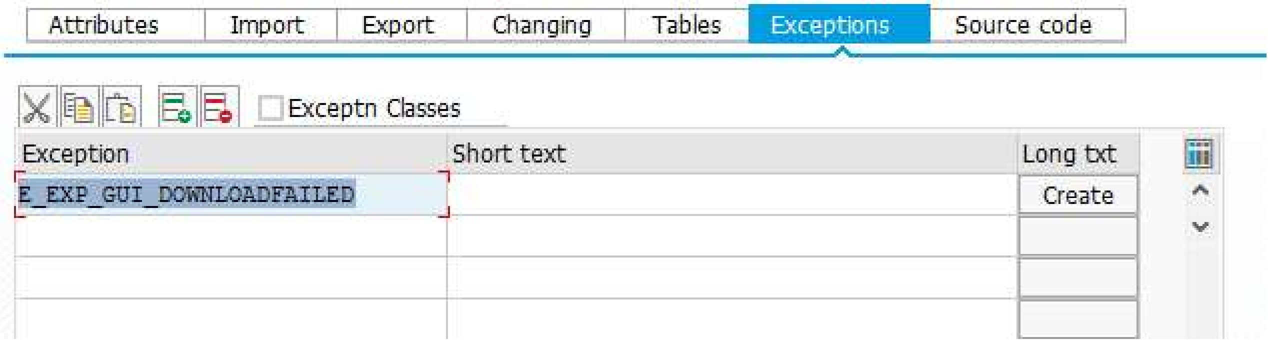
EXPORT_TABLE LIKE TAB512



| Parameter Name | Typing | Associated Type | Optional | Short text | Long T... |
|---|---|---|---|---|---|
| EXPORT_TABLE | LIKE | TAB512 | ☐ | Table with a 512 byte field | Create |
| | | | ☐ | | |
| | | | ☐ | | |

* Navigate to the "Exceptions" tab and define the following exception:

*E_EXP_GUI_DOWNLOADFAILED*



| Exception | Short text | Long txt |
|---|---|---|
| E_EXP_GUI_DOWNLOADFAILED | | Create |
| | | |
| | | |
| | | |

Save the function (press ctrl+S or choose Function Module=>Save in the main menu)

click "Activate" icon on the toolbar (ctrl+F3) and press "Continue" button in dialog window.

If prompted, you should select the generated includes to be activated along with the main function module.

# Editing the ABAP Script

## Overview

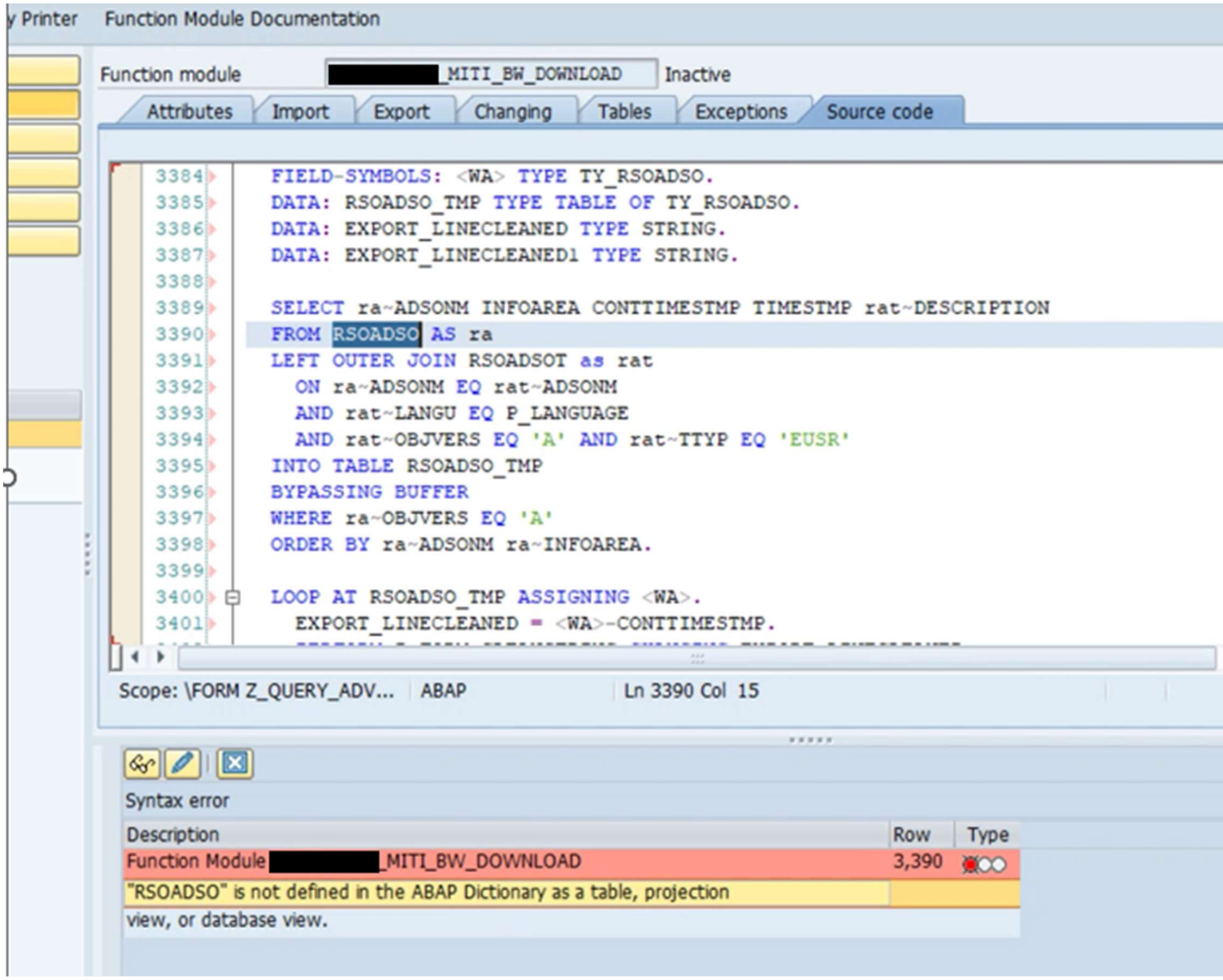
It is important to understand that SAP BW exists:

- In different product evolutions / editions from the original SAP BW, SAP BWonHANA, to the latest BW4HANA
- Each of these products/editions, with their own versions
- Each of these versions, with their service pack
- Each BW Product / Version / Service Pack adds, update and remove tables.
- Customers can also customize by deleting tables.

The ABAP script is then written to include all the tables required to collect all the supported metadata, even though the particular product/versions/service pack that you have installed may not support some of those tables. This fact is presented when you "Activate" the ABAP function module where they fail with syntax errors if a table does not exist.

## Example

If a certain FORM is encountering an error such as the below: *"RSOADSO" is not defined in the ABAP Dictionary as a table, projection, view or database view.*



Function module [redacted]MITI_BW_DOWNLOAD Inactive

```
3384  FIELD-SYMBOLS: <WA> TYPE TY_RSOADSO.
3385  DATA: RSOADSO_TMP TYPE TABLE OF TY_RSOADSO.
3386  DATA: EXPORT_LINECLEANED TYPE STRING.
3387  DATA: EXPORT_LINECLEANED1 TYPE STRING.
3388
3389  SELECT ra~ADSONM INFOAREA CONTTIMESTMP TIMESTMP rat~DESCRIPTION
3390  FROM RSOADSO AS ra
3391  LEFT OUTER JOIN RSOADSOT as rat
3392    ON ra~ADSONM EQ rat~ADSONM
3393    AND rat~LANGU EQ P_LANGUAGE
3394    AND rat~OBJVERS EQ 'A' AND rat~TTYP EQ 'EUSR'
3395  INTO TABLE RSOADSO_TMP
3396  BYPASSING BUFFER
3397  WHERE ra~OBJVERS EQ 'A'
3398  ORDER BY ra~ADSONM ra~INFOAREA.
3399
3400  LOOP AT RSOADSO_TMP ASSIGNING <WA>.
3401    EXPORT_LINECLEANED = <WA>-CONTTIMESTMP.
```

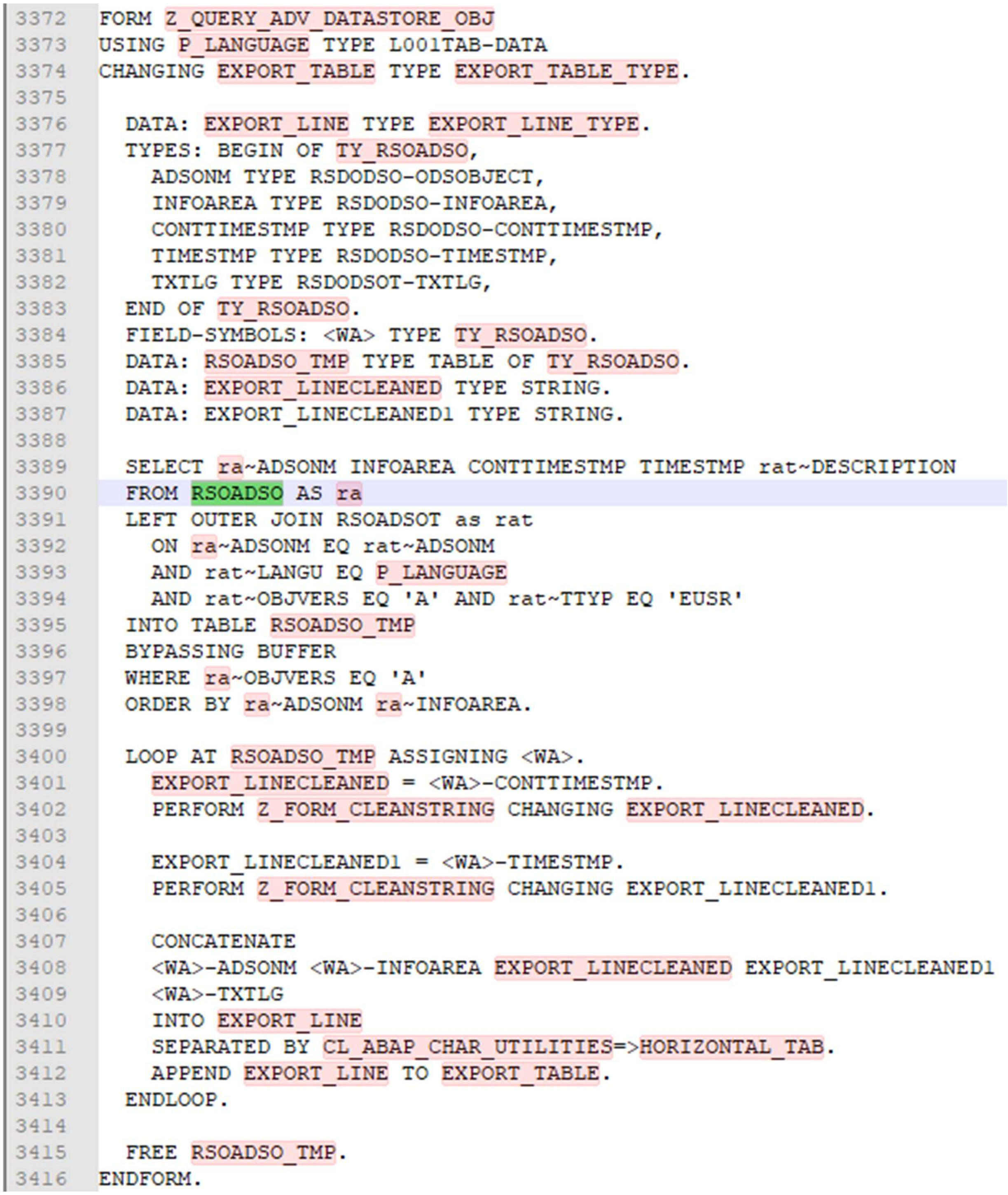
Scope: \FORM Z_QUERY_ADV... ABAP Ln 3390 Col 15

Syntax error

| Description | Row | Type |
|---|---|---|
| Function Module [redacted]MITI_BW_DOWNLOAD | 3,390 | ●○○ |
| "RSOADSO" is not defined in the ABAP Dictionary as a table, projection view, or database view. | | |

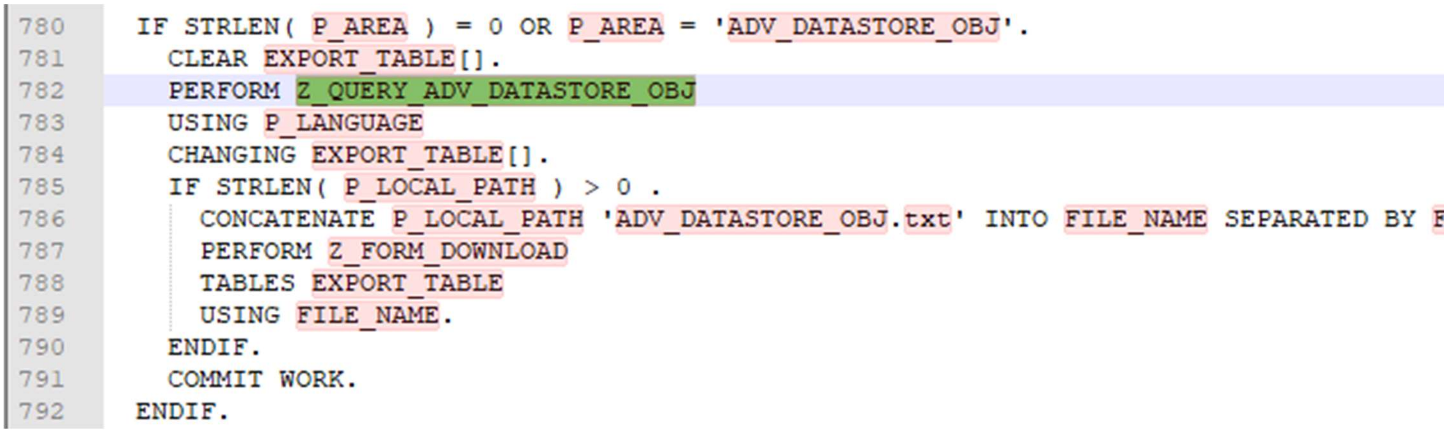
You may delete or comment out (by using * on each line) this FORM. Recommended method is to search the ABAP script file for the subroutine in question, go up several lines until you get to the FORM subroutine.

Table "RSOADSO" is used by the FORM "Z_QUERY_ADV_DATASTORE_OBJ", we will either comment out each line with an asterisk ( * ) or simply delete lines 3372 through line 3416.



```
FORM Z_QUERY_ADV_DATASTORE_OBJ
USING P_LANGUAGE TYPE L001TAB-DATA
CHANGING EXPORT_TABLE TYPE EXPORT_TABLE_TYPE.

  DATA: EXPORT_LINE TYPE EXPORT_LINE_TYPE.
  TYPES: BEGIN OF TY_RSOADSO,
    ADSONM TYPE RSDODSO-ODSOBJECT,
    INFOAREA TYPE RSDODSO-INFOAREA,
    CONTTIMESTMP TYPE RSDODSO-CONTTIMESTMP,
    TIMESTMP TYPE RSDODSO-TIMESTMP,
    TXTLG TYPE RSDODSOT-TXTLG,
  END OF TY_RSOADSO.
  FIELD-SYMBOLS: <WA> TYPE TY_RSOADSO.
  DATA: RSOADSO_TMP TYPE TABLE OF TY_RSOADSO.
  DATA: EXPORT_LINECLEANED TYPE STRING.
  DATA: EXPORT_LINECLEANED1 TYPE STRING.

  SELECT ra~ADSONM INFOAREA CONTTIMESTMP TIMESTMP rat~DESCRIPTION
  FROM RSOADSO AS ra
  LEFT OUTER JOIN RSOADSOT as rat
    ON ra~ADSONM EQ rat~ADSONM
    AND rat~LANGU EQ P_LANGUAGE
    AND rat~OBJVERS EQ 'A' AND rat~TTYP EQ 'EUSR'
  INTO TABLE RSOADSO_TMP
  BYPASSING BUFFER
  WHERE ra~OBJVERS EQ 'A'
  ORDER BY ra~ADSONM ra~INFOAREA.

  LOOP AT RSOADSO_TMP ASSIGNING <WA>.
    EXPORT_LINECLEANED = <WA>-CONTTIMESTMP.
    PERFORM Z_FORM_CLEANSTRING CHANGING EXPORT_LINECLEANED.

    EXPORT_LINECLEANED1 = <WA>-TIMESTMP.
    PERFORM Z_FORM_CLEANSTRING CHANGING EXPORT_LINECLEANED1.

    CONCATENATE
    <WA>-ADSONM <WA>-INFOAREA EXPORT_LINECLEANED EXPORT_LINECLEANED1
    <WA>-TXTLG
    INTO EXPORT_LINE
    SEPARATED BY CL_ABAP_CHAR_UTILITIES=>HORIZONTAL_TAB.
    APPEND EXPORT_LINE TO EXPORT_TABLE.
  ENDLOOP.

  FREE RSOADSO_TMP.
ENDFORM.
```

We will either comment out each line with an asterisk ( * ) or simply delete lines 780 through line 792 for the subroutine invocation line (starts with PERFORM Z_*name of FORM*)



```
  IF STRLEN( P_AREA ) = 0 OR P_AREA = 'ADV_DATASTORE_OBJ'.
    CLEAR EXPORT_TABLE[].
    PERFORM Z_QUERY_ADV_DATASTORE_OBJ
    USING P_LANGUAGE
    CHANGING EXPORT_TABLE[].
    IF STRLEN( P_LOCAL_PATH ) > 0 .
      CONCATENATE P_LOCAL_PATH 'ADV_DATASTORE_OBJ.txt' INTO FILE_NAME SEPARATED BY F
      PERFORM Z_FORM_DOWNLOAD
      TABLES EXPORT_TABLE
      USING FILE_NAME.
    ENDIF.
    COMMIT WORK.
  ENDIF.
```

Repeat the above two steps for each subroutine that is not applicable to your BW Server version.

## Testing the Function

When all the previous steps were completed open Z_MITI_BW_DOWNLOAD function module.

Choose "Function Module=>Test=>Test Function Module" from the main menu (or press F8).

Enter a path to the folder on the local file system into parameter P_LOCAL_PATH and press "Execute" icon on the toolbar (or press F8). Put the name of the area of interest into P_AREA field if some particular file with metadata must be downloaded or updated. When the function finishes working the folder which has been indicated in P_LOCAL_PATH parameter must contain a number of files with metadata inside. The names of files mimic areas which can be specified in P_AREA field.

The function will finish its execution and metadata will be downloaded much faster in case of launching it on the machine which has high-speed network connection with SAP BW server.