

How to do Real-Time Data Acquisition through Web Service (Push Method)



Applies to:

BI NetWeaver 2004s. For more information, visit the [EDW homepage](#).

Summary

This paper describes How to do Real-Time Data Acquisition through Web Service (Push Method)

Author: Deshmukh Bharti

Company: Accenture Services Private Ltd

Created on: 30th of November 2010

Author Bio

Deshmukh Bharti is working as Senior SAP BI Consultant in Accenture Services Private Ltd and having extensive experience in implementation of BI projects.

Table of Contents

Purpose.....	3
Process Flow	4
Constraints.....	4
Two Ways Data Transfer Process	5
Using a service API.....	5
Using a Web service	5
Data Acquisition through Web Service	6
Prerequisites.....	6
Create a Data Source in BI	6
Create Interface between Data Source and Web Service (SAP XI)	8
Create an Info Package for Demon with Push Scenario	13
Create Data Transfer Process with Pull Scenario and Data Target	17
Create Process Chain to Automate the Jobs.....	18
Related Content.....	19
Copyright.....	20

Purpose

Real-time data acquisition supports tactical decision-making. It also supports operational reporting by allowing you to send data to the delta queue or PSA table in real-time. We then use a daemon to transfer DataStore objects to the operational DataStore layer at frequent regular intervals. The data is stored persistently in BI.

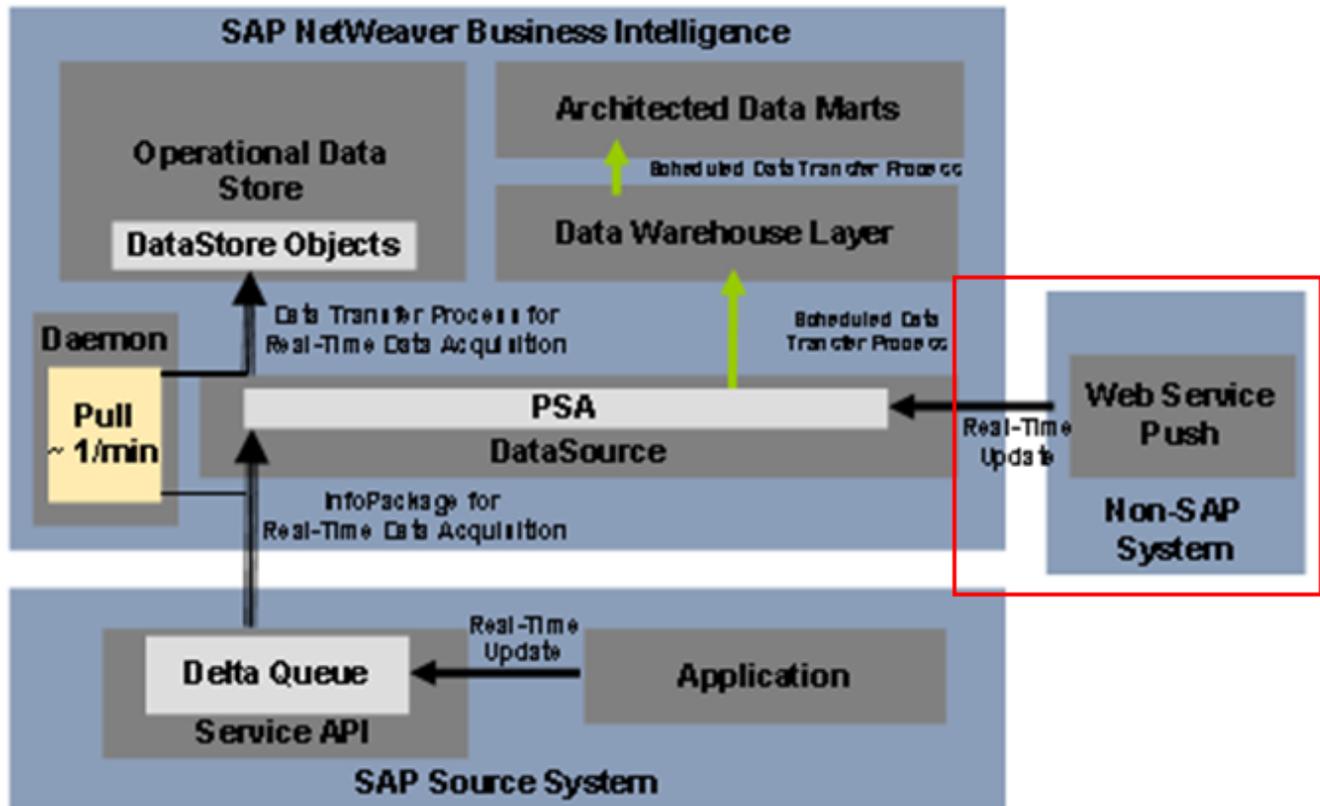
We recommend that you use real-time data acquisition if you want to transfer data to BI at shorter intervals (every minute) than scheduled data transfers and you need up-to-date data to be regularly available in reporting (several times a day, at least).

The following overview displays the differences between standard data acquisition using scheduled data requests and real-time data acquisition:

	Standard Data Acquisition	Real-Time Data Acquisition
Goal	Strategic Decision-Making (long-term planning)	Tactical Decision-Making (daily decisions)
Data Acquisition	Request Oriented (nightly background job)	Data Availability Oriented (permanently active background job)
Load Cycle	1/day ... 1/week	1/minute ... 1/hour
Resource Consumption	Processing normally nights (load balancing)	Permanent Resource Consumption

Process Flow

The following figure illustrates the process flow for real-time data acquisition:



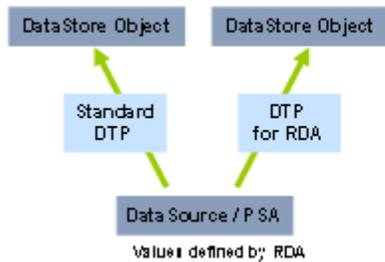
Data is loaded into BI at frequent, regular intervals and is then posted to the DataStore objects that are available for operational reporting. In BI, special InfoPackages are used for this purpose and data transfer processes for real-time data acquisition are created to further process data from the PSA in the DataStore objects. This is scheduled and executed regularly by a dedicated background process (the daemon). Data is available for reporting as soon as it has been successfully posted to the DataStore object and activated. Refresh the query display to display the recent data. The query shows the time that the query was last updated by a daemon run, even if no new data was posted.

Constraints

- We can only use real-time data acquisition to fill DataStore objects. A two-step data transfer is supported; data is first transferred into the PSA and then into the DataStore object. The DataStore object cannot be used as the source for a further real-time data transfer to another DataStore object.
- Master data cannot be transferred to the BI system with real-time data acquisition as otherwise the navigation attributes of the characteristic could no longer be used in aggregates. This is because aggregates cannot react to real-time updates since the change run in the RDA process cannot be triggered automatically for the loaded data.
- DataSources that are used for real-time data acquisition cannot be used in the delta process for standard data transfer (scheduled staging). A data transfer with RDA and a scheduled data transfer cannot be executed simultaneously in the delta process for a DataSource because there may be only one entry in the delta queue for each DataSource and target system.

In addition to a pure RDA update and a pure standard update, the following update methods, in which both transfer mechanisms are used in parallel, are possible for data that was transferred to the PSA by real-time data acquisition:

The data is updated from the PSA to a DataStore object with a DTP for real-time data acquisition, and to another DataStore object with a standard DTP.



- If you load data into a DataStore object with real-time data acquisition, you cannot load data into this DataStore object simultaneously with an additional DTP. This is because there can be only one open activation request in a DataStore object. Real-time data acquisition keeps an activation request open parallel to each DTP request. In a DTP request, multiple data packages can be loaded during a given time span. Each data package is activated in the DataStore object immediately after it is transferred. A further data transfer process cannot load into the same DataStore object as long as an activation request is open.

Depending on the requirements, you can nevertheless merge the data that you load with a real-time data acquisition in an InfoProvider with additional data sources..

- The DataStore object in which you load data with real-time data acquisition can be used in a MultiProvider or InfoSet.
- Using a process chain, you can restrict the time in which you load data into the DataStore object with real-time data acquisition. You can load data into the same DataStore object with a different data transfer process during the remainder of the time.

Two Ways Data Transfer Process

You can transfer data from the source to the entry layer of BI (the PSA) in two ways:

Using a service API

Data from an SAP source system can be loaded into the PSA using an InfoPackage created specifically for this purpose. This is triggered when the delta queue in the source system requests data. You have to simulate the initialization of the delta process for the DataSource beforehand.

The following two scenarios are possible:

- The source system application writes the data to the delta queue.
In this case, the daemon retrieves the data without calling the extractor.
- The application does not write data to the delta queue automatically; the extractor writes the data to the delta queue at the request of BI.

For extractors that transfer data synchronously from BI to the service API on request (generic extractors, for example), the daemon calls the extractor, and the extractor writes the data to the delta queue. The data is transferred to BI directly from the delta queue.

Using a Web service

You use the Web service to write the data from the source into the PSA. The transfer of data is controlled externally, without a request from BI. Only an InfoPackage (for full upload) is required to determine specific parameters for real-time data acquisition.

Data Acquisition through Web Service

Prerequisites

- Create a data source in BI
- Create interface between data source and web service (SAP XI)
- Create an info package for Demon with push scenario.
- Create Data Transfer Process with pull scenario and Data target.
- Create process chain to automate the jobs.
- Output: Created report based on master data table

Create a Data Source in BI

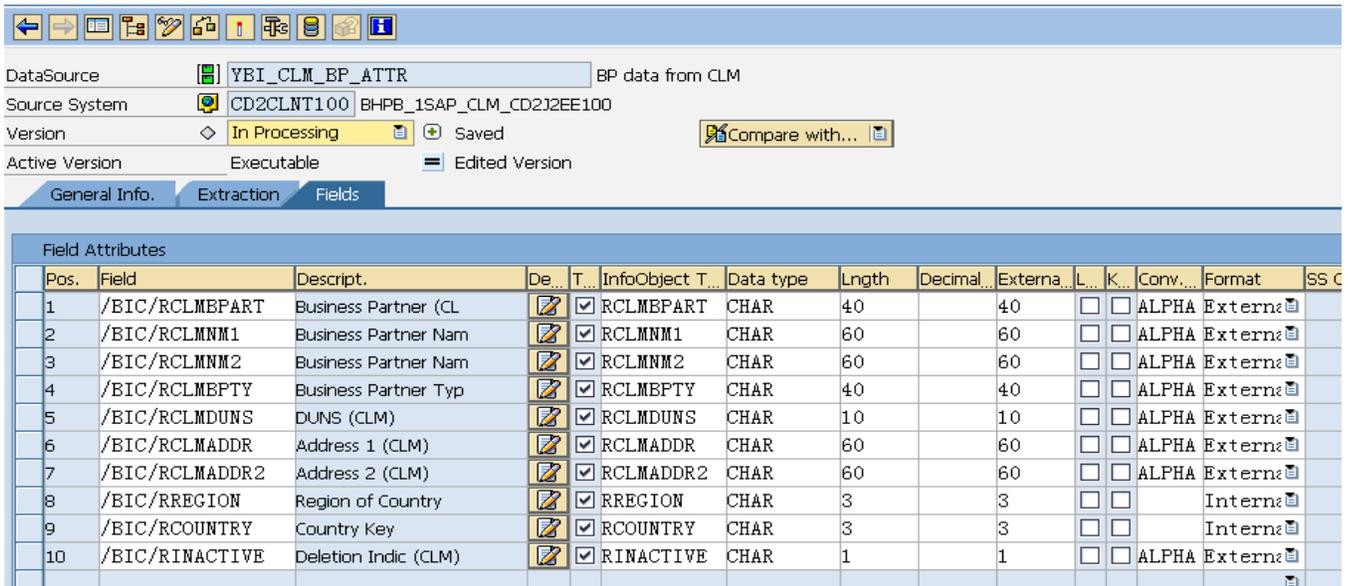
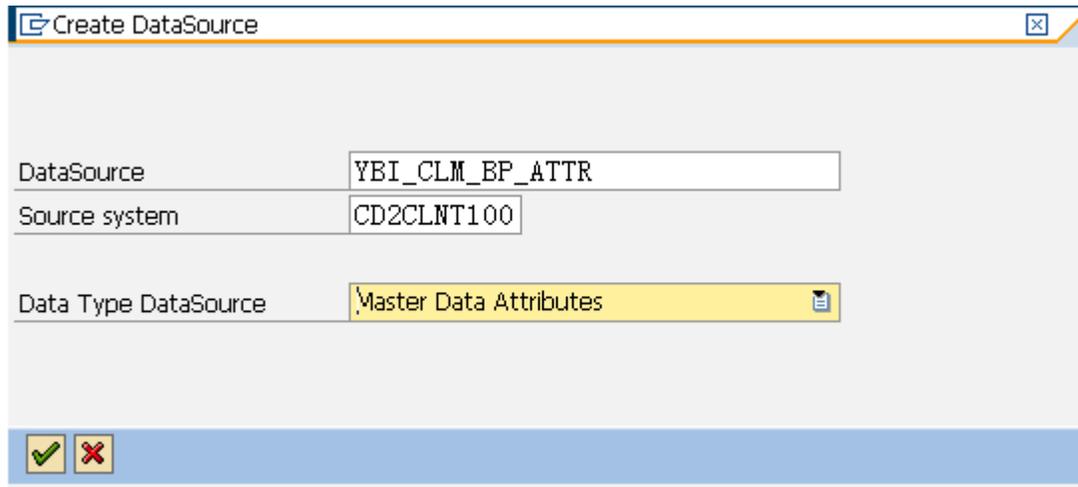
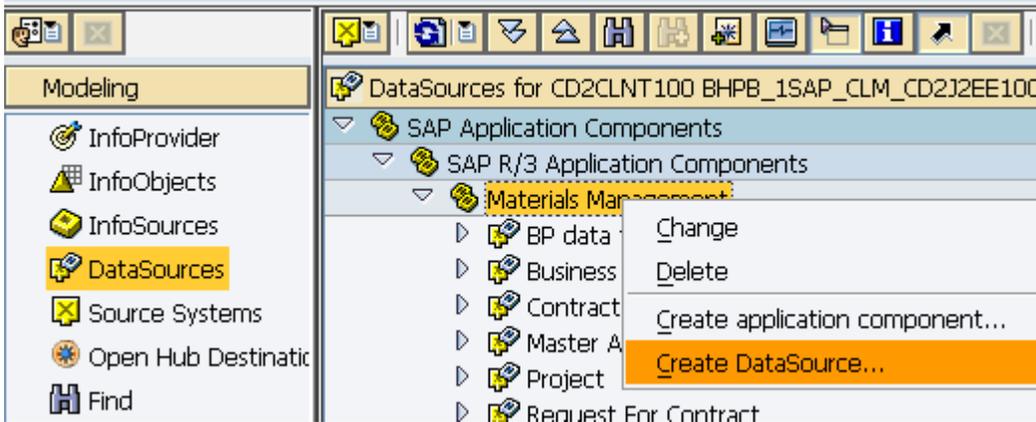
In BI through tcode RSA1 here we create a data source YBI_CLM_BP_ATTR with following fields

Source Structure		
Name	Field Length	Field Type
BusinessPartnerID	40	String
BusinessPartnerName1	60	String
BusinessPartnerName2	60	String
PartnerType	40	String
DUNSID	10	String
Address1	60	String
Address2	60	String
Region	3	String
Country	3	String

Data Warehousing Workbench: Modeling

The screenshot displays the SAP Data Warehousing Workbench Modeling interface. The left pane shows the 'Modeling' view with a tree structure including InfoProvider, InfoObjects, InfoSources, DataSources, Source Systems (highlighted), Open Hub Destination, Find, and Favorites. The right pane shows the 'Source Systems' configuration table with columns for Tech. Name, M..., Execute Function, and Display Tree. The table lists various source systems, including BI, SAP, External System, File, DB Connect, UD Connect, and Web Service. A specific source system, BHPB_1SAP_CLM_CD2J2EE100, is highlighted, showing its Tech. Name as CD2CLNT100 and its Execute Function as Display DataSource Tree.

Source Systems	Tech. Name	M ...	Execute Function	Display Tree
BI	BI		Create...	
SAP	SAP		Create...	
External System	PARTNERS		Create...	
File	FILE		Create...	
DB Connect	DB		Create...	
UD Connect	UDC		Create...	
Web Service	WEB		Create...	
BHPB_1SAP_CLM_CD2J2EE100	CD2CLNT100		Display DataSource Tree	

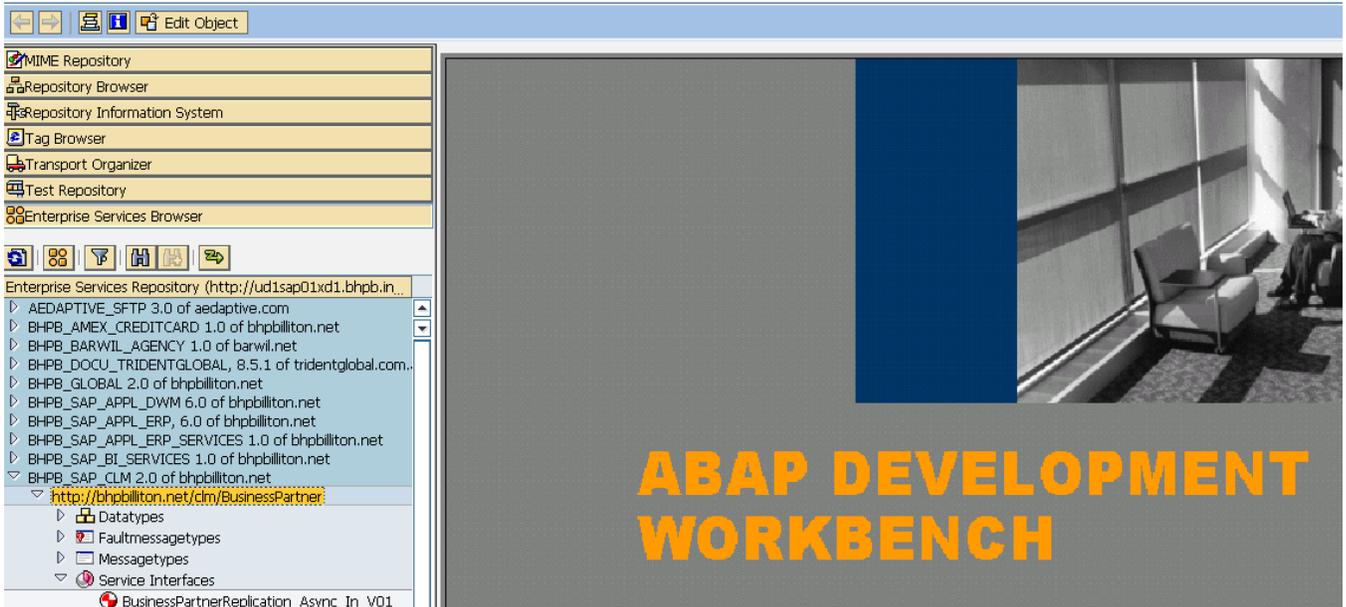


Then activate with button 

Create Interface between Data Source and Web Service (SAP XI)

Execute Tcode SPROXY for Enterprise Services Repository

Object Navigator



Generate a new service interface by double clicking on Business Partner 



Now enter the package, prefix and transport request for the proxy object then press continue.

The screenshot shows a wizard window titled "Enter Package/Request". The main area contains the instruction: "Enter the package, prefix and transport request for the proxy object." Below this is a section titled "Enter Package & Prefix" with three input fields: "Package", "Prefix", and "Request/Task". The "Package" field is highlighted in yellow. There is also a checkbox for "Local Object" which is currently unchecked. On the left side, a progress indicator shows three steps: "Enter Package/Request" (active, yellow background), "Configure Service" (disabled, red square), and "Complete" (disabled, red square). At the bottom, there are three buttons: "Back", "Continue", and "Cancel".

The screenshot shows the "Complete" step of the wizard. The main area contains the text: "On completion of the wizard the Proxy for **Object** , will be generated and displayed in the Proxy Editor." followed by "Do not forget to save or activate the result". On the left side, the progress indicator shows "Enter Package/Request" (disabled, blue text) and "Complete" (active, yellow background). At the bottom, the buttons are "Back", "Complete", and "Cancel".

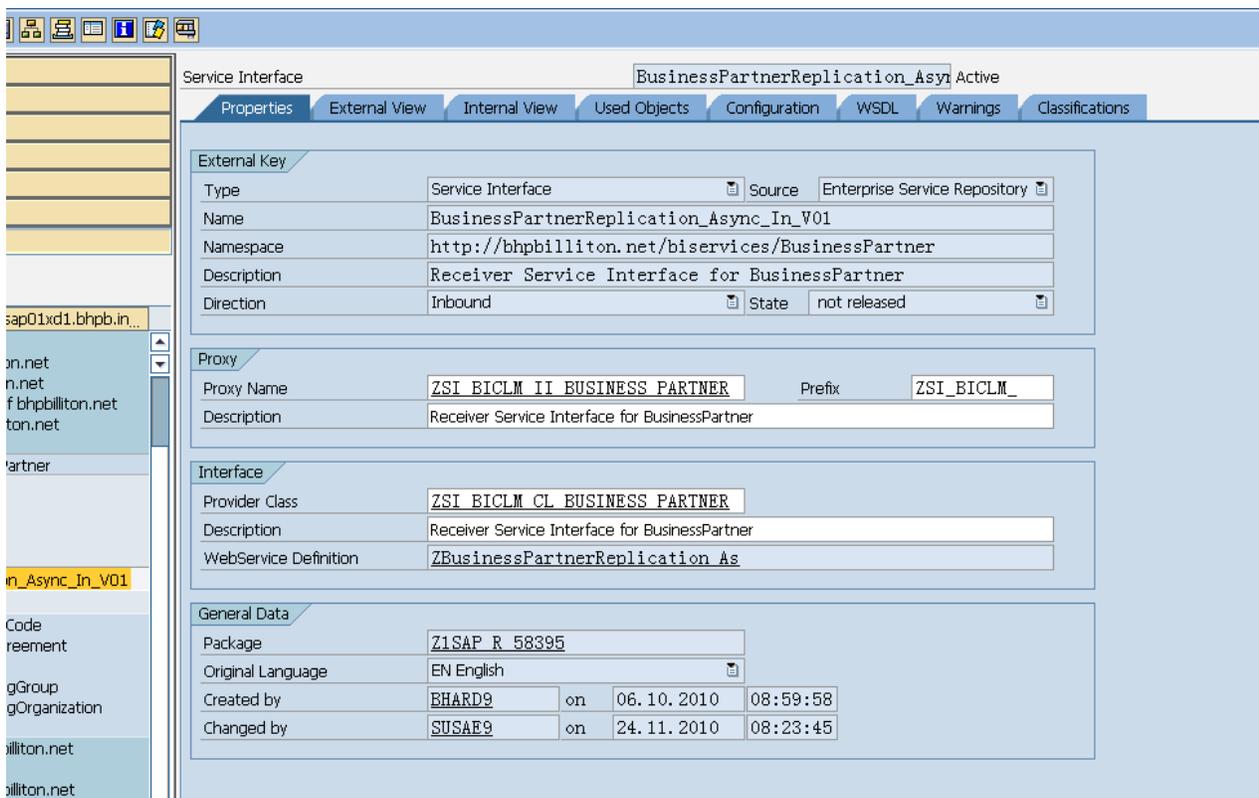
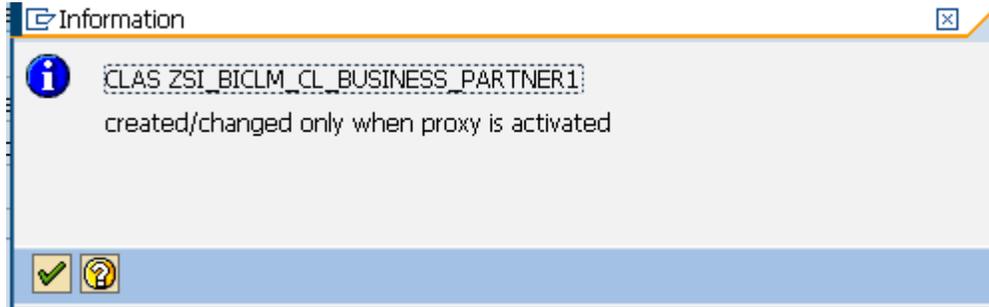
On completion of the wizard the Proxy for Object, will be generated and displayed in the Proxy Editor.

Do not forget to save or activate the result.

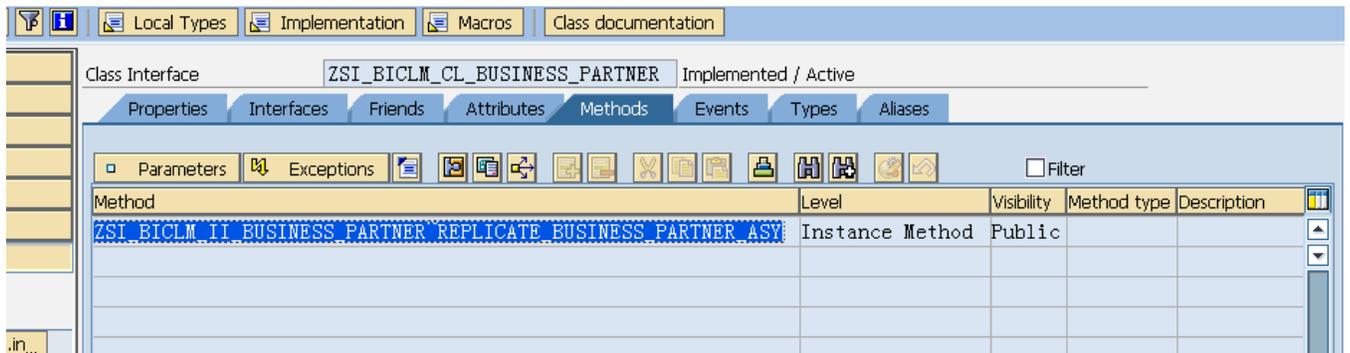
Once it completes, a proxy name and provider class will appear in service interface wizard then activate it.



Otherwise it gives information as



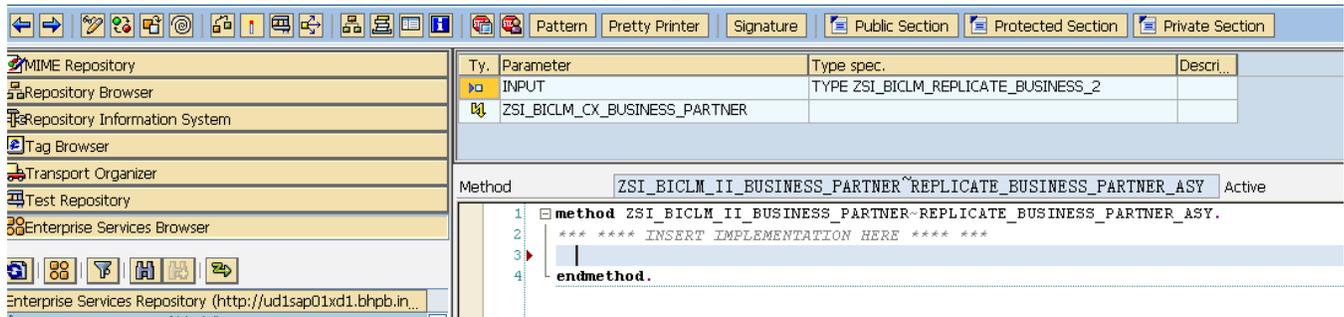
Now double click on class ZSI_BICLM_CL_BUSINESS_PARTNER. It will open method of class.



Now double click on method after activation.

ZSI_BICLM_II_BUSINESS_PARTNER~REPLICATE_BUSINESS_PARTNER_ASY.

Class Builder: Class ZSI_BICLM_CL_BUSINESS_PARTNER Change



Here we write the ABAP code between method & endmethod for one to one mapping between data source and interface fields.

```

-----
method ZSI_BICLM_II_BUSINESS_PARTNER~REPLICATE_BUSINESS_PARTNER_ASY.
*** ** INSERT IMPLEMENTATION HERE *** **
DATA: l_text TYPE string,
      l_s_data TYPE line of /BIC/WCQYBI_CLM00002000,
      l_t_data TYPE /BIC/WCQYBI_CLM00002000,
      LV_REG(3) TYPE C,
      LV_COUN(3) TYPE C .
FIELD-SYMBOLS <l_line> LIKE LINE OF input-REPLICATE_BUSINESS_PARTNER_ASY-
BUSINESS_PARTNER.

loop at input-REPLICATE_BUSINESS_PARTNER_ASY-BUSINESS_PARTNER ASSIGNING <l_line>.
  l_s_data-/BIC/RCLMBPART = <l_line>-BUSINESS_PARTNER_ID.
  l_s_data-/BIC/RCLMMN1 = <l_line>-BUSINESS_PARTNER_NAME1-CONTENT.
  l_s_data-/BIC/RCLMMN2 = <l_line>-BUSINESS_PARTNER_NAME2-CONTENT.
  l_s_data-/BIC/RCLMBPTY = <l_line>-PARTNER_TYPE.
  l_s_data-/BIC/RCLMDUNS = <l_line>-DUNSID.
  l_s_data-/BIC/RCLMADDR = <l_line>-ADDRESS1.
  l_s_data-/BIC/RCLMADDR2 = <l_line>-ADDRESS2.
  l_s_data-/BIC/RINACTIVE = <l_line>-INACTIVE.
  SPLIT <l_line>-REGION-CONTENT AT '_' INTO LV_REG LV_COUN.
  l_s_data-/BIC/RREGION = LV_REG.
  l_s_data-/BIC/RCOUNTRY = <l_line>-COUNTRY.

  APPEND l_s_data TO l_t_data.
  clear l_s_data.
endloop.

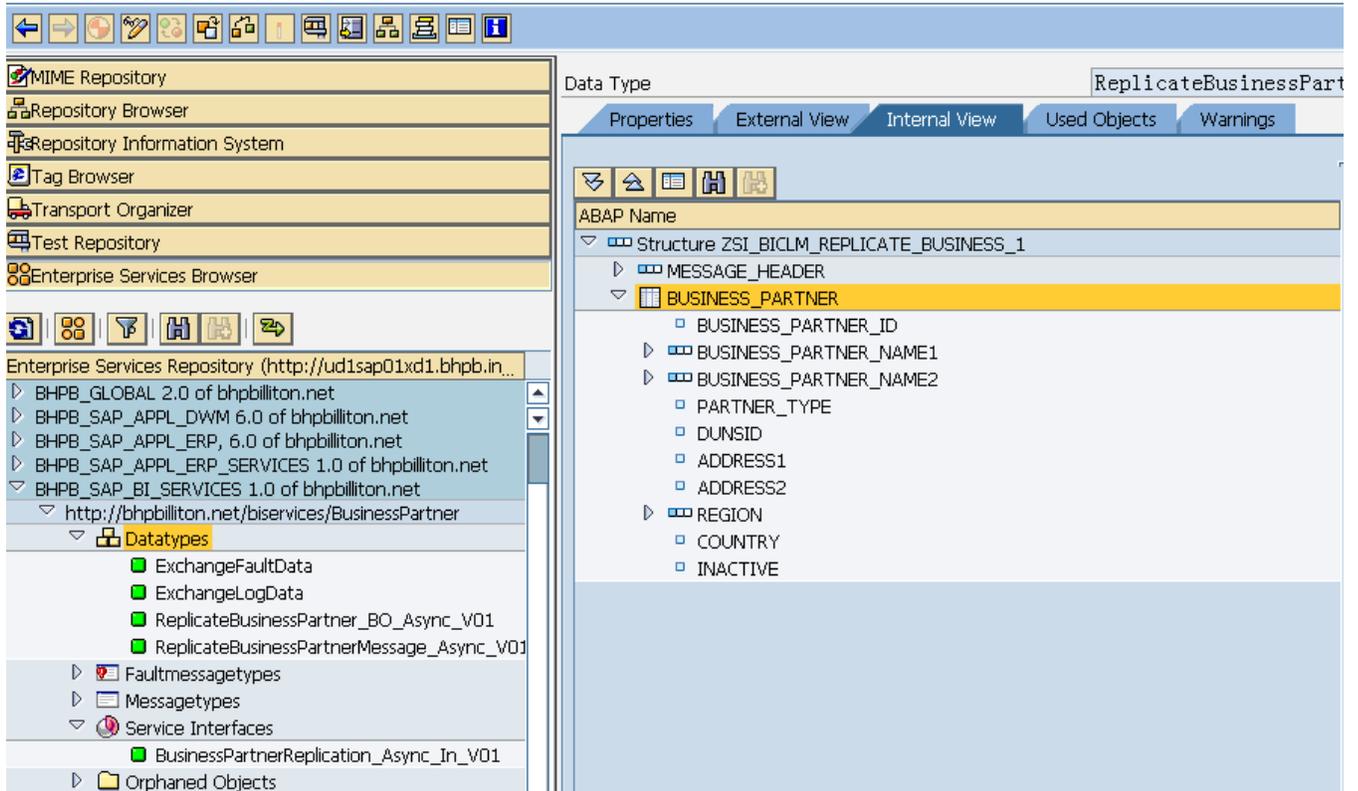
CALL FUNCTION '/BIC/CQYBI_CLM00002000' (Function module name of Data source)
EXPORTING
  DATA = l_t_data
EXCEPTIONS
  OTHERS = 1.
IF SY-SUBRC <> 0.

ENDIF.
endmethod.
-----

```

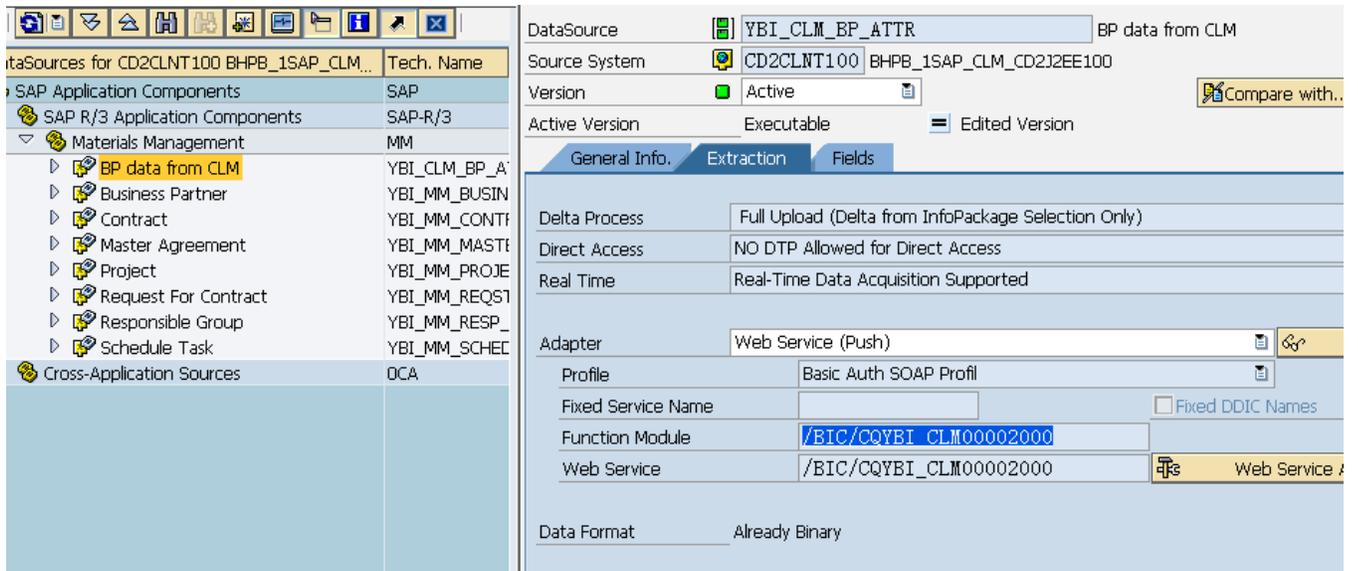
Note: 1. Here we consider interface's fields name from internal view tab.

Display Data Type ReplicateBusinessPartnerMessage_Async_V01



- In ABAP coding,
- Data Type and
 - CALL FUNCTION

Value should be considered as Function module name of related data source as mentioned in below figure.



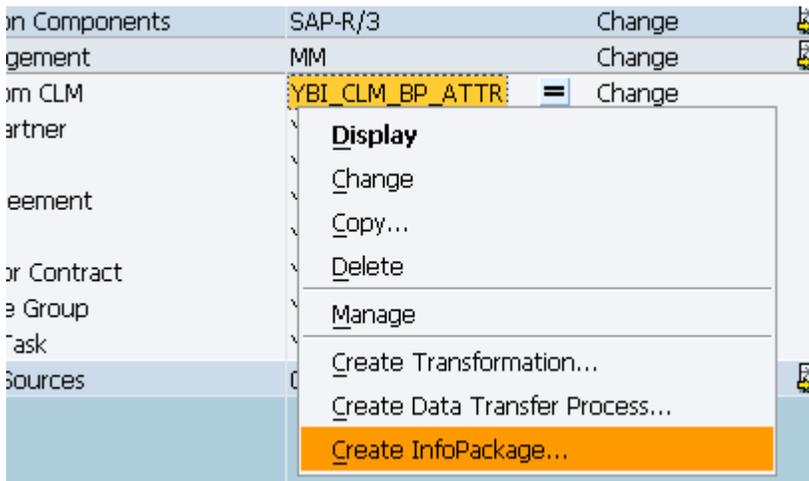
Once coding is done, we activate again and now interface icon will be turned into green ( → )



Now interface has been mapped with data source YBI_CLM_BP_ATTR.

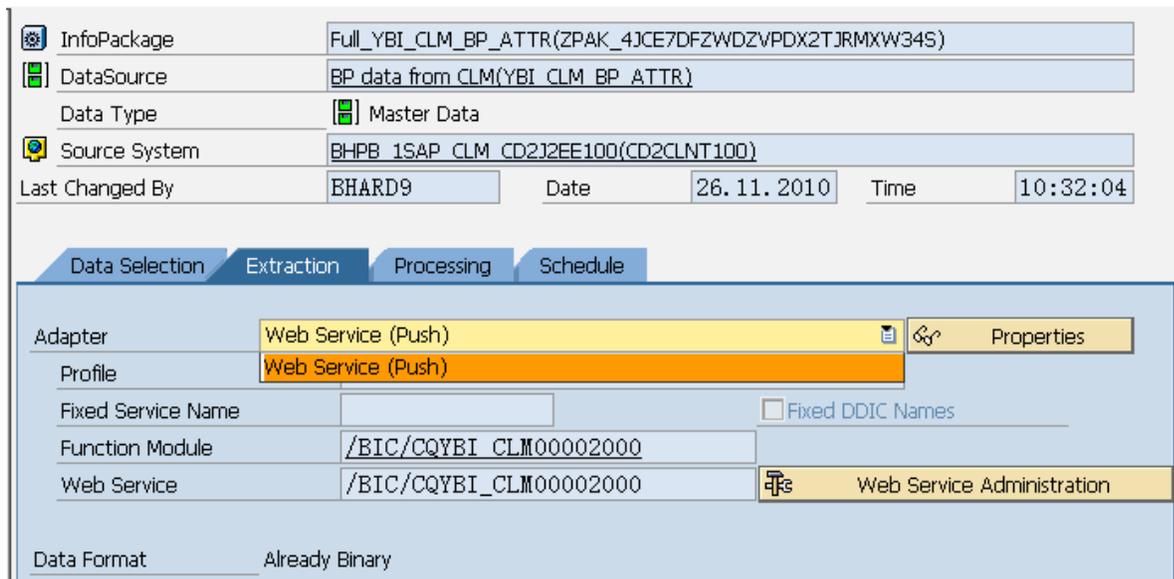
Create an Info Package for Demon with Push Scenario

Create an info package Full_YBI_CLM_BP_ATTR on data source.



This info package follows the Push mechanism where its attributes are

- Simulated Delta Initialization Supported
- No Data Parsing Necessary
- Adapter Is Multi-Segment Capable
- Real-Time Data Acquisition via Push



After creation of info package we click on Assign Daemon as

InfoPack.	MM	Last Changed By	Date
P data from CLM	YBI_CLM_BP_A	BHARD9	
Full_YBI_CLM_BP_ATTR	ZPAK_4JCE7DF		
Dataflow Upwards	_DATAFLOW_L		
Business Partner	YBI_MM_BUSIN		
Contract	YBI_MM_CONTR		
Master Agreement	YBI_MM_MASTE		
Project	YBI_MM_PROJE		
Request For Contract	YBI_MM_REQST		
Responsible Group	YBI_MM_RESP_		
Schedule Task	YBI_MM_SCHEC		

Assign InfoPackage for Real-Time Data Acquisition to Daemon

Create daemon as

Monitor: Real-Time Data Acquisition

Daemon / DataSource / InfoPack. / Req. ID	Data Target
Unassigned Objects	

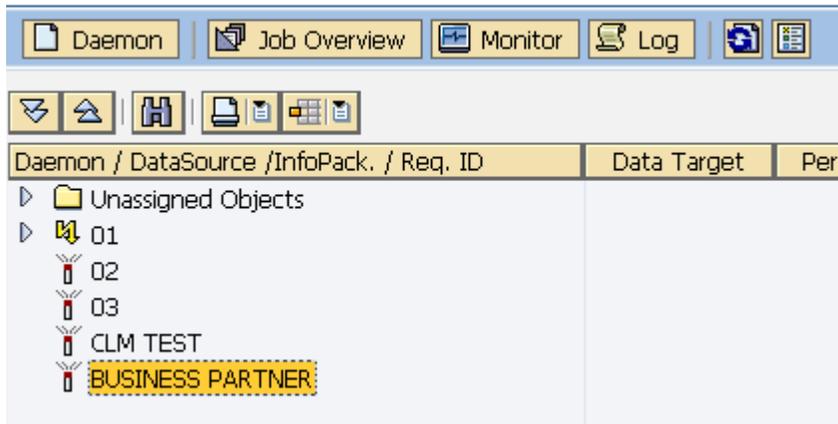
Here we define short description of daemon and time, where daemon schedules every slotted time interval.

Daemon Settings

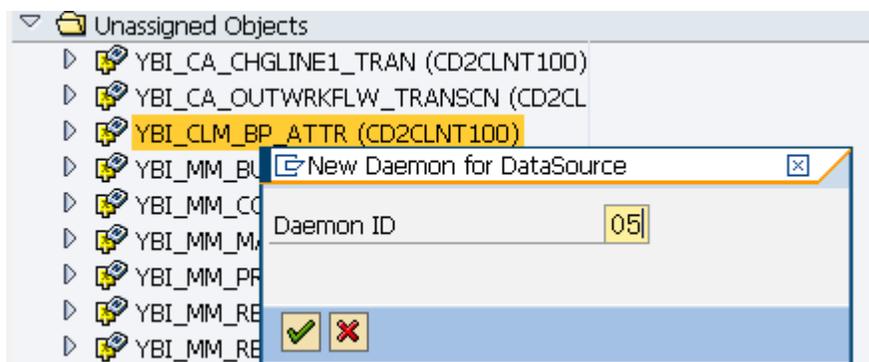
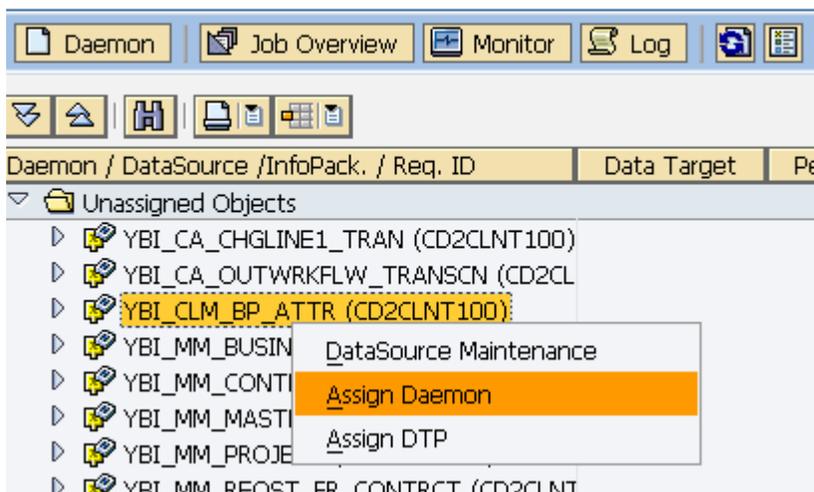
Daemon	05
Short Description	BUSINESS PARTNER
Period [Min]	0001

Assign your related data source to daemon no. 5 (BUSINESS PARTNER) from unassigned object.

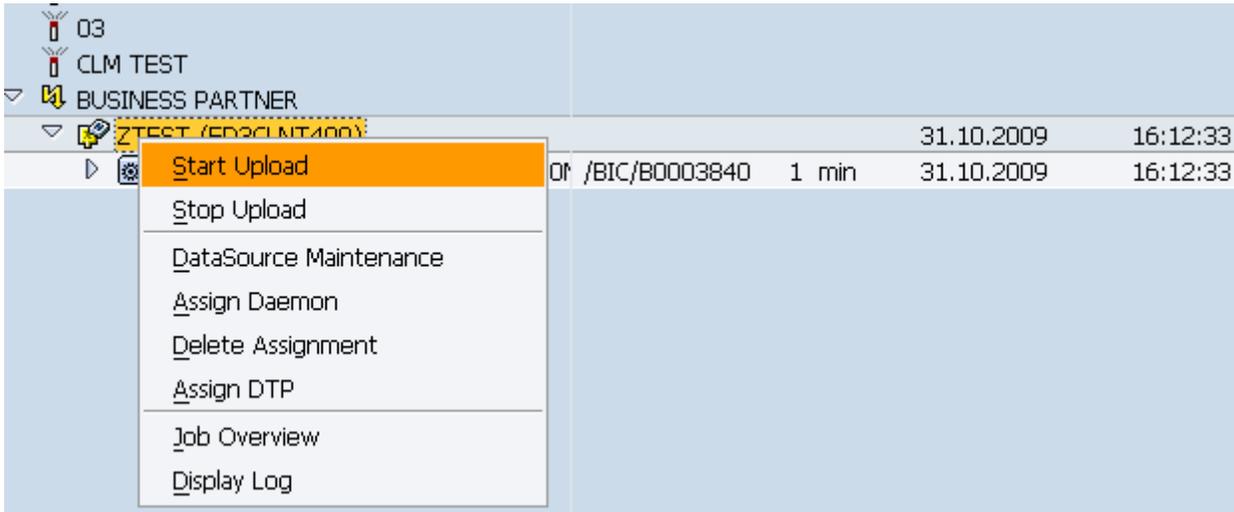
Monitor: Real-Time Data Acquisition



Monitor: Real-Time Data Acquisition



Now we start the daemon for upload data at PSA level.



It however, starts the data load.

DataSource /InfoPack. / Req. ID	Data Target	Period	Last Upload	Last Upload	Records in Upload	Records in Request	Error
assigned Objects							
YBI_CA_CHGLINE1_TRAN (CD2CLNT100)			26.11.2010	10:11:11			
YBI_CA_OUTWRKFLW_TRANSCN (CD2CLNT100)			26.11.2010	10:11:11			
YBI_CLM_BP_ATTR (CD2CLNT100)			26.11.2010	05:32:04			
OCO ZPAK_4JCE7DFZWDZVPDX2TJRMXW34 /BIC/B0004695	/BIC/B0004695	min	26.11.2010	05:32:04	2	2	
OCO REQU_4JYTCYXZG5B81OTZD0003 /BIC/B0004695	/BIC/B0004695				2	2	
YBI_MM_BUSINESS_PARTNER (CD2CLNT100)							

Here daemon will be trigger every min because it is defined in its daemon setting.

Re...	W...	Moni...	SID of the R...	InfoPackage	Request	Loading date	Loading ti...	Te
194193		OCO		Full_YBI_CLM_BP_ATTR	REQU_4JYTCYXZG5B81OTZD0003DOM4	26.11.2010	05:27:08	ZP
189237		OCO		Full_YBI_CLM_BP_ATTR	REQU_4JYJTTL3W3YDGM7C10JZA3G	25.11.2010	08:54:51	ZP
189091		OCO		Full_YBI_CLM_BP_ATTR	REQU_4JYAFKRL4756Q6LT5GZHSENZ0	25.11.2010	07:01:14	ZP
183574		OCO		Full_YBI_CLM_BP_ATTR	REQU_4JXRCOR5J8ACKOFYABKKO06VW	24.11.2010	03:58:05	ZP

Here we can check data load in PSA to given data source.

Monitor - Administrator Workbench

Header	Status	Details
Total	OCO	
Technical	OCO	Correct data request; processing run
Wait Until	25.12.2010	05:32:03
Processing Step		Service API

Create Data Transfer Process with Pull Scenario and Data Target

Now we create an info object (RCLMBPART) as a master data where data will be populate in this table.

Business Partner (CLM)	RCLMBPART	Change	InfoObjects
Business Partner (CLM) (Attribute)	ATTRIBUTES RCLMBPART	Manage	InfoProvider
RSDS YBI_CLM_BP_ATTR CD2CLNT100 -> IOBJ RCLMBPART	OG2SJK262X0HJ1MCHQWL2UIT15M62BCY	Change	
RSDS YBI_MM_MASTER_AGREEMENT CD2CLNT100 -> IOBJ RCLMBPART	O20FBY7D5USU2MWHXYVVIADJKYVXJD9	Change	
Data Transfer Processes	ATTRIBUTES RCLMBPART	Create Data Tr...	
YBI_CLM_BP_ATTR / CD2CLNT100 -> RCLMBPART	DTP_4JC2L14T9NVFCGAVNOZGID5UK	Change	
BP data from CLM	YBI_CLM_BP_ATTR	Change	DataSources
Full_YBI_CLM_BP_ATTR	ZPAK_4JCE7DFZWDZVPOX2TJRMXW34S	Schedule	DataSources

Here transformation is 1 to 1 mapping.

Transformation Display

Transformation: RSDS YBI_CLM_BP_ATTR CD2CLNT100 -> IOBJ RCLMBPART

Source: BP data from CLM (YBI_CLM_BP_ATTR)

Target: Business Partner (CLM) (RCLMBPART)

Version: Active

Active Version: Executable

Edited Version: []

Posi	Ke	Field	Descript.
1	/BIC/RCLMBPART	Business Partner (CL	
2	/BIC/RCLMNM1	Business Partner Nam	
3	/BIC/RCLMNM2	Business Partner Nam	
4	/BIC/RCLMBPTY	Business Partner Typ	
5	/BIC/RCLMDUNS	DUNS (CLM)	
6	/BIC/RCLMADDR	Address 1 (CLM)	
7	/BIC/RCLMADDR2	Address 2 (CLM)	
8	/BIC/RREGION	Region of Country	
9	/BIC/RCOUNTRY	Country Key	
10	/BIC/RINACTIVE	Deletion Indic (CLM)	

Rul	Rule Name	Posi	Key	InfoObject	Icor	Descript.	Inte
=	RCLMBPART	1		RCLMBPART	[]	Business Partner (CLM)	[]
=	RCLMNM1	2		RCLMNM1	[]	Business Partner Name 1 (CLM)	[]
=	RCLMNM2	3		RCLMNM2	[]	Business Partner Name 2 (CLM)	[]
=	RCLMBPTY	4		RCLMBPTY	[]	Business Partner Type (CLM)	[]
=	RCLMDUNS	5		RCLMDUNS	[]	DUNS (CLM)	[]
=	RCLMADDR	6		RCLMADDR	[]	Address 1 (CLM)	[]
=	RCLMADDR2	7		RCLMADDR2	[]	Address 2 (CLM)	[]
=	RCOUNTRY	8		RCOUNTRY	[]	Country	[]
=	RREGION	9		RREGION	[]	Region of Country	[]
[]		10		ROBJCOUNT	[]	Object Count	[]
=	RINACTIVE	11		RINACTIVE	[]	Deletion Indicator (CLM)	[]

Then we create DTP to populate the data in Master Table with Pull scenario.

Data Transfer Process: YBI_CLM_BP_ATTR / CD2CLNT100 -> RCLMBPART

ID: DTP_4JC2L14T9NVFCGAVNOZGID5UK

Version: Active

Delta Status: Active

Extraction Mode: Delta

Package Size: 50,000

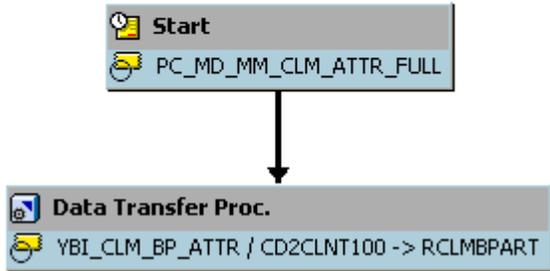
Extraction: [] Only Get Delta Once

Execute: [x] Get All New Data Request By Request

Create Process Chain to Automate the Jobs

To automate the job we create process chain

PC_MD_MM_CLM_ATTR_FULL



Note: Here we can't assign info package because that info package follow PUSH scenario and can not be appeared during variant assignment.

Output: Created Report based on Master Data Table

Business Partner Data Report							
Author BHARD9				Status of Data 25/11/2010 05:05:08			
Filter		Information					
Table							
Business Partner (CLM)	Business Partner Name 1	Business Partner Name 2	Business Partner Type	DUNS (CLM)	Address 1 (CLM)	Address 2 (CLM)	Region of Country
BP-0000407	BHP Billiton BP21	BP	Local	22518748	Shenton way	test_mf	SG/#
BP-0000408	TEST_FOR_BW	BW TEST1	Local	112223333	ADDRESS TEST 1	ADDRESS TEST 2	IN/#
BP-0000409	BW test 2	test 2 name 2	Local	334446666	test 2 -address1	test 2 -address2	SG/#
BP-0000410	BW test3	Test3 BW	Local	112223333	MUM001, India	MUM002, Bhart	BR/10_
BP-0000411	BW TEST 4	BW TEST 4 lower case	Local	445557777	address1	addreas 2	IN/01
BP-0009593	BHP Billiton BP41	BP	Local	42518748	Shenton way	Not assigned	SG/SG
BP-0010193	BHP Billiton810	BP	Local	442448751	Shenton way	Level 10	SG/SG
BP-0010196	BHP Billiton77	BP	Local	402488748	Shenton way	Level 10	SG/SG
BP-0019955	BHP Billiton801	Not assigned	Local	185448748	Shenton way	Not Applicable	SG/SG

Related Content

For more information, visit the [EDW homepage](#)

Copyright

© Copyright 2010 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.