



**How-to Guide
SAP NetWeaver 2004s**

How To... implement the API for Real- Time InfoCubes

Version 1.00 – May 2006

**Applicable Releases:
SAP NetWeaver 2004s
(Business Information Management - Enterprise
Data Warehousing)**

© Copyright 2006 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation 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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, 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 in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

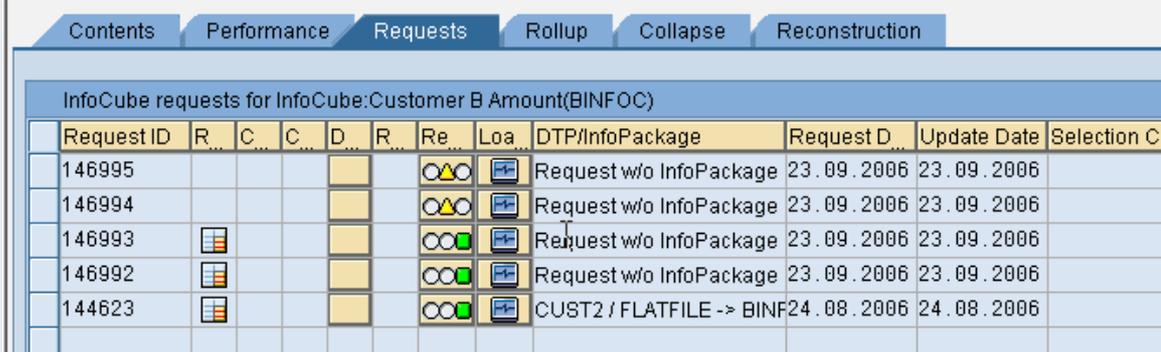
SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines /strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1 Scenario

You want to test your queries or authorization concept and therefore need to write test data to an InfoCube without using the Data Transfer Process (DTP). This is necessary since your development system is not connected to a productive source system and hence it is very difficult to use the existing data flow to load feasible test data.

Result:



The screenshot shows the SAP Request Administration interface with the 'Requests' tab selected. The table displays the following data:

Request ID	R...	C...	C...	D...	R...	Re...	Loa...	DTP/InfoPackage	Request D...	Update Date	Selection C...
146995						○△○	☒	Request w/o InfoPackage	23.09.2006	23.09.2006	
146994						○△○	☒	Request w/o InfoPackage	23.09.2006	23.09.2006	
146993	☒					○○○	☒	Request w/o InfoPackage	23.09.2006	23.09.2006	
146992	☒					○○○	☒	Request w/o InfoPackage	23.09.2006	23.09.2006	
144623	☒					○○○	☒	CUST2 / FLATFILE -> BIN	24.08.2006	24.08.2006	

2 Introduction

You have an ABAP routine to generate test data and you want to store the results directly in an InfoCube without using the normal data flow.

The Application Programming Interface (API) RSDRI_CUBE_WRITE_PACKAGE RFC can be used to write data per Request into a Real-Time InfoCube.

This interface has the following properties:

- Write interface
- Programming language is ABAP
- Mass data is supported
- Execution in background (no high frequency, for example, per minute)
- Programming proficiency is required
- Data is updated in an InfoCube

This API is a not released BAPI. Please note that for each call a new Request is created and therefore it is not recommended to write data with high frequency to an InfoCube. This can result in heavy performance problems in the data load process, and it will decrease the Request Administration Performance.

Note: This program is not supported by SAP.

3 The Step By Step Solution

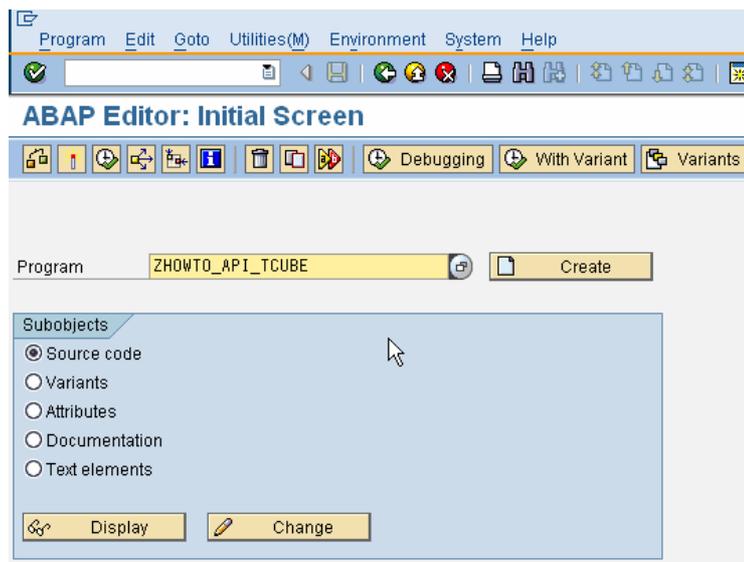
The attached program writes generated test data to an existing Real-Time InfoCube based on master data information.

3.1 Create a sample program

1. To create a program, call transaction SE38.

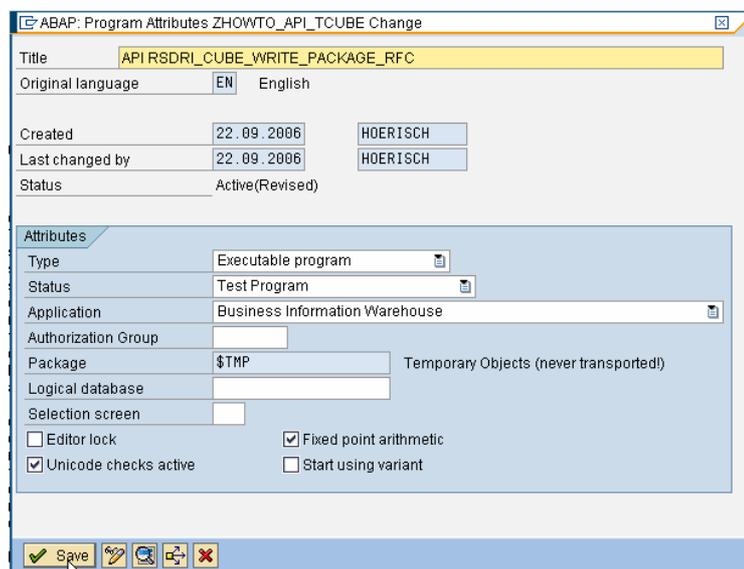
Create a program with for example the name

ZHOWTO_API_TCUBE



2. The program attributes are for example the following ones:

Type: Executable program
Status: Test Program
Application: Business Information Warehouse

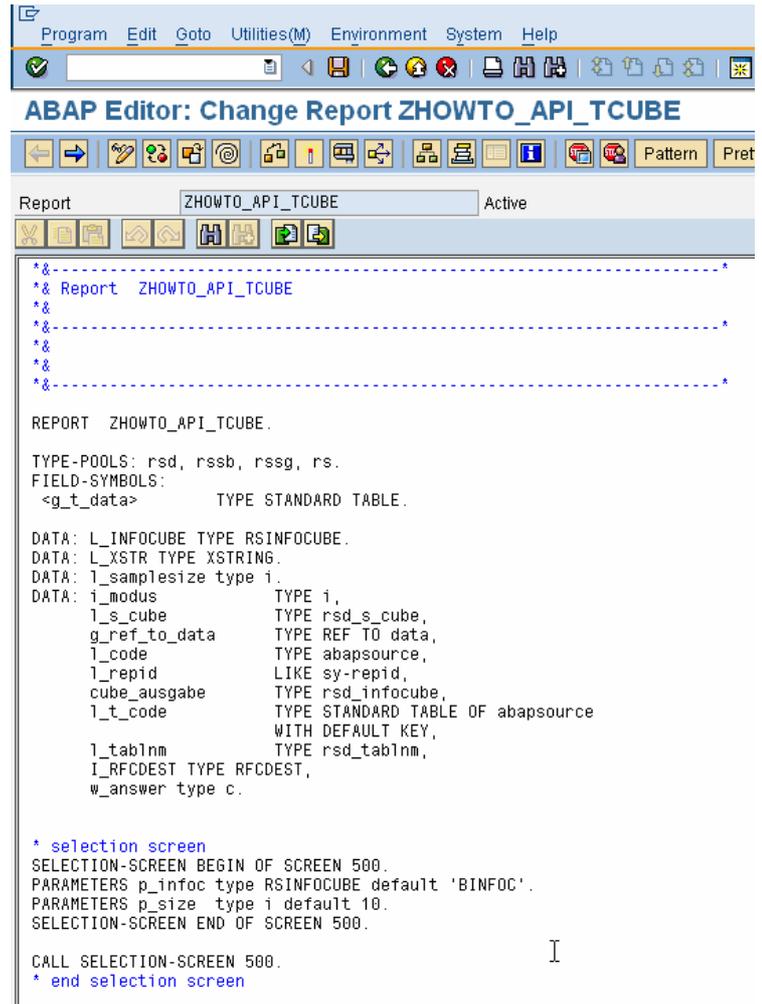


- Copy and paste the attached coding to the program

See [Appendix \(Coding for the program\)](#)

To save the data, choose Save 

To activate the program, choose **Activate** 



```

*&-----*
*& Report  ZHOWTO_API_TCUBE
*&
*&-----*
*&
*&-----*

REPORT  ZHOWTO_API_TCUBE.

TYPE-POOLS: rsd, rssb, rssg, rs.
FIELD-SYMBOLS:
<g_t_data>          TYPE STANDARD TABLE.

DATA: L_INFOCUBE TYPE RSINFOCUBE.
DATA: L_XSTR TYPE XSTRING.
DATA: l_samplesize type i.
DATA: i_modus          TYPE i,
      l_s_cube         TYPE rsd_s_cube,
      g_ref_to_data    TYPE REF TO data,
      l_code           TYPE abapsource,
      l_repid         LIKE sy-repid,
      cube_ausgabe     TYPE rsd_infocube,
      l_t_code        TYPE STANDARD TABLE OF abapsource
                      WITH DEFAULT KEY,
      l_tablnm        TYPE rsd_tablnm,
      I RFCDEST TYPE RFCDEST,
      w_answer type c.

* selection screen
SELECTION-SCREEN BEGIN OF SCREEN 500.
PARAMETERS p_infoc type RSINFOCUBE default 'BINFOC'.
PARAMETERS p_size  type i default 10.
SELECTION-SCREEN END OF SCREEN 500.

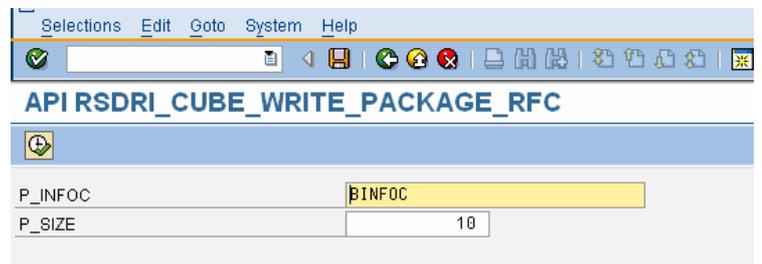
CALL SELECTION-SCREEN 500.
* end selection screen

```

- Return to the entry screen of transaction SE38. To execute the program, choose Direct Processing 

- Fill the parameters

P_INFOC (technical InfoCube name)
P_SIZE (The number records of the Request)



API RSDRI_CUBE_WRITE_PACKAGE RFC	
P_INFOC	BINFOC
P_SIZE	10

- To execute the program, choose **Execute** 

3.2 API description

1. IMPORTING parameters are

I_INFOCUBE

The technical name of the InfoCube

I_CURR_CONVERSION

If this flag is checked, the currency unit will be converted to the internal database format.

I_RFCDATA_UC

Is the input data in Unicode conform format (the generated sample data in this case)

2. EXPORTING parameters are

E_REQUID

The generated Request number

E_RECORDS:

The number of the inserted records in the InfoCube.

3. Tables parameters (can be modified in the API):

I_T_RFCDATA

The input data In non-Unicode conform format

```
*****
* API call
* Write generated data to InfoCube *
*****
CALL FUNCTION 'RSDRI_CUBE_WRITE_PACKAGE_RFC'
  DESTINATION I_RFCDDEST
  EXPORTING
    I_INFOCUBE           = I_INFOCUBE
    I_CURR_CONVERSION   = ' '
    I_RFCDATA_UC        = L_XSTR
  TABLES
    I_T_RFCDATA = I_T_RFCDATA
  EXCEPTIONS
    COMMUNICATION_FAILURE = 1
    SYSTEM_FAILURE        = 2
    NOT_TRANSACTIONAL     = 3.
```

4 Appendix (Coding for the program)

```
*&-----*
*& Report  ZHOWTO_API_TCUBE
*&
*&-----*
*&
*&
*&-----*

REPORT  ZHOWTO_API_TCUBE.

*****
* Data Types *
*****
TYPE-POOLS: rsd, rrsb, rrs, rs.
FIELD-SYMBOLS:
  <g_t_data>          TYPE STANDARD TABLE.

DATA: L_INFOCUBE TYPE RSINFOCUBE.
DATA: L_XSTR TYPE XSTRING.
DATA: l_samplesize type i.
DATA: i_modus      TYPE i,
      l_s_cube     TYPE rsd_s_cube,
      g_ref_to_data TYPE REF TO data,
      l_code       TYPE abapsource,
      l_repid      LIKE sy-repid,
      cube_ausgabe TYPE rsd_infocube,
      l_t_code     TYPE STANDARD TABLE OF abapsource
                    WITH DEFAULT KEY,
      l_tablnm     TYPE rsd_tablnm,
      I_RFCDEST TYPE RFCDEST,
      w_answer type c.

*****
* Selection Screen *
*****
SELECTION-SCREEN BEGIN OF SCREEN 500.
PARAMETERS p_infoc type RSINFOCUBE default 'BINFOC'.
PARAMETERS p_size type i default 10.
SELECTION-SCREEN END OF SCREEN 500.

CALL SELECTION-SCREEN 500.
* end selection screen

check p_infoc is not initial.
check p_size is not initial.

CALL FUNCTION 'POPUP_TO_CONFIRM'
  EXPORTING
    TEXT_QUESTION      = 'Are you certain ... ?'
    TEXT_BUTTON_1     = 'YES' (001)
    TEXT_BUTTON_2     = 'NO' (002)
  IMPORTING
    ANSWER             = w_answer
  EXCEPTIONS
    TEXT_NOT_FOUND    = 1
    OTHERS            = 2
.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
```

```

if w_answer = '2' or w_answer = 'A'.
  exit.
endif.
move p_infoc to l_infocube.
move p_size to l_samplesize.

*l_infocube = 'BINFOC'.
*l_samplesize = 10.

*****
* Get MetaData Information of the InfoCube *
*****

* get information about the infocube
CALL FUNCTION 'RSD_CUBE_GET'
  EXPORTING
    i_infocube      = l_infocube
  IMPORTING
    e_s_cube       = l_s_cube
  EXCEPTIONS
    infocube_not_found = 1
    illegal_input     = 2
    OTHERS            = 4.
IF sy-subrc <> 0.
  MESSAGE ID 'BRAIN' TYPE 'I' NUMBER 330
    WITH l_infocube.
  CALL SCREEN 100.
ENDIF.

*****
* AUTHORITY CHECK *
*****
CALL FUNCTION 'RSSB_AUTHORITY_ADMWB_INFOCUBE'
  EXPORTING
*     I_INFOAREA      =
    i_actvt          = rssb_c_auth_actvt-delete
    i_infocube       = l_infocube
    i_icubeobj       = rssb_c_auth_icubeobj-data
*     I_TRY_DISPLAY  =
*   IMPORTING
*     E_DISPLAY_ONLY =
  EXCEPTIONS
    user_not_authorized = 1
    OTHERS              = 2.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

*****
* generate Data Type for the InfoCube *
*****
* view all infoobjects
l_tablnm = l_s_cube-viewtiobjnm2.

PERFORM get_report_name
  CHANGING l_repid.

CONCATENATE 'REPORT' l_repid '.' INTO l_code SEPARATED BY space.
APPEND l_code TO l_t_code.
CONCATENATE 'TYPES: L_S_DATA TYPE' l_tablnm ','
  INTO l_code SEPARATED BY space.
APPEND l_code TO l_t_code.
APPEND ' L_T_DATA TYPE STANDARD TABLE OF L_S_DATA' TO l_t_code.
APPEND ' WITH DEFAULT KEY.' TO l_t_code.
APPEND 'FORM CREATE_TABLE_TYPES' TO l_t_code.

```

```
APPEND '    CHANGING E_REF_TO_CHAVLTAB TYPE REF TO DATA.' TO l_t_code.
APPEND '  CREATE DATA E_REF_TO_CHAVLTAB TYPE L_T_DATA.' TO l_t_code.
APPEND 'ENDFORM.' TO l_t_code.
```

```
INSERT REPORT l_repid FROM l_t_code.
```

```
PERFORM create_table_types IN PROGRAM (l_repid)
    CHANGING g_ref_to_data.
```

```
* get the columns-headers
ASSIGN g_ref_to_data->* TO <g_t_data>.
DELETE REPORT l_repid.
```

```
CLEAR l_t_code.
```

```
*****
* create random key figures          *
* and characteristics                *
*****
* create random key figures and characteristics
CALL FUNCTION 'RSSAMPLE_CUBE_SAMPLE_CREATE'
    EXPORTING
        i_infocube      = l_infocube
        i_samplesize    = l_samplesize
        i_use_masterdata = 'X'
        i_random_seed   = 0
        i_modus         = 3
    IMPORTING
        e_t_data        = <g_t_data>
    EXCEPTIONS
        illegal_input   = 1
        inherited_error = 2
        OTHERS          = 3.
IF sy-subrc <> 0.
    MESSAGE ID 'BRAIN' TYPE 'E' NUMBER '332'.
```

```
ENDIF.
```

```
* Send the data using compressed XSTRING via RFC
EXPORT RSDRI = <g_t_data> TO DATA BUFFER L_XSTR
    COMPRESSION ON.
FREE <g_t_data>.
```

```
DATA: I_T_RFCDATA TYPE RSDRI_T_RFCDATA.
```

```
*****
* API call          *
* Write generated data to InfoCube *
*****
CALL FUNCTION 'RSDRI_CUBE_WRITE_PACKAGE RFC'
    DESTINATION I_RFCDST
    EXPORTING
        I_INFOCUBE      = l_INFOCUBE
        I_CURR_CONVERSION = ' '
        I_RFCDATA_UC    = L_XSTR
    TABLES
        I_T_RFCDATA = I_T_RFCDATA
    EXCEPTIONS
        COMMUNICATION_FAILURE = 1
        SYSTEM_FAILURE        = 2
        NOT_TRANSACTIONAL     = 3.
```

```
*-----*
* FORM get_report_name *
*-----*
```

```

*-----*
*          *
*-----*
* --> E_REPID
*-----*
FORM get_report_name
  CHANGING e_repid LIKE sy-repid.

DATA:
  l_uniqueid TYPE rsgg_unic25.

*-generate a Unique-ID -----*
CALL FUNCTION 'RSS_SYSTEM_GET_UNIQUE_ID'
  IMPORTING
    e_unic25 = l_uniqueid.

  CONCATENATE 'GP' l_uniqueid INTO e_repid.
ENDFORM.                "get_report_name

```

<http://www.sdn.sap.com/irj/sdn/howtoguides>