

Create Monitor Entries in BI Using Start Routine



Applies to:

SAP BW 3.5 and above system

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

Summary

To describe the process followed to create monitor entries using start routine

Author: R. Prem Kumar

Company: SAP

Created on: 23 May 2009

Author Bio



The author works for SAP. His expertise includes SAP BI and ABAP

Table of Contents

Introduction	3
Goal.....	3
Graphical view	3
Master Data Object	3
Data Source	3
Required Logic.....	3
Start Routine	4
Class Builder: Se24.....	4
Method	5
Code.....	6
Declarations Inside Local Types	7
Result	8
Copyright.....	10

Introduction

To describe the process followed to create monitor entries using start routine

Goal

This document describes how to create monitor entries for a particular logic on the data to be loaded using the start routine. The code for creating the monitor entries is written in a method for a class and this method return would be called in the start routine

Graphical view

Master Data Object

In this document we shall use the info-object **0PROJECT** as master data object

Data Source

The data source **0PROJECT_ATTR** would be used as an example in this document

It is assumed that this data source is already replicated and is available inside BI and is also connected to **0PROJECT**

Project Definition	0PROJECT	=	Change
Project Definition (Attribute)	ATTRIBUTES 0PRO...		Manage
from Project definition	0PROJECT_ATTR		Change
Project definition	0PROJECT_ATTR		Change
Delta Load 0proje	ZPAK_4CNQ9V1XR...		Schedule
Full load test	ZPAK_4CQIZUG5K...		Schedule
Init Load 0project	ZPAK_4CNQ9DCE1...		Schedule

This data source is used to fetch the attributes of a project

Required Logic

The screen shot below is the data in the PSA. The monitor entries are to be created for a project when the year in the start date and the year in the end are not the same

Project def. (ext.)	Project Definition	Changed on	Person res	Applicant	Company Co	B	Controllin	P	Object cur	Plant	C	Created on	BW Status:	Project Pr	Start Date	End Date
M.DK.03T	M.DK.03T	16.12.2002	50006662	50006662	DK02		DK01		DKK	DK98		12.12.2002	2	ZM00045	16.12.2002	19.12.2003
M.FI.03C	M.FI.03C	15.01.2004	50006692	50006692	FI02		FI01		EUR	FI98		11.12.2002	2	ZM00058	16.12.2002	16.01.2004
M.FI.03F	M.FI.03F	15.12.2002	50006692	50006692	FI02		FI01		EUR	FI98		11.12.2002	2	ZM00058	16.12.2002	19.12.2003
M.FI.03M	M.FI.03M	15.12.2002	50006689	50006689	FI02		FI01		EUR	FI98		11.12.2002	2	ZM00058	16.12.2002	19.12.2003
M.FI.03T	M.FI.03T	15.12.2002	50006692	50006692	FI02		FI01		EUR	FI98		11.12.2002	1	ZM00058	16.12.2002	19.12.2003

Start Routine

The logic to achieve the above mentioned requirement is written in the start routine. However, it is not written directly in the routine, instead it is written in a method and is called in the start routine

This is particularly useful when changes are required to be made to the code. The code can directly be changed in the method leaving the start routine untouched

```

$$$ end of global - insert your declaration only before this line $$$
FORM STARTROUTINE
  USING    G_S_MINFO TYPE RSSM_S_MINFO
  CHANGING DATAPAK type TAB_TRANSTRU
           G_T_ERRORLOG TYPE rssm_t_errorlog_int
           ABORT LIKE SY-SUBRC. "set ABORT <> 0 to cancel datapackage
  +$$$ begin of routine - insert your code only below this line +$$
  + DATA: l_s_datapak line type TRANSFER_STRUCTURE,
  +        l_s_errorlog TYPE rssm_s_errorlog_int.

  CALL METHOD ZCL_TR_RR=>START_0PROJECT
  EXPORTING
    G_S_MINFO    = G_S_MINFO
  CHANGING
    DATAPAK      = DATAPAK
    G_T_ERRORLOG = G_T_ERRORLOG.

```

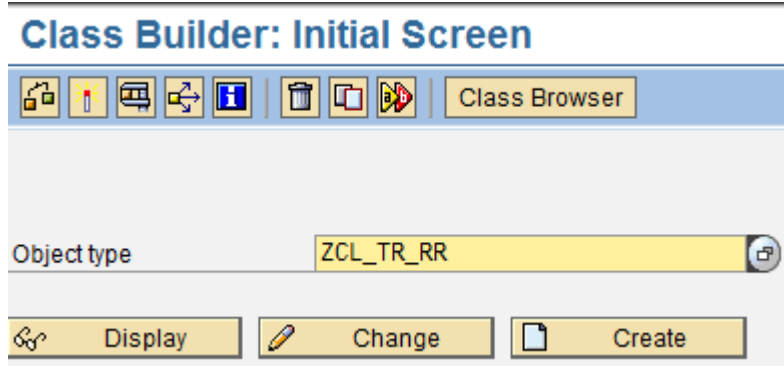
The above screen shows the method being called in the start routine.

The name of the method is **START_0PROJECT**

This method belongs to the class **ZCL_TR_RR**

Class Builder: Se24

The T-code SE24 is used to create/change/display a class



Method

A method **START_0PROJECT** is created in this class

Class Builder: Display Class ZCL_TR_RR

Method	Level	Visi	M...	Description
COMPUTE_FISCPER13	Static	Publ		Derive 0FISCPER - PCA
COMPUTE_ZPRODDTL	Static	Publ		Derive IOBJ ZPRODDTL
START_0PROJECT	Static	Publ		Start routine for 0PROJECT master Load
START_0CO_OM_WBS_2_ZRREUC	Static	Publ		Start routine for 0CO_OM_WBS_2 to ZRREUC11
COMPUTE_ZICCUAP_IOBJ	Static	Publ		Derive ZUNIT, ZICCUSTA, ZCONTREU, ZMARKET and ZEUCHAN

The parameters of the method should be created according to the inputs being received from the start routine and the required output

Here, in the start routine the using parameter **G_S_MINFO** gives the technical details and the changing parameter **DATAPAK** will contain the source data and **G_T_ERRORLOG** is the internal table to be used to create monitor entries

Parameters are to be created for the method which should be similar to the above mentioned parameters, in this example the method parameters are also created with the same names as the parameters in the start routine

Class Builder: Display Class ZCL_TR_RR

Parameter	Type	Pa	O...	Typing M...	Associated Type	Default value	Description
G_S_MINFO	Importin	<input type="checkbox"/>	<input type="checkbox"/>	Type	RSSM_S_MINFO		
DATAPAK	Changin	<input type="checkbox"/>	<input type="checkbox"/>	Type	TABLE		
G_T_ERRORLOG	Changin	<input type="checkbox"/>	<input type="checkbox"/>	Type	RSSM_T_ERRORLOG_I		Logs for Incorrect Records (Intern...

Below is the code written inside the method based on the logic mentioned at the beginning of this document, to create monitor entries

Code

```

METHOD START_0PROJECT.

DATA: IT_DATE_SOURCE TYPE TABLE OF LTY_DATE_SOURCE,
      LWA_SLINE TYPE LTY_DATE_SOURCE,
      WA_MON LIKE LINE OF G_T_ERRORLOG.

IF DATAPAK[] IS INITIAL.
  RETURN.
ENDIF.

IT_DATE_SOURCE[] = DATAPAK[].

LOOP AT IT_DATE_SOURCE[] INTO LWA_SLINE.

DATA: ST_GJAHR TYPE BDATJ,
      EN_GJAHR TYPE BDATJ.

CLEAR:ST_GJAHR.
IF LWA_SLINE-ZZPLFAZ IS NOT INITIAL.

  CALL FUNCTION 'DATE_TO_PERIOD_CONVERT'
    EXPORTING
      I_DATE           = LWA_SLINE-ZZPLFAZ
*   I_MONMIT          = 00
      I_PERIV          = 'A1'
    IMPORTING
*   E_BUPER           =
      E_GJAHR          = ST_GJAHR
* EXCEPTIONS
*   INPUT_FALSE       = 1
*   T009_NOTFOUND     = 2
*   T009B_NOTFOUND    = 3
*   OTHERS             = 4
    .

  IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.
ENDIF.

CLEAR:EN_GJAHR.
IF LWA_SLINE-ZZPLSEZ IS NOT INITIAL.

  CALL FUNCTION 'DATE_TO_PERIOD_CONVERT'
    EXPORTING
      I_DATE           = LWA_SLINE-ZZPLSEZ
*   I_MONMIT          = 00
      I_PERIV          = 'A1'
    IMPORTING
*   E_BUPER           =
      E_GJAHR          = EN_GJAHR
* EXCEPTIONS

```

```

* INPUT_FALSE           = 1
* T009_NOTFOUND        = 2
* T009B_NOTFOUND       = 3
* OTHERS                = 4
.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
ENDIF.

IF ST_GJAHR NE EN_GJAHR.
*start year for <@PROJECT> is not equal end year.
WA_MON-RECORD = LWA_SLINE-RECORD.
WA_MON-MSGID = 'RSM'.
WA_MON-MSGTY = 'I'.
WA_MON-MSGNO = '799'.
CONCATENATE 'start year for ' LWA_SLINE-PSPID INTO WA_MON-MSGV1.
WA_MON-MSGV2 = 'is not equal to end year'.
APPEND WA_MON TO G_T_ERRORLOG.
ENDIF.
ENDLOOP.

ENDMETHOD.

```

Declarations Inside Local Types

```

BEGIN OF LTY_DATE_SOURCE,
  REQUEST(30) TYPE C,
  DATAPAKID(6) TYPE N,
  PARTNO TYPE INT4,
* Record number to be filled in case of adding row(s)
* to enable 'error handling'
  RECORD TYPE RSARECORD,
* InfoObject @PROJECT_EX: CHAR - 000024
  PROEX(000024) TYPE C,
* InfoObject @PROJECT: CHAR - 000024
  PSPID(000024) TYPE C,
* InfoObject @CH_ON: DATS - 000008
  AEDAT TYPE D,
* InfoObject @PS_RESPNO: NUMC - 000008
  VERNR(000008) TYPE N,
* InfoObject @PS_APPLNO: NUMC - 000008
  ASTNR(000008) TYPE N,
* InfoObject @COMP_CODE: CHAR - 000004
  VBUKR(000004) TYPE C,
* InfoObject @BUS_AREA: CHAR - 000004
  VGSBR(000004) TYPE C,
* InfoObject @CO_AREA: CHAR - 000004
  VKOKR(000004) TYPE C,
* InfoObject @PROFIT_CTR: CHAR - 000010
  PRCTR(000010) TYPE C,
* InfoObject @OBJ_CURR: CUKY - 000005
  PWHIE(000005) TYPE C,

```

```

*   InfoObject 0PLANT: CHAR - 000004
    WERKS(000004) TYPE C,
*   InfoObject 0COSTCENTER: CHAR - 000010
    KOSTL(000010) TYPE C,
*   InfoObject 0CREATEDON: DATS - 000008
    ERDAT TYPE D,
*   InfoObject 0STATUSSYS0: NUMC - 000002
    BWSTSYS0(000002) TYPE N,
*   InfoObject ZPROFL: CHAR - 000007
    ZZPROFL(000007) TYPE C,
*   InfoObject ZSTARTDT: DATS - 000008
    ZZPLFAZ TYPE D,
*   InfoObject ZENDDATE: DATS - 000008
    ZZPLSEZ TYPE D,
END   OF LTY_DATE_SOURCE.

```

Result

When the info-package is executed the data is loaded to the PSA, and when the same data is loaded from PSA to DSO it goes through the start routine. Since the method is used within this routine, it gets executed and based on the logic written inside the method the monitor entries are created which can be seen from the data load monitor

Note: In this example the messages to be created in the monitor entries should be of the format “**start year for <PROJECT> is not equal to end year**”

The screenshot displays the SAP Data Load Monitor interface. It shows a tree view of messages under the 'Details' tab. The overall status is 'Everything OK'. The messages are as follows:

- Overall status: Everything OK
 - Requests (messages): Everything OK
 - Extraction (messages): Everything OK
 - Transfer (IDocs and TRFC): Everything OK
 - Processing (data packet): Everything OK
 - Data Package 1 (13227 Records) : Everything OK
 - Update PSA (13227 Records posted) : No errors
 - Transfer Rules (13227 -> 13227 Records) : No errors
 - Master data received. Processing being started.
 - Data records for package 1 selected in PSA - 0 error(s)
 - Record 1 :Error: start year forPBE032002 is not equal to end year
 - Record 2 :Error: start year forPBE042002 is not equal to end year
 - Record 3 :Error: start year forM.NL.MIPASEN is not equal to end year
 - Record 4 :Error: start year forCBE31997024 is not equal to end year
 - Record 5 :Error: start year forT.BE.02.CHS is not equal to end year
 - Record 6 :Error: start year forT.BE.02.COF is not equal to end year
 - Record 7 :Error: start year forT.BE.02.CON is not equal to end year
 - Record 8 :Error: start year forT.BE.02.GRC is not equal to end year
 - Record 9 :Error: start year forT.NL.02.CHS is not equal to end year
 - Record 10 :Error: start year forT.NL.02.COF is not equal to end year
 - ...more messages in the PSA display (double-click on LED)
 - Transfer 13227 data records in communication structure

More messages can be seen in the PSA

Status	DataPacket	Data Rec.	Project de	Project De	Changed on	Person res	Applicant	Company Co
🔍	1	13	CBE11999001	CBE11999001	21.04.2005	50032347	50000945	BE03
🔍	1	14	CBE12000001	CBE12000001	16.03.2002	50000945	50000946	BE03
🔍	1	44	CBE12000002	CBE12000002	16.03.2002	50000945	50000946	BE03
🔍	1	99	CBE12001001	CBE12001001	21.04.2004	50032347	50000945	BE03
🔍	1	103	CBE12001002	CBE12001002	16.03.2002	50000945	50000946	BE03
🔍	1	105	CBE12001003	CBE12001003	16.03.2002	50000945	50000946	BE03
🔍	1	109	CBE12001004	CBE12001004	17.03.2002	50000944	50000946	BE03
🔍	1	119	CBE12001050	CBE12001050	19.02.2003	50000944	50000946	BE03
🔍	1	120	CBE12001051	CBE12001051	16.03.2002	50000945	50000946	BE03
🔍	1	121	CBE12001052	CBE12001052	16.03.2002	50000945	50000946	BE03
🔍	1	122	CBE12001053	CBE12001053	16.03.2002	50000944	50000946	BE03
🔍	1	123	CBE12002001	CBE12002001		1	1	BE03
🔍	1	124	CBE12002002	CBE12002002	16.03.2002	50000945	50000946	BE03

Status	DataPacket	Data Rec.	Project de	Project De	Changed on	Person res	Applicant	Company Co	Business a	Controllin										
🔍	1	13	CBE11999001	CBE11999001	21.04.2005	50032347	50000945	BE03		BE01										
🔍	1	14	CBE12000001	CBE12000001	16.03.2002	50000945	50000946	BE03		BE01										
🔍	1	44	CBE12000002	CBE12000002	16.03.2002	50000945	50000946	BE03		BE01										
🔍	1	99	CBE12001001	CBE12001001	21.04.2004	50032347	50000945	BE03		BE01										
🔍	1	103	CBE12001002	CBE12001002	16.03.2002	50000945	50000946	BE03		BE01										
🔍	1	Messages for data record 13																		
🔍	1	<table border="1"> <thead> <tr> <th>Type</th> <th>Message Text</th> <th>ID</th> <th>No.</th> <th>LTxt</th> </tr> </thead> <tbody> <tr> <td>🟢</td> <td>Error: start year for CBE11999001 is not equal to end year</td> <td>RSM</td> <td>799</td> <td></td> </tr> </tbody> </table>									Type	Message Text	ID	No.	LTxt	🟢	Error: start year for CBE11999001 is not equal to end year	RSM	799	
Type	Message Text	ID	No.	LTxt																
🟢	Error: start year for CBE11999001 is not equal to end year	RSM	799																	
🔍	1	124	CBE12002002	CBE12002002	16.03.2002	50000945	50000946	BE03		BE01										
🔍	1	125	CBE12002003	CBE12002003		1	1	BE03		BE01										

Copyright

© Copyright 2009 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.