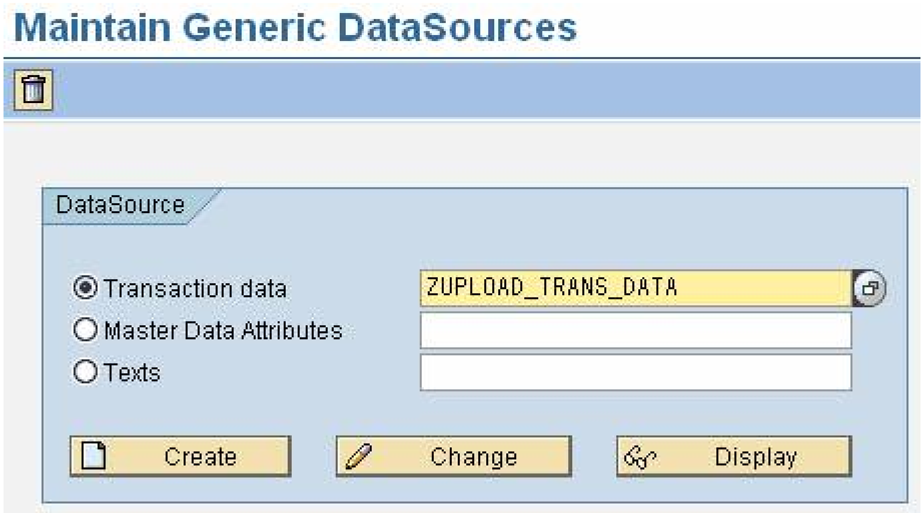
TOPICS COVERED

- To upload Sales Order data from SAP R/3 to BI using generic DataSources
- Integrating SAP data (R/3 or BW Data) with the BO-Xcelsius (Business Objects) using BSP
    - **Part 1**: SAP BSP + BO-Xcelsius (Excel XML Maps Connection)
    - **Part 2**: SAP BSP + BO-Xcelsius (web Service Connection)
    - **Part 3**: SAP BSP + BO-Xcelsius (XML Data Connection)
- Integrating BO-Xcelsius (SWF) file with BO-Crystal Reports

To upload Sales Order data from SAP R/3 to BI using generic DataSources

Steps to be followed in R/3

Step 1: Goto T-code **RSO2.** Give the **DataSource** name in Transactional data (as we are uploading transactional data) and click on **Create**



Maintain Generic DataSources

DataSource

● Transaction data ZUPLOAD_TRANS_DATA

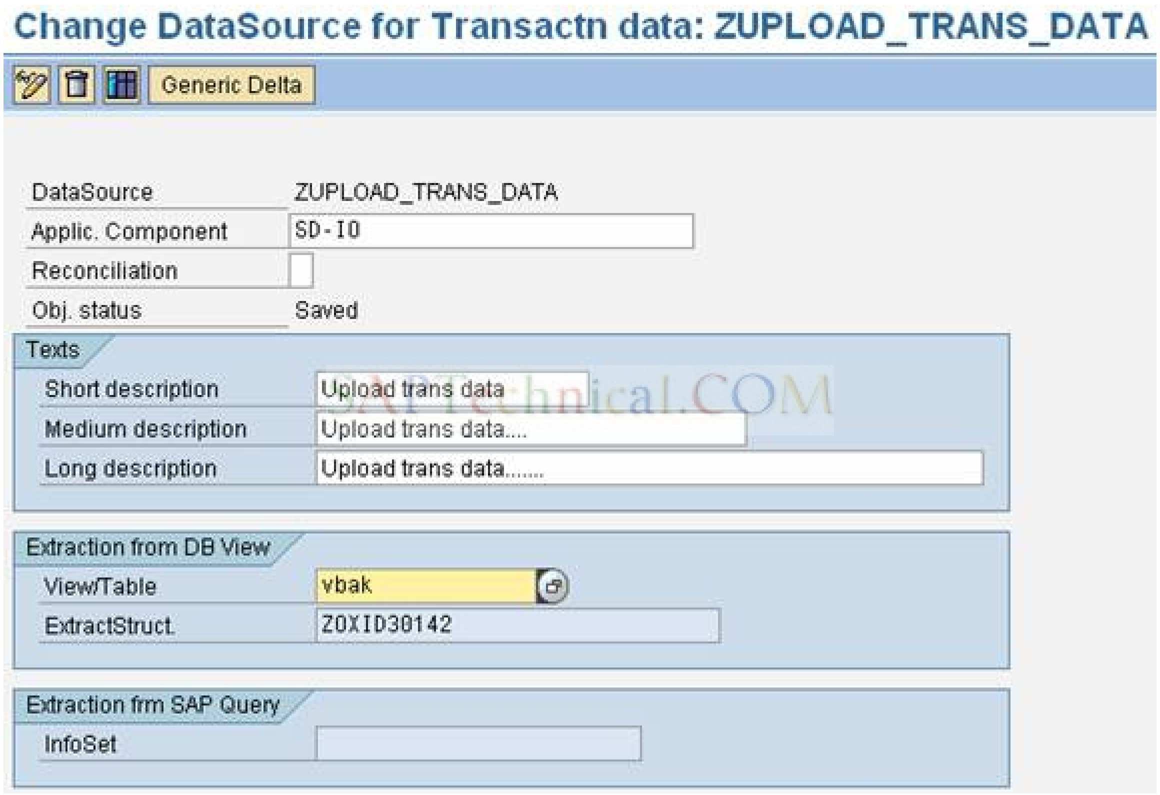
○ Master Data Attributes

○ Texts

Step 2:

1. Specify the Application Component name
2. Fill in the text column with meaningful description
3. In “**Extraction from DB view**” give the **table/View** name from which you want to extract data.
4. **SAVE**

Note: Click on F4 to chose Application Component name



Change DataSource for Transactn data: ZUPLOAD_TRANS_DATA

DataSource	ZUPLOAD_TRANS_DATA
Applic. Component	SD-IO
Reconciliation	□
Obj. status	Saved

Texts

Short description	Upload trans data
Medium description	Upload trans data....
Long description	Upload trans data.......

Extraction from DB View

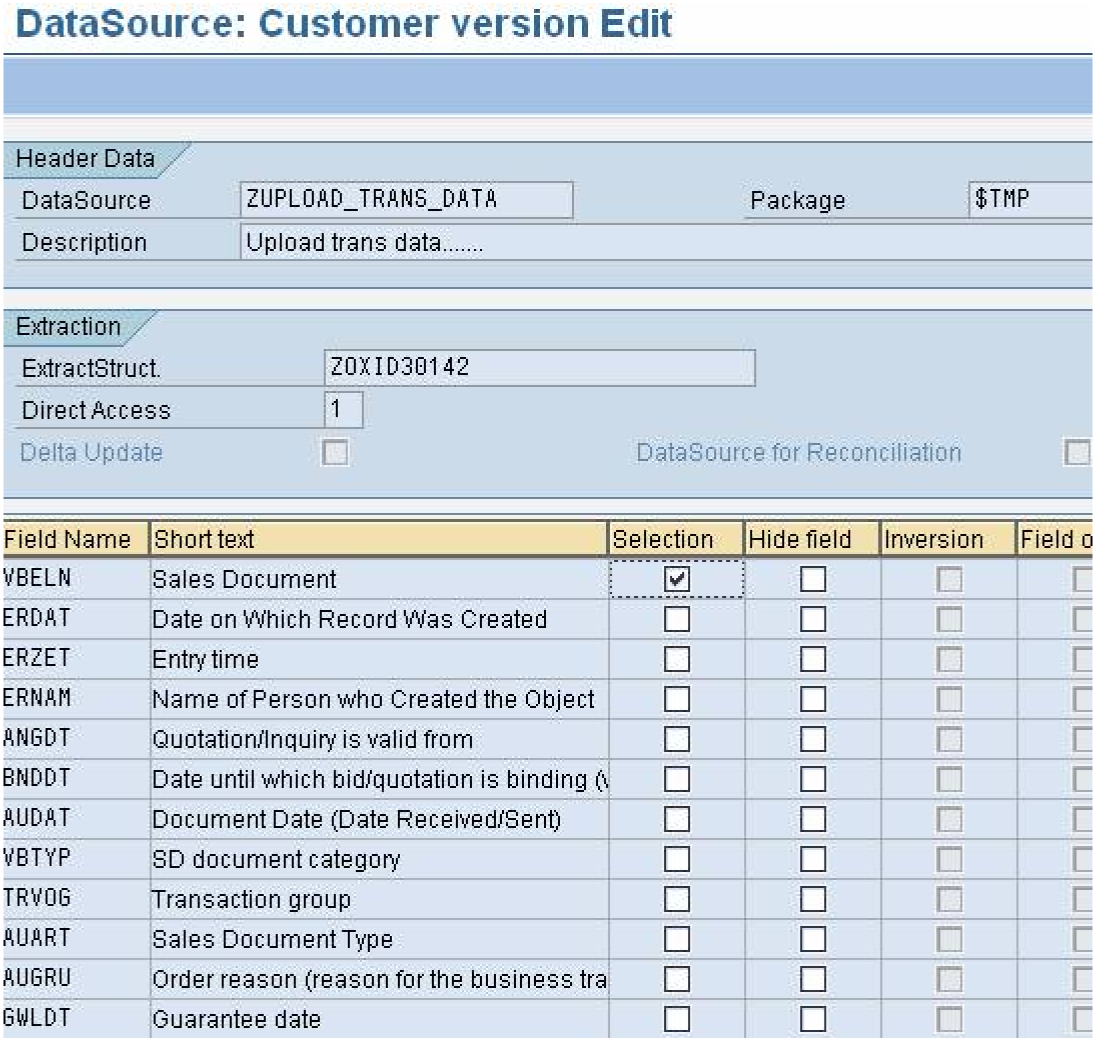
View/Table	vbak
ExtractStruct.	ZOXID30142

Extraction frm SAP Query

InfoSet	

Step 3: Clicking on **SAVE** will lead to the below screen.

Note: We can do “**Selection”**, “**Hide”** the fields etc as shown in the below screen shot



DataSource: Customer version Edit

Header Data

DataSource: ZUPLOAD_TRANS_DATA

Package: $TMP

Description: Upload trans data.......

Extraction

ExtractStruct.: ZOXID30142

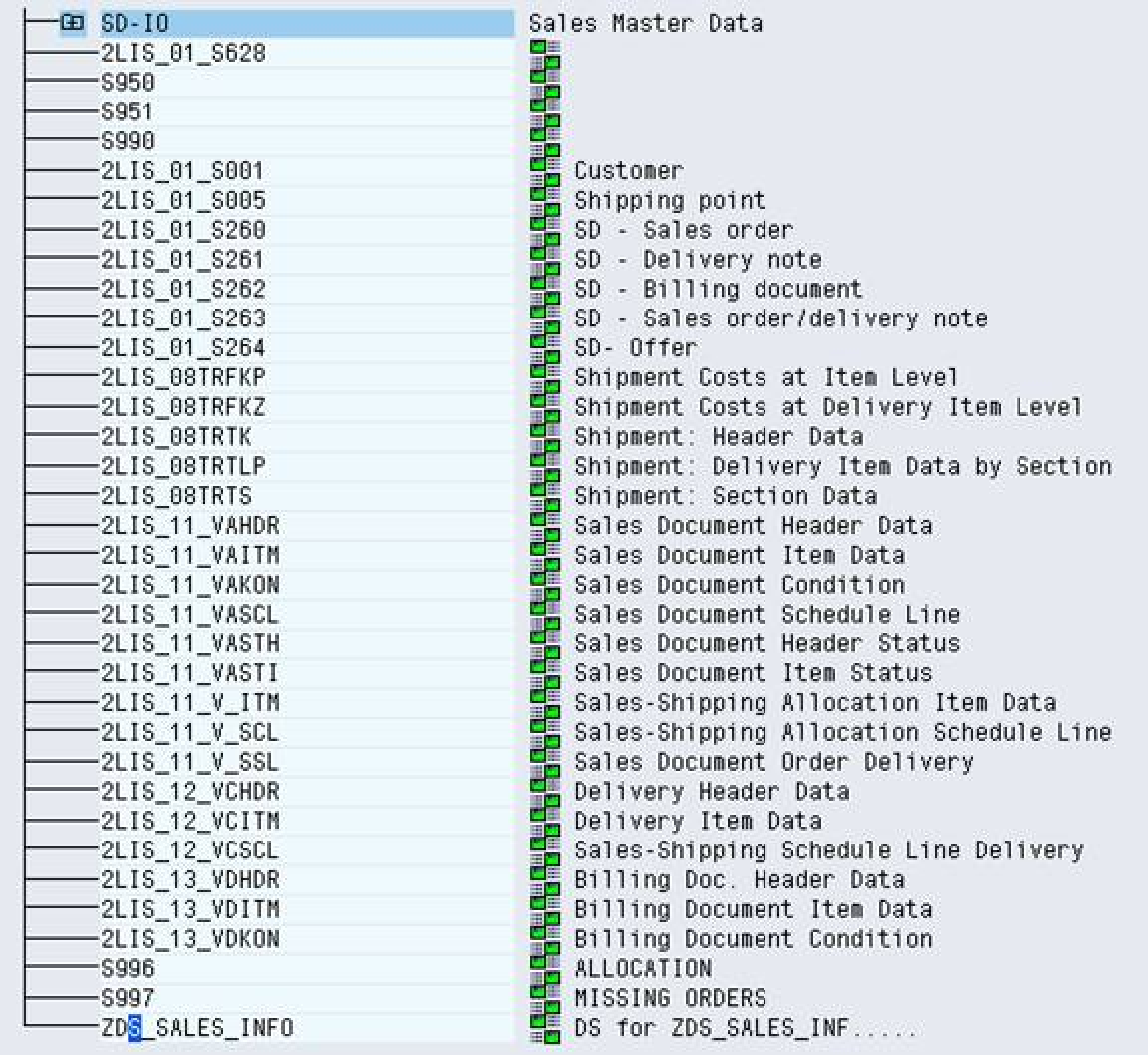
Direct Access: 1

Delta Update □

DataSource for Reconciliation □

Field Name	Short text	Selection	Hide field	Inversion	Field o
VBELN	Sales Document	☑	□	□	□
ERDAT	Date on Which Record Was Created	□	□	□	□
ERZET	Entry time	□	□	□	□
ERNAM	Name of Person who Created the Object	□	□	□	□
ANGDT	Quotation/Inquiry is valid from	□	□	□	□
BNDDT	Date until which bid/quotation is binding (v	□	□	□	□
AUDAT	Document Date (Date Received/Sent)	□	□	□	□
VBTYP	SD document category	□	□	□	□
TRVOG	Transaction group	□	□	□	□
AUART	Sales Document Type	□	□	□	□
AUGRU	Order reason (reason for the business tra	□	□	□	□
GWLDT	Guarantee date	□	□	□	□

Step 4: Go to T-code **RSA6** to know if the DataSource is successfully activated



SD-IO	Sales Master Data
2LIS_01_S628	
S950	
S951	
S990	
2LIS_01_S001	Customer
2LIS_01_S005	Shipping point
2LIS_01_S260	SD - Sales order
2LIS_01_S261	SD - Delivery note
2LIS_01_S262	SD - Billing document
2LIS_01_S263	SD - Sales order/delivery note
2LIS_01_S264	SD- Offer
2LIS_08TRFKP	Shipment Costs at Item Level
2LIS_08TRFKZ	Shipment Costs at Delivery Item Level
2LIS_08TRTK	Shipment: Header Data
2LIS_08TRTLP	Shipment: Delivery Item Data by Section
2LIS_08TRTS	Shipment: Section Data
2LIS_11_VAHDR	Sales Document Header Data
2LIS_11_VAITM	Sales Document Item Data
2LIS_11_VAKON	Sales Document Condition
2LIS_11_VASCL	Sales Document Schedule Line
2LIS_11_VASTH	Sales Document Header Status
2LIS_11_VASTI	Sales Document Item Status
2LIS_11_V_ITM	Sales-Shipping Allocation Item Data
2LIS_11_V_SCL	Sales-Shipping Allocation Schedule Line
2LIS_11_V_SSL	Sales Document Order Delivery
2LIS_12_VCHDR	Delivery Header Data
2LIS_12_VCITM	Delivery Item Data
2LIS_12_VCSCL	Sales-Shipping Schedule Line Delivery
2LIS_13_VDHDR	Billing Doc. Header Data
2LIS_13_VDITM	Billing Document Item Data
2LIS_13_VDKON	Billing Document Condition
S996	ALLOCATION
S997	MISSING ORDERS
ZDS_SALES_INFO	DS for ZDS_SALES_INF.....

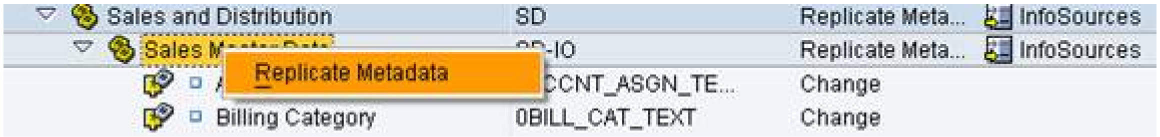
Note: Only Datasources which are activated will be displayed in RSA6 transaction

Steps to be followed in BI

Note: The RFC connection should be configured before you replicate data to BI system

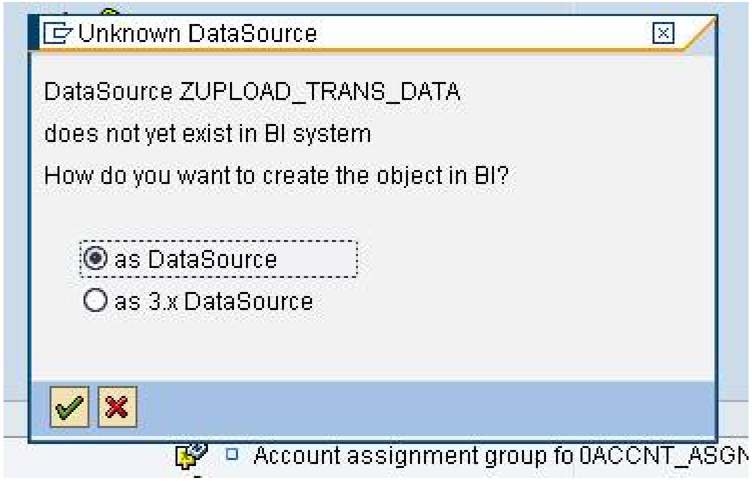
Step1:

1. Goto RSA1 => DataSources
2. Rtclk on Sales and Distribution tree => **Replicate Metadata**
3. **Activate**



Sales and Distribution	SD	Replicate Meta...	InfoSources
Sales M	D-IO	Replicate Meta...	InfoSources
A	CCNT_ASGN_TE...	Change	
Billing Category	0BILL_CAT_TEXT	Change	

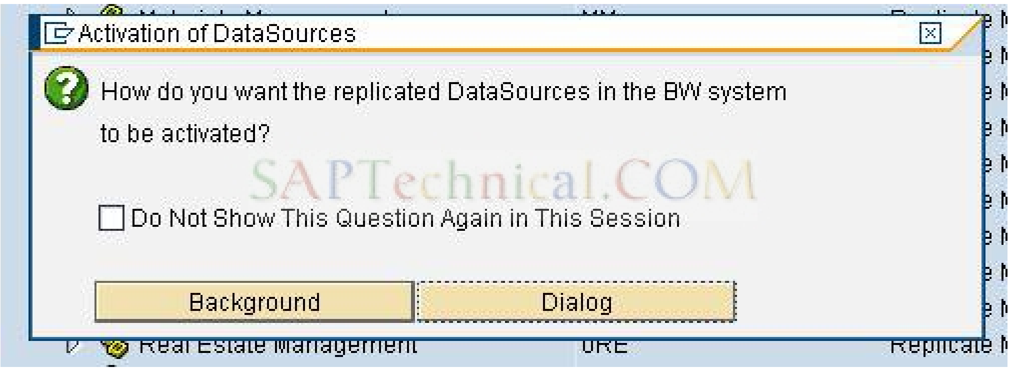
Step 2: Once the DataSource is replicated, you get a below popup, we need to choose "**as DataSource**" as we are using BI 7.0



Unknown DataSource

DataSource ZUPLOAD_TRANS_DATA
does not yet exist in BI system
How do you want to create the object in BI?

● as DataSource
○ as 3.x DataSource



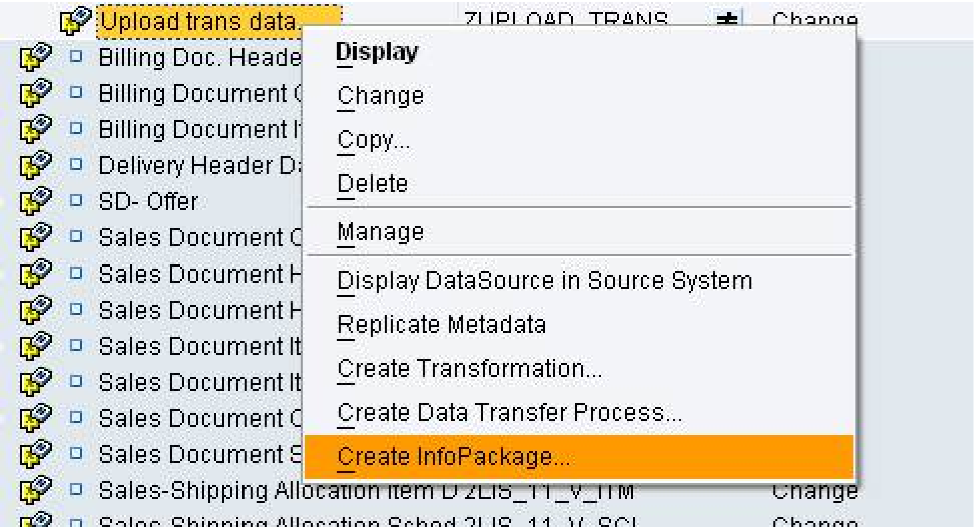
Activation of DataSources

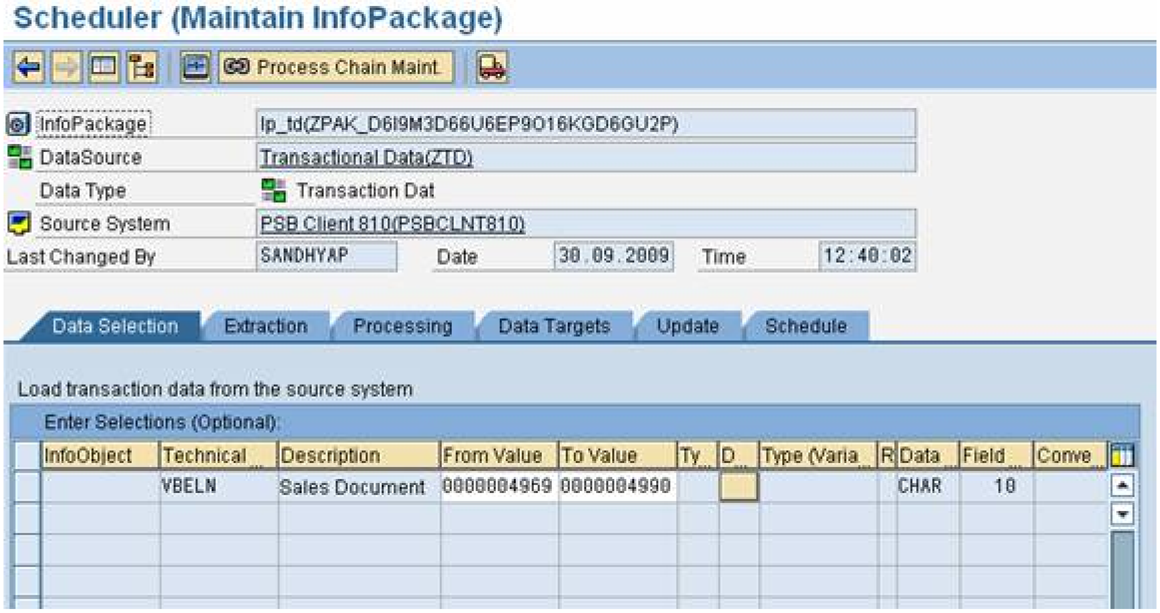
How do you want the replicated DataSources in the BW system to be activated?

□ Do Not Show This Question Again in This Session

Step 3: Create **InfoPackage** on the DataSource

InfoPackage: It acts as a link to extract data from Source system and gives it to PSA





Scheduler (Maintain InfoPackage)

InfoPackage	Ip_td(ZPAK_D6I9M3D66U6EP9O16KGD6GU2P)				
DataSource	Transactional Data(ZTD)				
Data Type	Transaction Dat				
Source System	PSB Client 810(PSBCLNT810)				
Last Changed By	SANDHYAP	Date	30.09.2009	Time	12:40:02

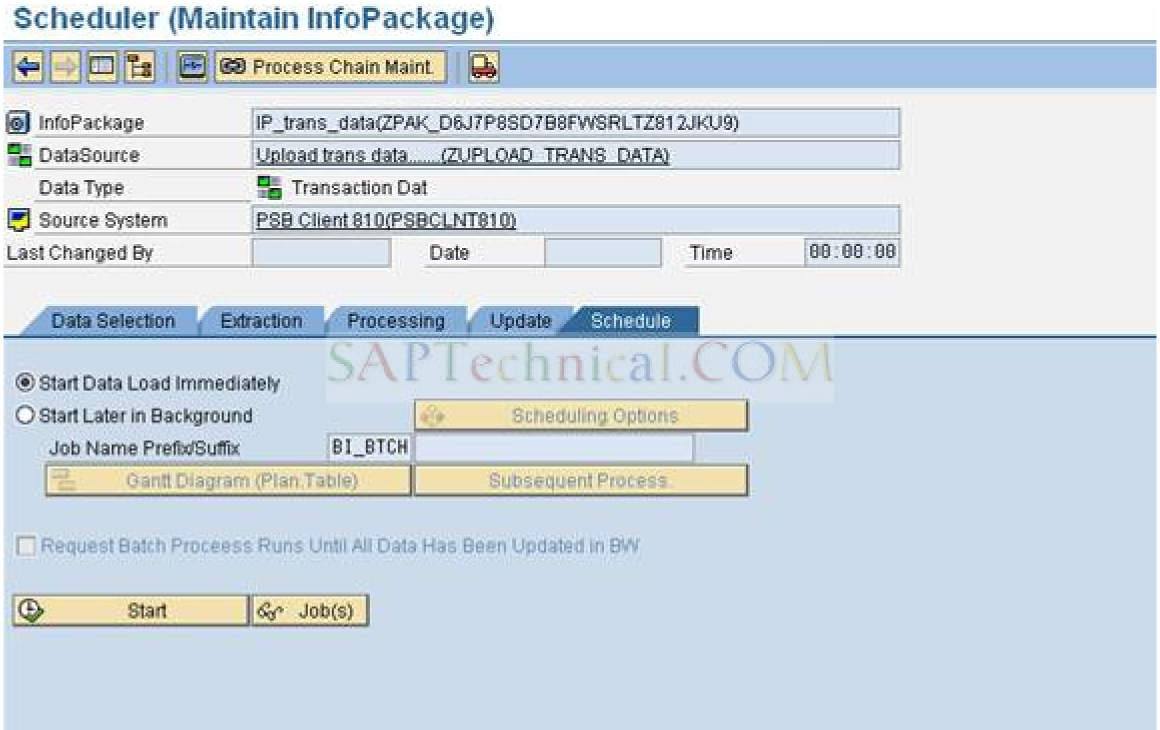
Load transaction data from the source system

Enter Selections (Optional):

InfoObject	Technical ...	Description	From Value	To Value	Ty	D	Type (Varia	R	Data	Field	Conve
	VBELN	Sales Document	0000004969	0000004990					CHAR	10	

Note: We can restrict data in "**Data Selection**" tab once you create InfoPackage as shown in the below screen shot

Step 4: Goto "**Schedule"** and click on **"Start"**



Scheduler (Maintain InfoPackage)

InfoPackage	IP_trans_data(ZPAK_D6J7P8SD7B8FWSRLTZ812JKU9)				
DataSource	Upload trans data......(ZUPLOAD_TRANS_DATA)				
Data Type	Transaction Dat				
Source System	PSB Client 810(PSBCLNT810)				
Last Changed By		Date		Time	00:00:00

● Start Data Load Immediately

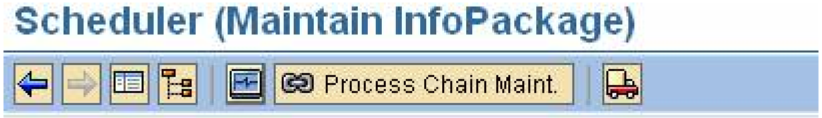
○ Start Later in Background

Job Name Prefix/Suffix BI_BTCH

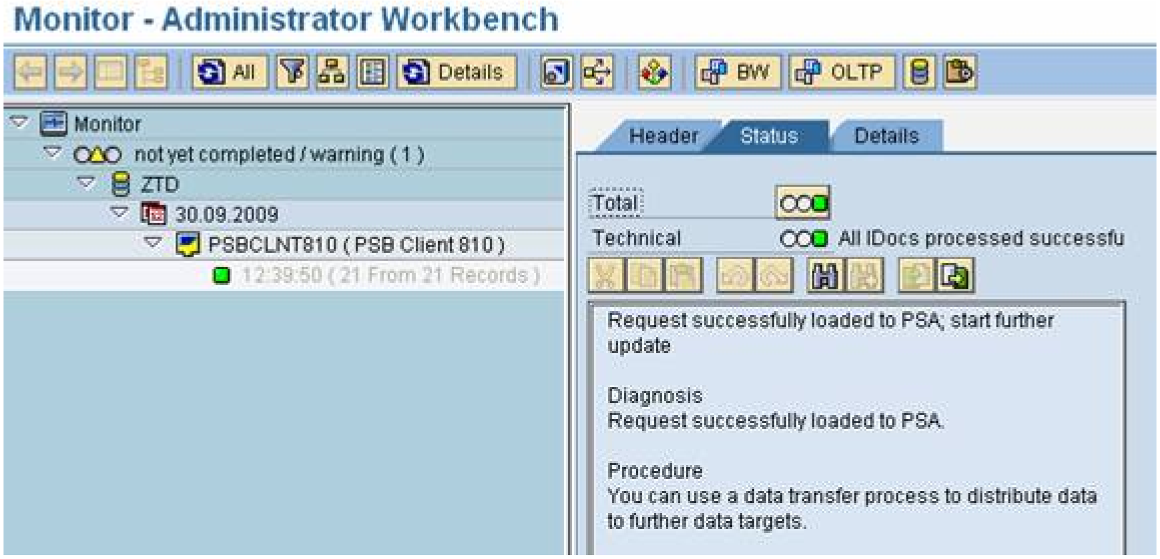
☐ Request Batch Process Runs Until All Data Has Been Updated in BW

Step 5: Click on icon "**Monitor**" on the toolbar to view if the data is successfully loaded to **PSA**

PSA: Persistent Staging Area is a staging area where you can load the data temporarily before loading to target



In the below screen shot, you can see status – “**Request successfully loaded to PSA**”



Monitor
not yet completed / warning (1)
ZTD
30.09.2009
PSBCLNT810 (PSB Client 810)
12:39:50 (21 From 21 Records)

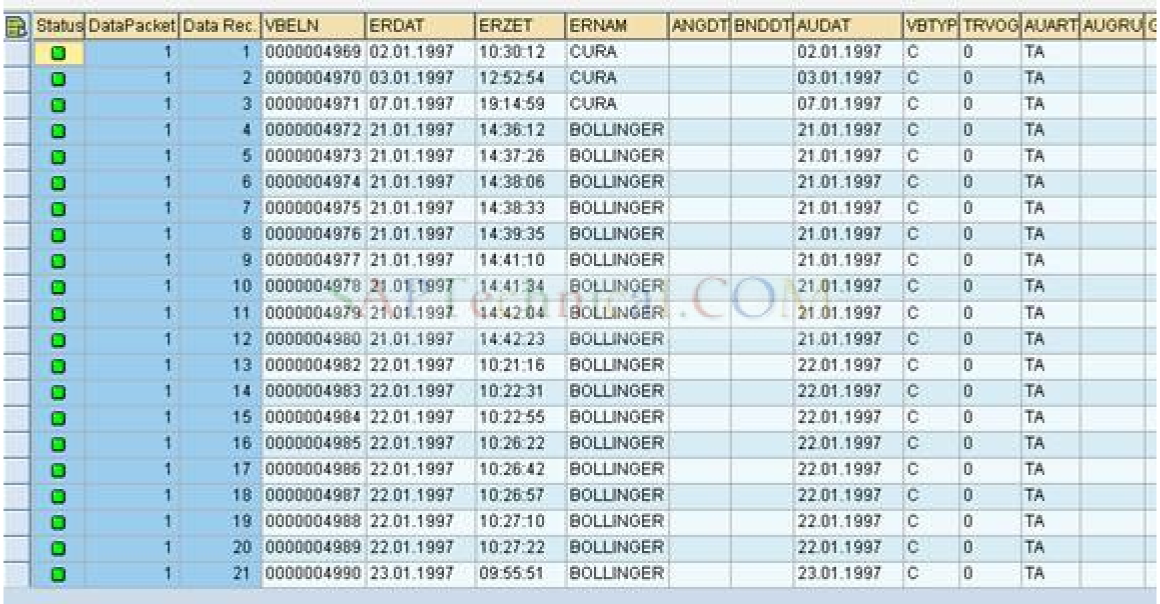
Total
Technical All IDocs processed successfu

Request successfully loaded to PSA; start further update

Diagnosis
Request successfully loaded to PSA.

Procedure
You can use a data transfer process to distribute data to further data targets.

Below Screen shot shows the data loaded to **PSA**

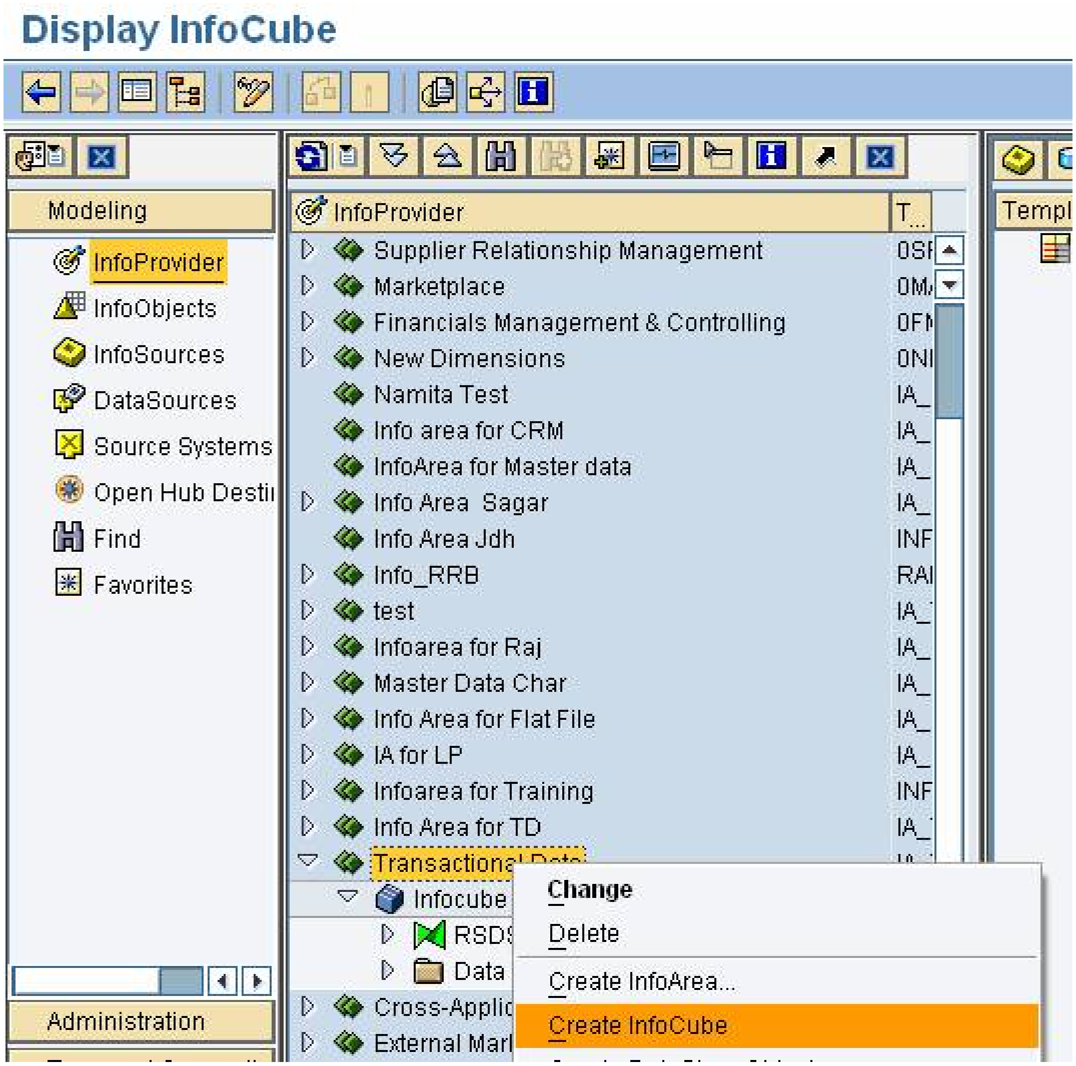


Status	DataPacket	Data Rec.	VBELN	ERDAT	ERZET	ERNAM	ANGDT	BNDDT	AUDAT	VBTYP	TRVOG	AUART	AUGRU	G
🟩	1	1	0000004969	02.01.1997	10:30:12	CURA			02.01.1997	C	0	TA		
🟩	1	2	0000004970	03.01.1997	12:52:54	CURA			03.01.1997	C	0	TA		
🟩	1	3	0000004971	07.01.1997	19:14:59	CURA			07.01.1997	C	0	TA		
🟩	1	4	0000004972	21.01.1997	14:36:12	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	5	0000004973	21.01.1997	14:37:26	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	6	0000004974	21.01.1997	14:38:06	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	7	0000004975	21.01.1997	14:38:33	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	8	0000004976	21.01.1997	14:39:35	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	9	0000004977	21.01.1997	14:41:10	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	10	0000004978	21.01.1997	14:41:34	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	11	0000004979	21.01.1997	14:42:04	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	12	0000004980	21.01.1997	14:42:23	BOLLINGER			21.01.1997	C	0	TA		
🟩	1	13	0000004982	22.01.1997	10:21:16	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	14	0000004983	22.01.1997	10:22:31	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	15	0000004984	22.01.1997	10:22:55	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	16	0000004985	22.01.1997	10:26:22	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	17	0000004986	22.01.1997	10:26:42	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	18	0000004987	22.01.1997	10:26:57	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	19	0000004988	22.01.1997	10:27:10	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	20	0000004989	22.01.1997	10:27:22	BOLLINGER			22.01.1997	C	0	TA		
🟩	1	21	0000004990	23.01.1997	09:55:51	BOLLINGER			23.01.1997	C	0	TA		

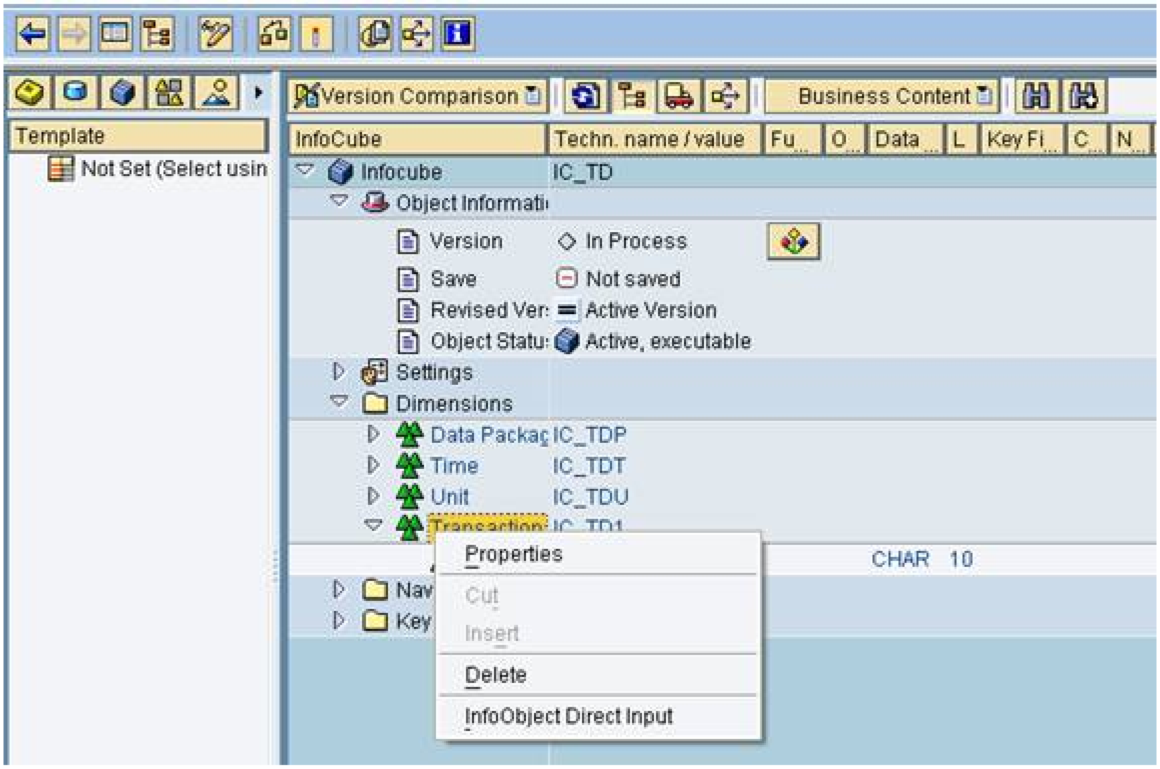
Steps to build the Target system

Step 6:

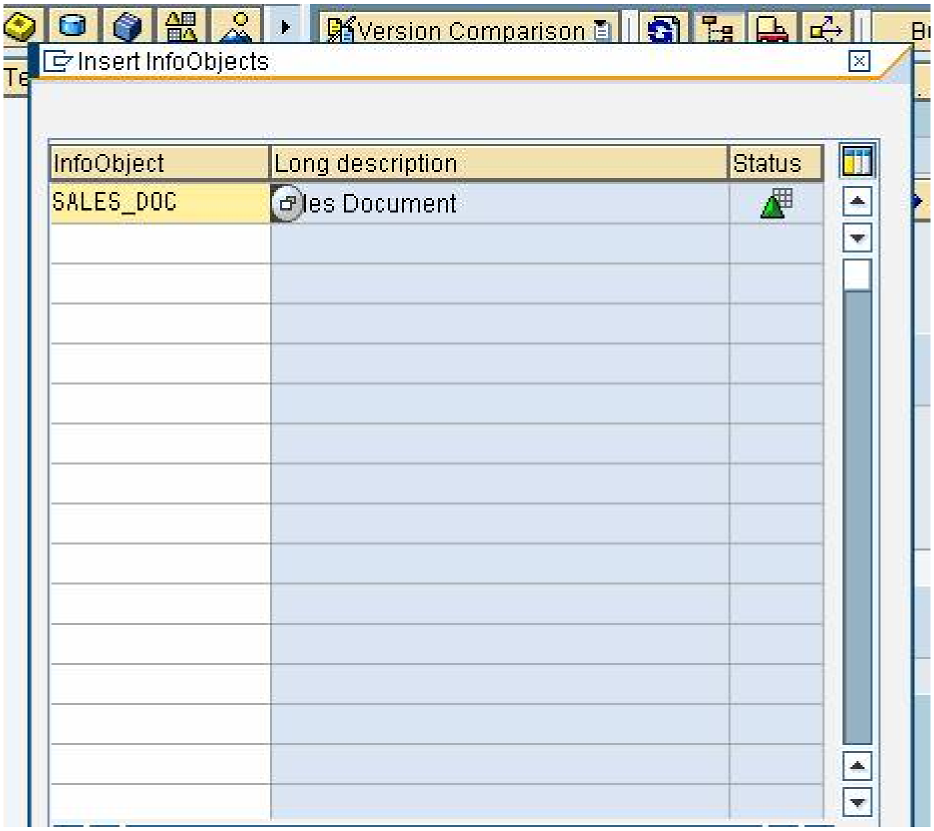
1. Rtclk on the **DataSource**, clk on "**Create InfoCube**"



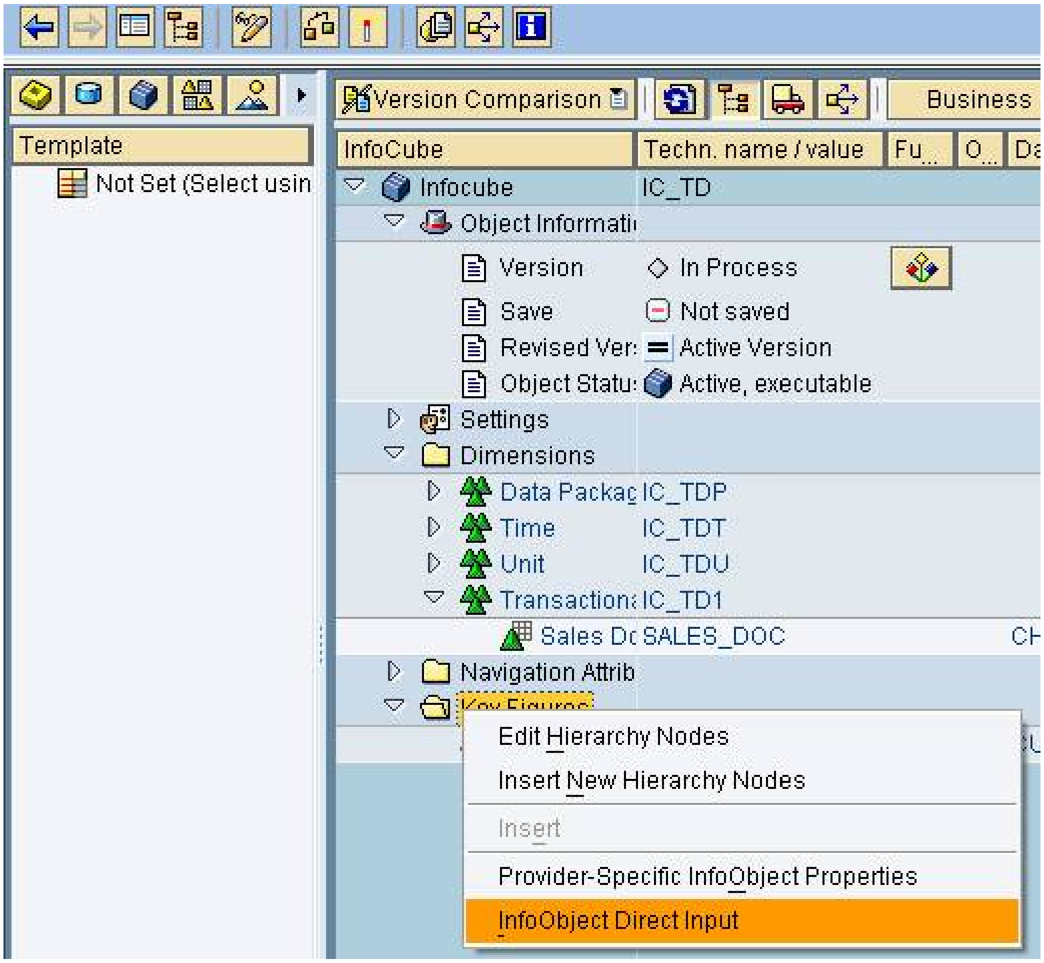
Step 7: Create Dimension table. Dimension table will have all "**Primary Keys**". In our example **VBELN – Sales Doc No** is the primary key



InfoCube	Techn. name / value	Fu...	O...	Data...	L	Key Fi...	C...	N...
Infocube	IC_TD							
Object Informati								
Version	In Process							
Save	Not saved							
Revised Ver:	Active Version							
Object Statu:	Active, executable							
Settings								
Dimensions								
Data Packag	IC_TDP							
Time	IC_TDT							
Unit	IC_TDU							
Transaction	IC_TD1							
				CHAR	10			
Nav								
Key								



InfoObject	Long description	Status
SALES_DOC	les Document	



Infocube IC_TD

Object Informati

Version ◇ In Process

Save Not saved

Revised Ver: Active Version

Object Statu: Active, executable

Settings

Dimensions

Data Packag IC_TDP

Time IC_TDT

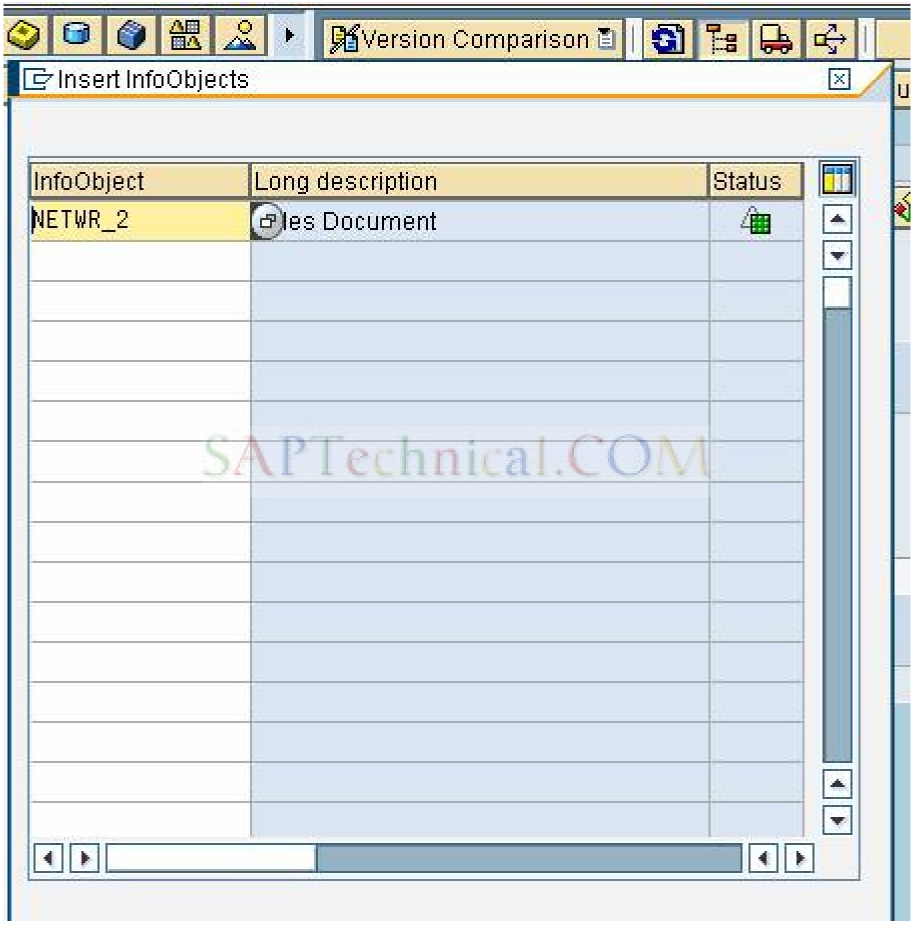
Unit IC_TDU

Transaction: IC_TD1

Sales Dc SALES_DOC CH

Navigation Attrib

Key Figures

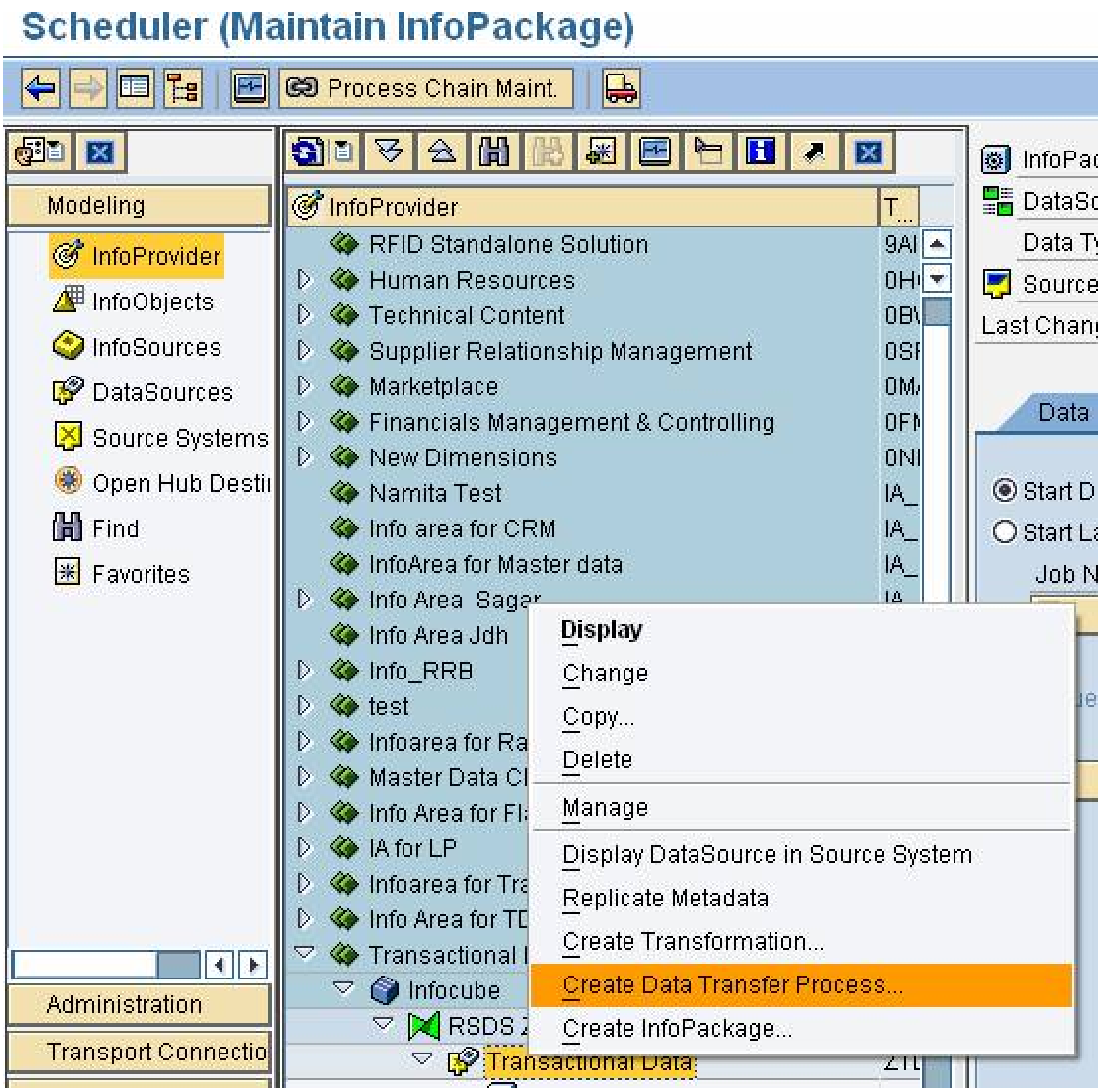


InfoObject	Long description	Status
NETWR_2	les Document	

Note: We need to define at least one **KeyFigure**

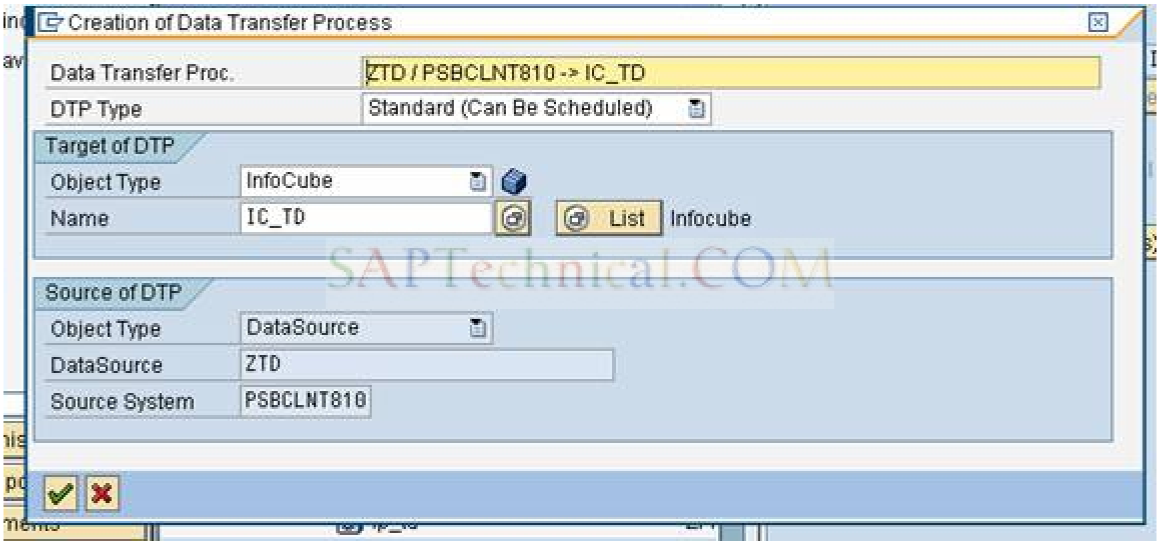
Step 8: Create DTP

DTP: DTP is used to transfer data from **PSA** to **Data target**. In our case the target is **InfoCube**.



The below screen shot shows the **Source** and **Target** DTP

<u>Note: </u>In our case the source is **DataSource** and Target is **InfoCube**



Creation of Data Transfer Process

Data Transfer Proc.	ZTD / PSBCLNT810 -> IC_TD
DTP Type	Standard (Can Be Scheduled)

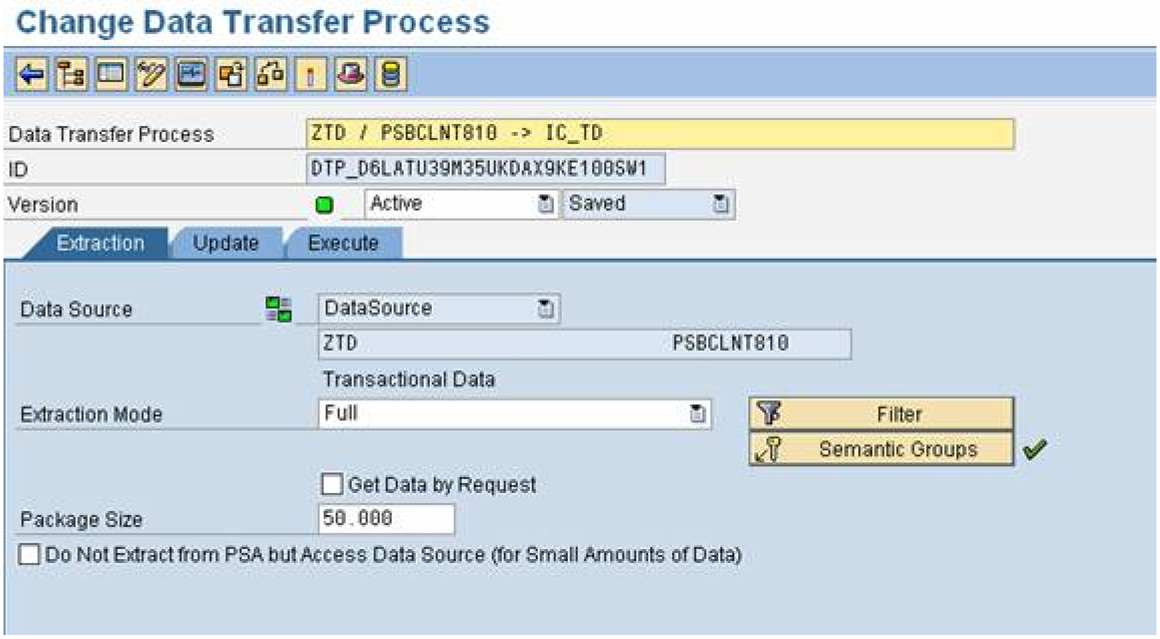
Target of DTP

Object Type	InfoCube
Name	IC_TD

Source of DTP

Object Type	DataSource
DataSource	ZTD
Source System	PSBCLNT810

Select Extraction Mode as “**Full**” as we are loading the data for the first time



Change Data Transfer Process

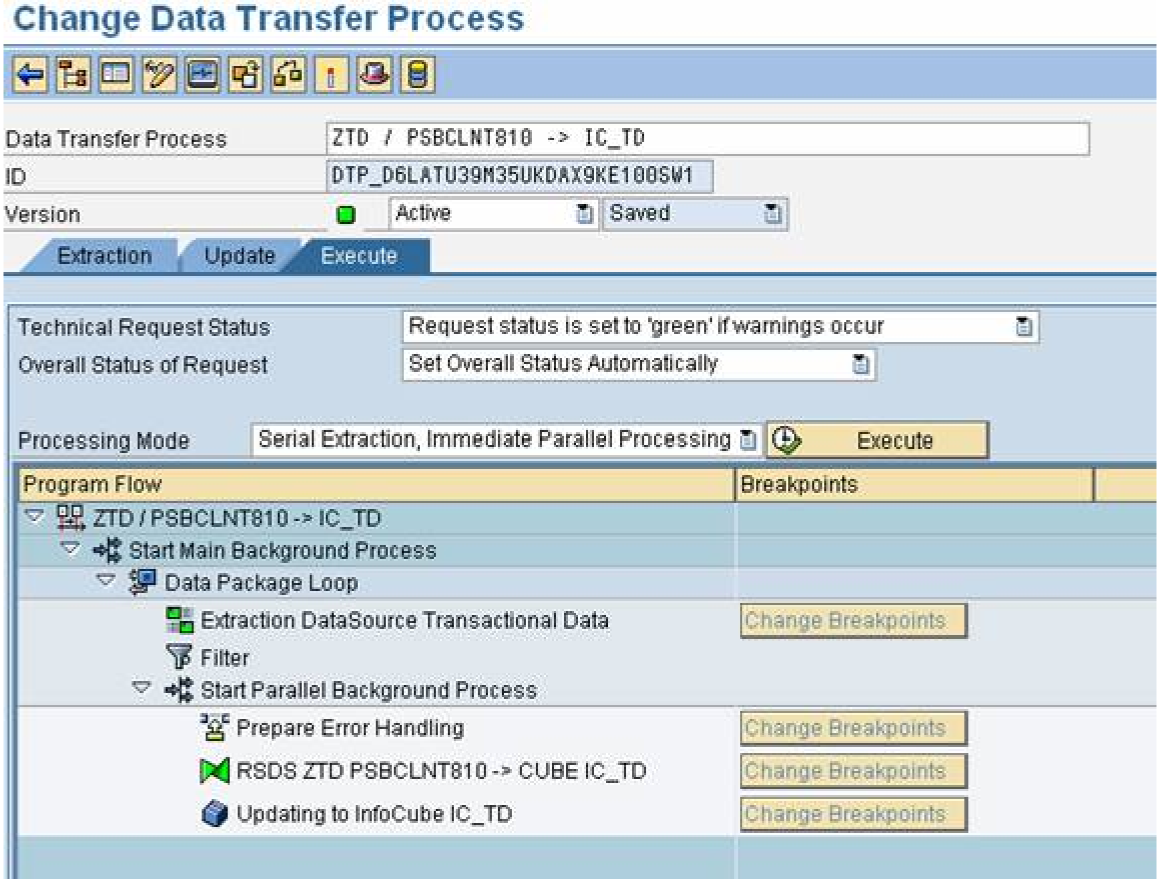
Data Transfer Process	ZTD / PSBCLNT810 -> IC_TD
ID	DTP_D6LATU39M35UKDAX9KE100SW1
Version	Active Saved

Data Source	DataSource ZTD PSBCLNT810 Transactional Data
Extraction Mode	Full
	□ Get Data by Request
Package Size	50.000

□ Do Not Extract from PSA but Access Data Source (for Small Amounts of Data)

Step 9:

1. Save and Activate
2. Go to “**Execute**” tab and click on Execute



Change Data Transfer Process

Data Transfer Process: ZTD / PSBCLNT810 -> IC_TD

ID: DTP_D6LATU39M35UKDAX9KE100SW1

Version: 🟩 Active Saved

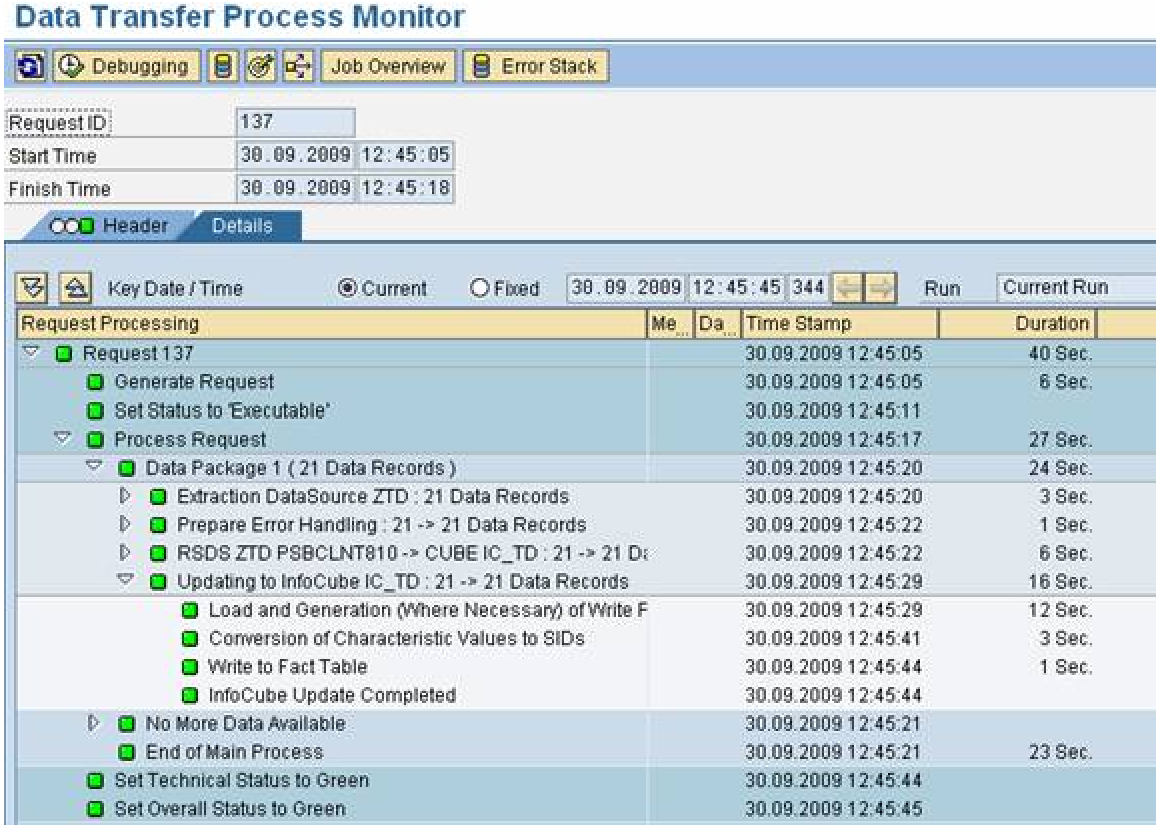
Technical Request Status: Request status is set to 'green' if warnings occur

Overall Status of Request: Set Overall Status Automatically

Processing Mode: Serial Extraction, Immediate Parallel Processing

Program Flow	Breakpoints
ZTD / PSBCLNT810 -> IC_TD	
Start Main Background Process	
Data Package Loop	
Extraction DataSource Transactional Data	Change Breakpoints
Filter	
Start Parallel Background Process	
Prepare Error Handling	Change Breakpoints
RSDS ZTD PSBCLNT810 -> CUBE IC_TD	Change Breakpoints
Updating to InfoCube IC_TD	Change Breakpoints

Below screen shot shows the **DTP monitor**. Green indicates that the data is successfully loaded to the target system



Data Transfer Process Monitor

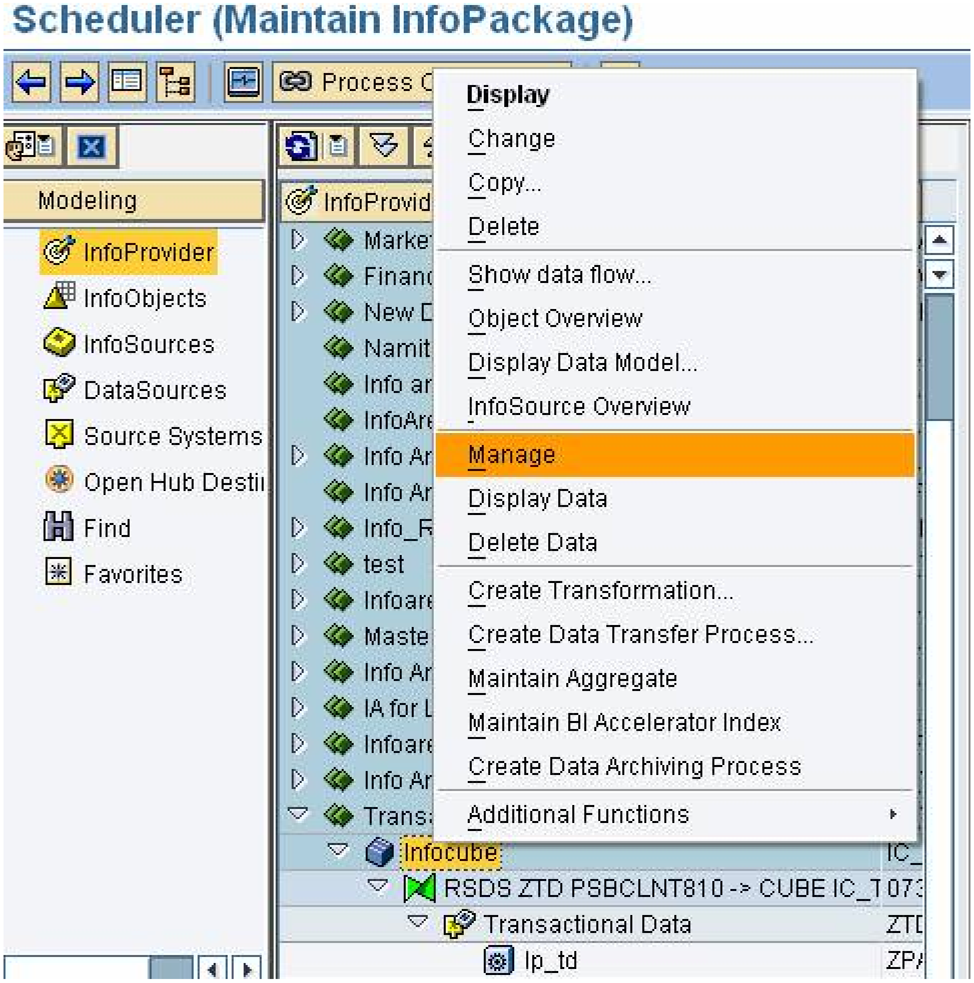
Request ID	137
Start Time	30.09.2009 12:45:05
Finish Time	30.09.2009 12:45:18

Key Date / Time ● Current ○ Fixed 30.09.2009 12:45:45 344 Run Current Run

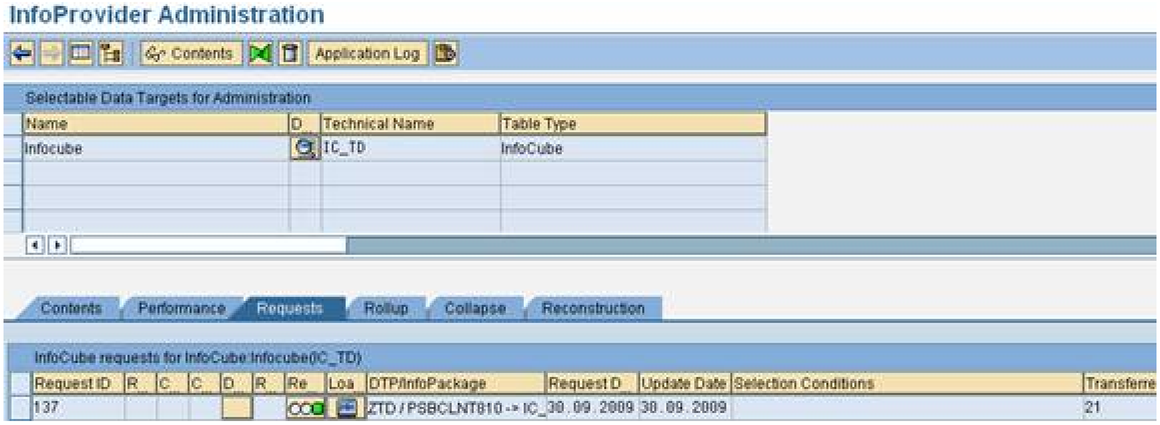
Request Processing	Me...	Da...	Time Stamp	Duration
🟩 Request 137			30.09.2009 12:45:05	40 Sec.
🟩 Generate Request			30.09.2009 12:45:05	6 Sec.
🟩 Set Status to 'Executable'			30.09.2009 12:45:11	
🟩 Process Request			30.09.2009 12:45:17	27 Sec.
🟩 Data Package 1 (21 Data Records)			30.09.2009 12:45:20	24 Sec.
🟩 Extraction DataSource ZTD : 21 Data Records			30.09.2009 12:45:20	3 Sec.
🟩 Prepare Error Handling : 21 -> 21 Data Records			30.09.2009 12:45:22	1 Sec.
🟩 RSDS ZTD PSBCLNT810 -> CUBE IC_TD : 21 -> 21 Da			30.09.2009 12:45:22	6 Sec.
🟩 Updating to InfoCube IC_TD : 21 -> 21 Data Records			30.09.2009 12:45:29	16 Sec.
🟩 Load and Generation (Where Necessary) of Write F			30.09.2009 12:45:29	12 Sec.
🟩 Conversion of Characteristic Values to SIDs			30.09.2009 12:45:41	3 Sec.
🟩 Write to Fact Table			30.09.2009 12:45:44	1 Sec.
🟩 InfoCube Update Completed			30.09.2009 12:45:44	
🟩 No More Data Available			30.09.2009 12:45:21	
🟩 End of Main Process			30.09.2009 12:45:21	23 Sec.
🟩 Set Technical Status to Green			30.09.2009 12:45:44	
🟩 Set Overall Status to Green			30.09.2009 12:45:45	

Step 10: To see if the data is successfully loaded to target system,

Rtclk on the **Infocube** => **Manage**



Note: When you create DTP a **Request Id** will be generated as shown in the below screen shot



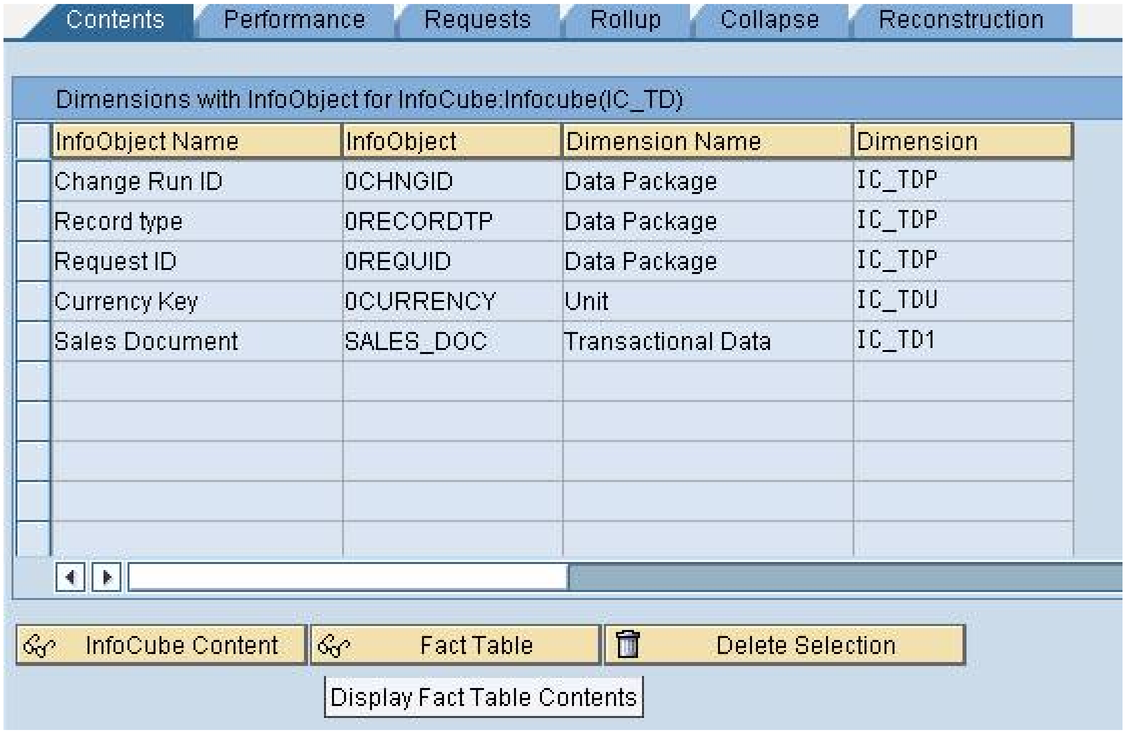
Selectable Data Targets for Administration

Name	D	Technical Name	Table Type
Infocube		IC_TD	InfoCube

InfoCube requests for InfoCube Infocube(IC_TD)

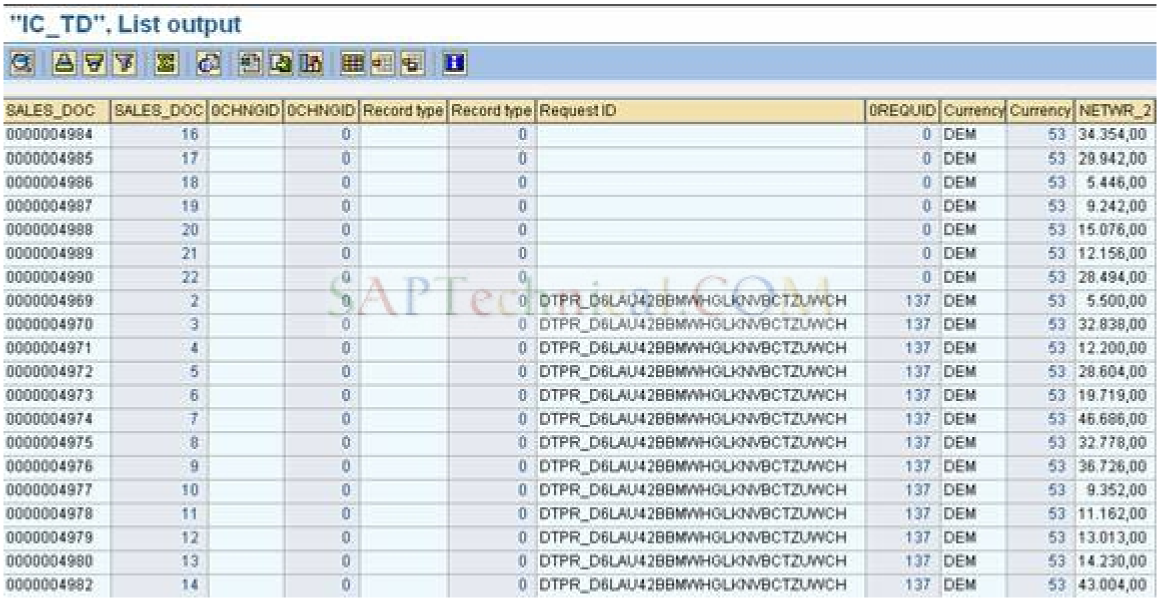
Request ID	R	C	C	D	R	Re	Loa	DTP/InfoPackage	Request D	Update Date	Selection Conditions	Transferre
137								ZTD / PSBCLNT810 -> IC_	30.09.2009	30.09.2009		21

Step 11: Go to “**Contents”** => “**InfoCube Content**” to see the output



Dimensions with InfoObject for InfoCube:Infocube(IC_TD)

InfoObject Name	InfoObject	Dimension Name	Dimension
Change Run ID	0CHNGID	Data Package	IC_TDP
Record type	0RECORDTP	Data Package	IC_TDP
Request ID	0REQUID	Data Package	IC_TDP
Currency Key	0CURRENCY	Unit	IC_TDU
Sales Document	SALES_DOC	Transactional Data	IC_TD1



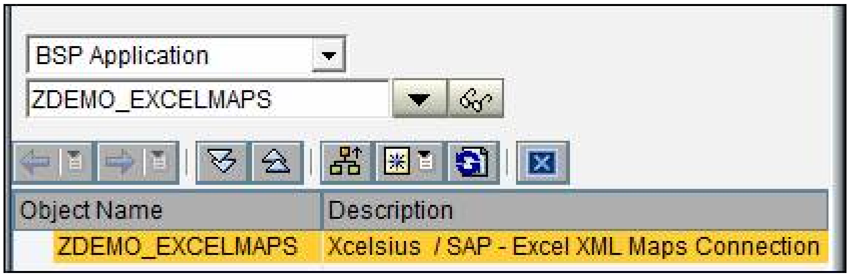
"IC_TD", List output

SALES_DOC	SALES_DOC	0CHNGID	0CHNGID	Record type	Record type	Request ID	0REQUID	Currency	Currency	NETWR_2
0000004984	16		0		0		0	DEM	53	34.354,00
0000004985	17		0		0		0	DEM	53	29.942,00
0000004986	18		0		0		0	DEM	53	5.446,00
0000004987	19		0		0		0	DEM	53	9.242,00
0000004988	20		0		0		0	DEM	53	15.076,00
0000004989	21		0		0		0	DEM	53	12.156,00
0000004990	22		0		0		0	DEM	53	28.494,00
0000004969	2		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	5.500,00
0000004970	3		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	32.838,00
0000004971	4		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	12.200,00
0000004972	5		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	28.604,00
0000004973	6		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	19.719,00
0000004974	7		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	46.686,00
0000004975	8		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	32.778,00
0000004976	9		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	36.726,00
0000004977	10		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	9.352,00
0000004978	11		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	11.162,00
0000004979	12		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	13.013,00
0000004980	13		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	14.230,00
0000004982	14		0		0	DTPR_D6LAU42BBMWHGLKNVBCTZUWCH	137	DEM	53	43.004,00

Integrating SAP data (R/3 or BW Data) with the BO-Xcelsius (Business Objects) using BSP

Part 1: SAP BSP + BO-Xcelsius (Excel XML Maps Connection)

Create a BSP application using SE80 transaction (**ZDEMO_EXCELMAPS**)



BSP Application

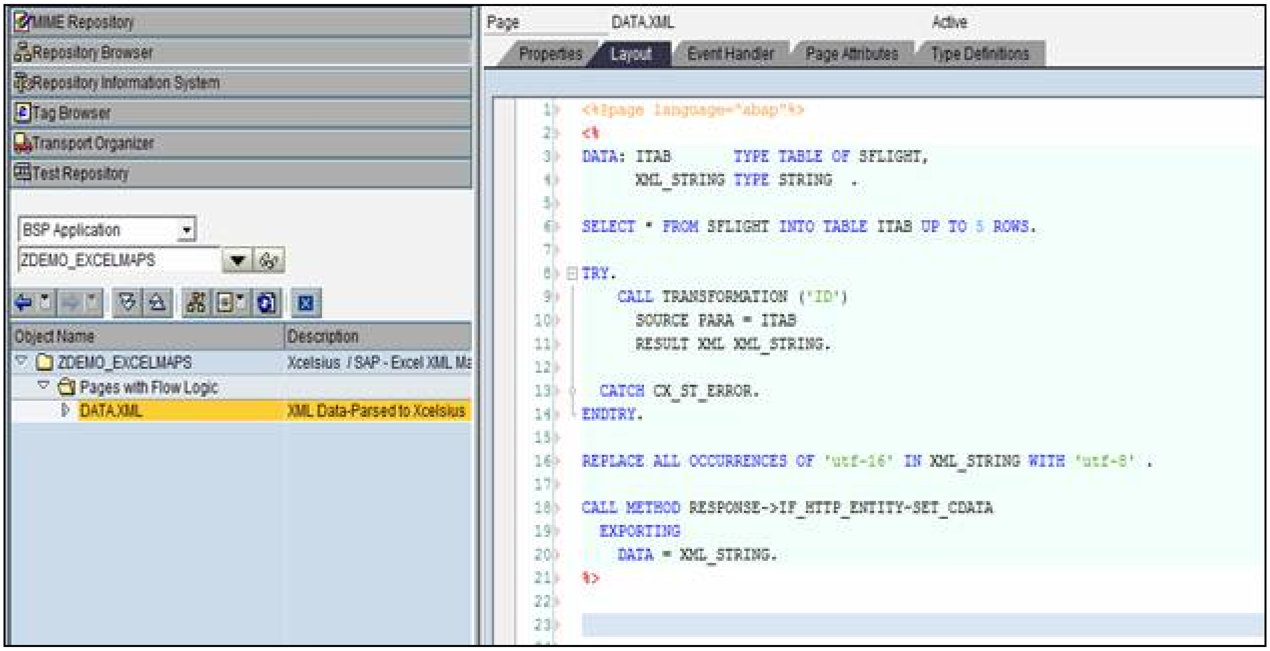
ZDEMO_EXCELMAPS

Object Name	Description
ZDEMO_EXCELMAPS	Xcelsius / SAP - Excel XML Maps Connection

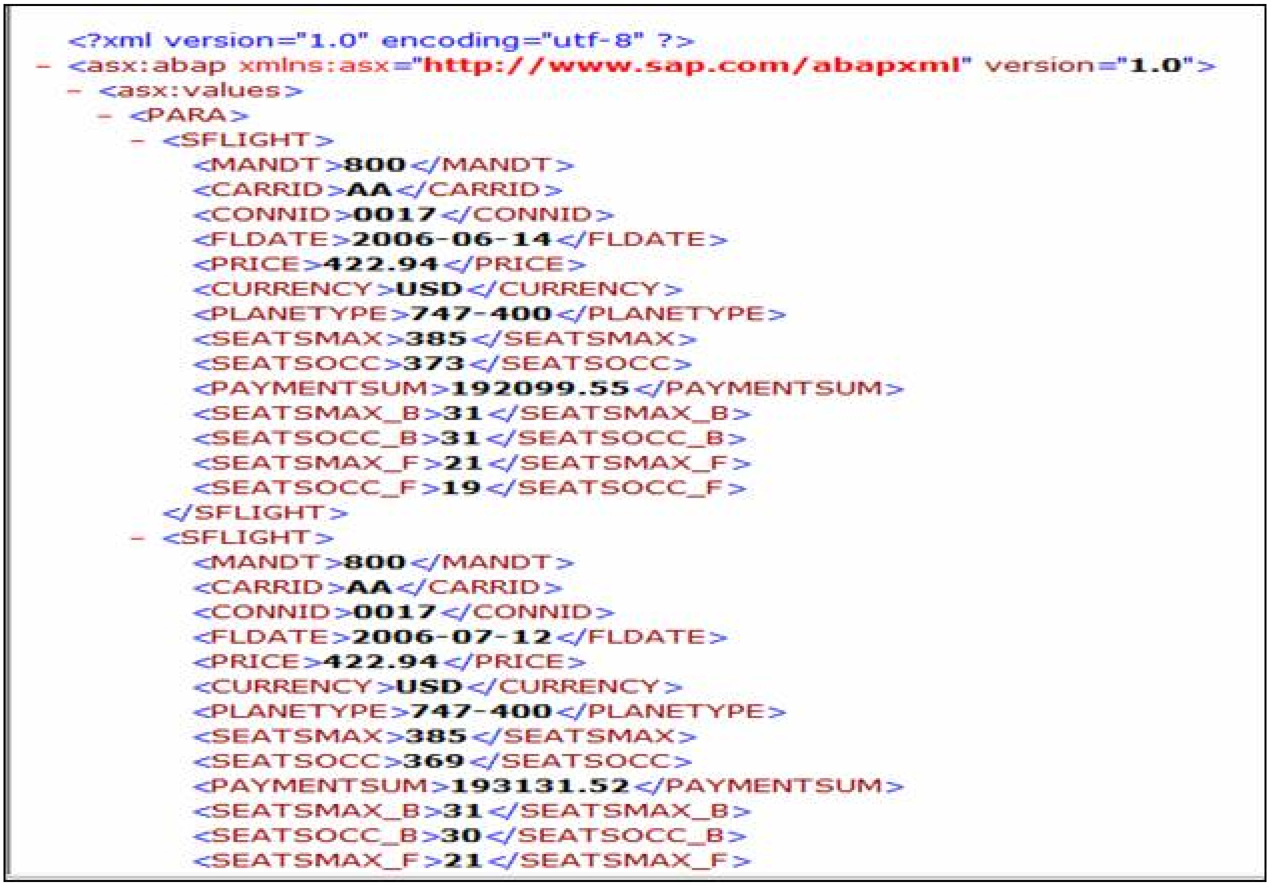
Create a View (PWFL)-**DATA.XML** which will have the XML data. This data will be acting as the main source for the Xcelsius file which is going to be created in the further steps. Use the below code in the layout of the XML view.

```
<%@page language="abap"%>
<%                                                  DATA: ITAB      TYPE TABLE OF SFLIGHT,
                                                             XML_STRING TYPE STRING  .

  SELECT * FROM SFLIGHT INTO TABLE ITAB UP TO 5 ROWS.

TRY.
                                                            CALL TRANSFORMATION ('ID')
                                                                  SOURCE PARA = ITAB
                                                              RESULT XML XML_STRING.

    CATCH CX_ST_ERROR.
  ENDTRY.

  REPLACE ALL OCCURRENCES OF 'utf-16' IN XML_STRING WITH 'utf-8' .

  CALL METHOD RESPONSE->IF_HTTP_ENTITY~SET_CDATA
                                                                          EXPORTING
                                                                   DATA = XML_STRING.
%>
```

The above code is used to generate the XML data. The internal table returned from the select statement will be presented to the standard “**CALL TRANSFORMATION (ID)** “to generate the XML. The XML format may be either in UTF-8 or UTF-16 or anything. Based on the setting and server configurations the versions have to be set. The XML file has to be tested and the generated XML will be saved locally in order to design the Chart.

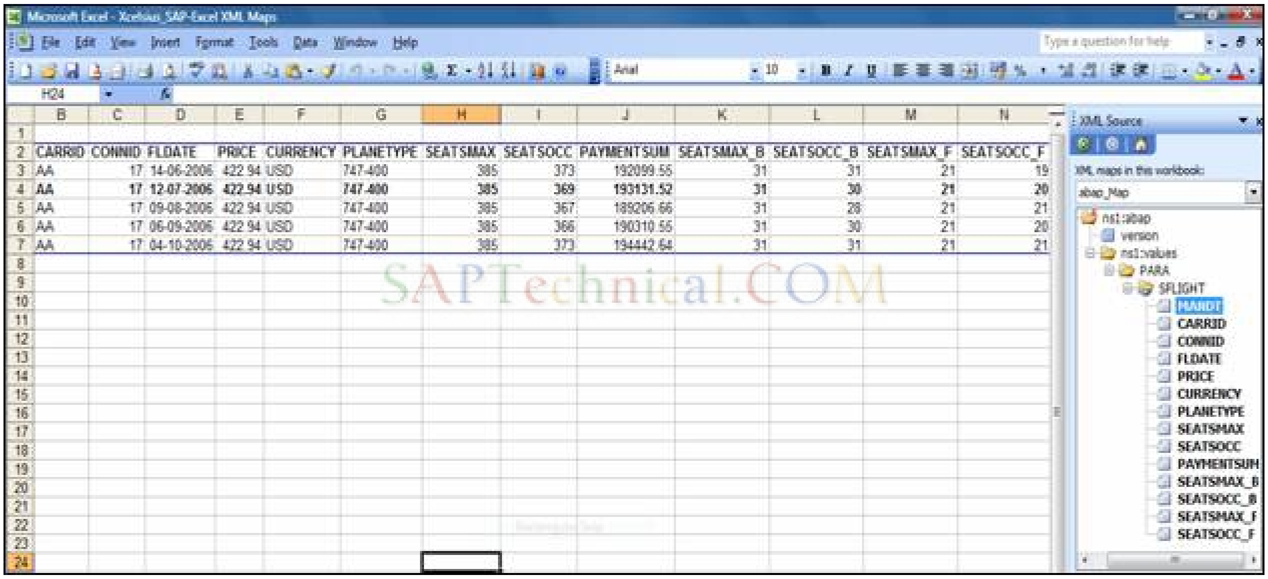


```
<%@page language="abap"%>
<%
DATA: ITAB       TYPE TABLE OF SFLIGHT,
      XML_STRING TYPE STRING  .

SELECT * FROM SFLIGHT INTO TABLE ITAB UP TO 5 ROWS.

TRY.
    CALL TRANSFORMATION ('ID')
      SOURCE PARA = ITAB
      RESULT XML XML_STRING.

  CATCH CX_ST_ERROR.
ENDTRY.

REPLACE ALL OCCURRENCES OF 'utf-16' IN XML_STRING WITH 'utf-8' .

CALL METHOD RESPONSE->IF_HTTP_ENTITY-SET_CDATA
  EXPORTING
    DATA = XML_STRING.
%>
```



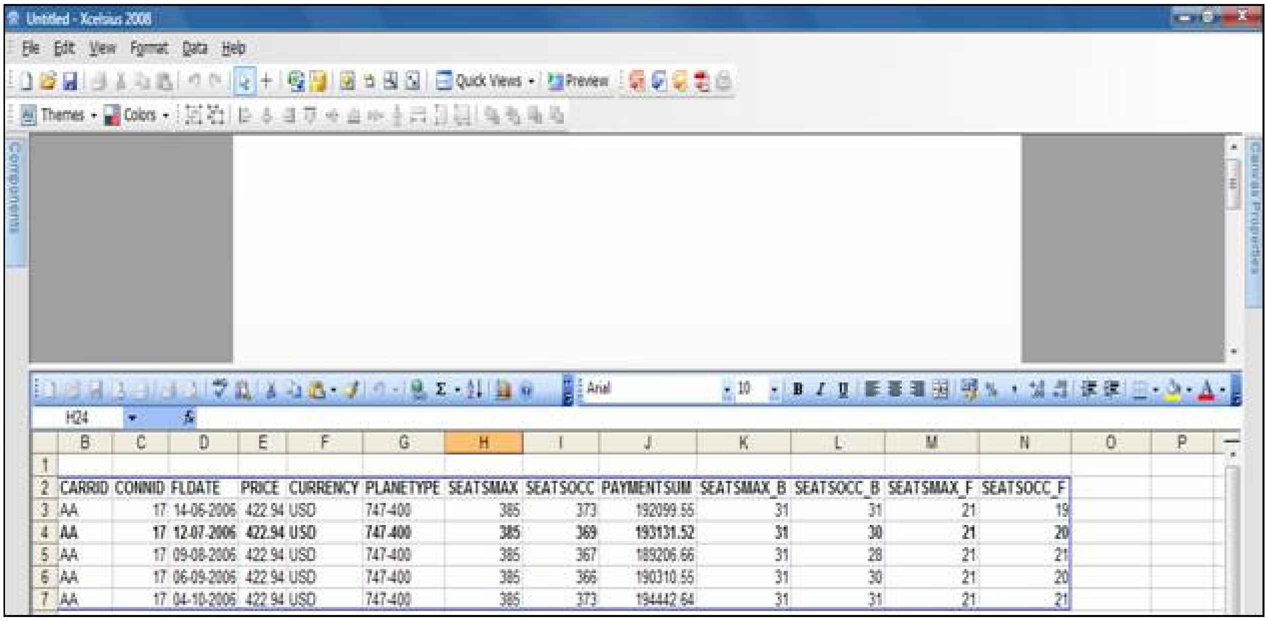
```
  <?xml version="1.0" encoding="utf-8" ?>
- <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  - <asx:values>
    - <PARA>
      - <SFLIGHT>
          <MANDT>800</MANDT>
          <CARRID>AA</CARRID>
          <CONNID>0017</CONNID>
          <FLDATE>2006-06-14</FLDATE>
          <PRICE>422.94</PRICE>
          <CURRENCY>USD</CURRENCY>
          <PLANETYPE>747-400</PLANETYPE>
          <SEATSMAX>385</SEATSMAX>
          <SEATSOCC>373</SEATSOCC>
          <PAYMENTSUM>192099.55</PAYMENTSUM>
          <SEATSMAX_B>31</SEATSMAX_B>
          <SEATSOCC_B>31</SEATSOCC_B>
          <SEATSMAX_F>21</SEATSMAX_F>
          <SEATSOCC_F>19</SEATSOCC_F>
        </SFLIGHT>
      - <SFLIGHT>
          <MANDT>800</MANDT>
          <CARRID>AA</CARRID>
          <CONNID>0017</CONNID>
          <FLDATE>2006-07-12</FLDATE>
          <PRICE>422.94</PRICE>
          <CURRENCY>USD</CURRENCY>
          <PLANETYPE>747-400</PLANETYPE>
          <SEATSMAX>385</SEATSMAX>
          <SEATSOCC>369</SEATSOCC>
          <PAYMENTSUM>193131.52</PAYMENTSUM>
          <SEATSMAX_B>31</SEATSMAX_B>
          <SEATSOCC_B>30</SEATSOCC_B>
          <SEATSMAX_F>21</SEATSMAX_F>
```

Now an Excel file is imported with the XML data saved locally. The data returned from the FM will be presented in the excel file and the file will be saved locally.

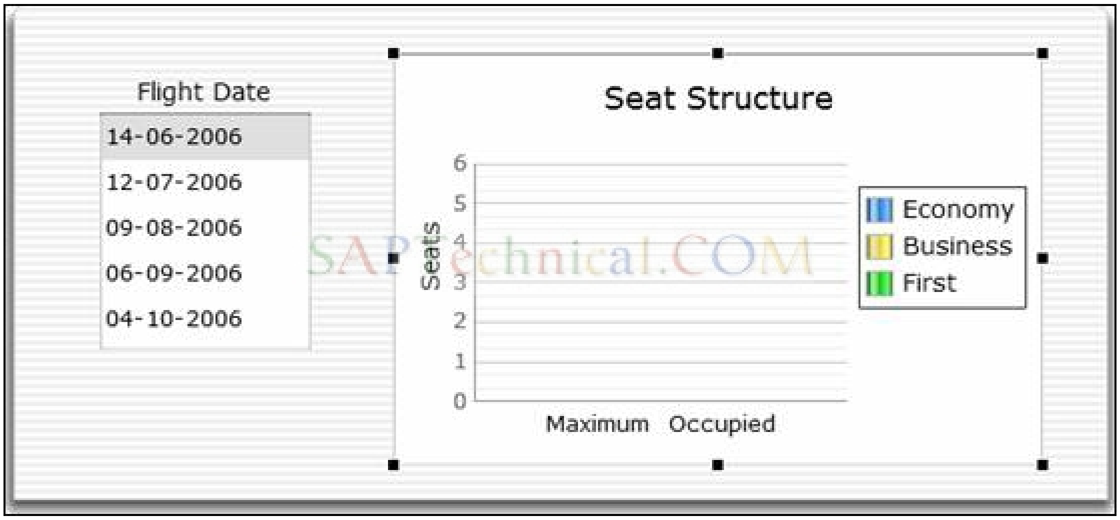


CARRID	CONNID	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F	SEATSOCC_F
AA	17	14-06-2006	422.94	USD	747-400	385	373	192099.55	31	31	21	19
AA	**17**	**12-07-2006**	**422.94**	**USD**	**747-400**	**385**	**369**	**193131.52**	**31**	**30**	**21**	**20**
AA	17	09-08-2006	422.94	USD	747-400	385	367	189206.66	31	28	21	21
AA	17	06-09-2006	422.94	USD	747-400	385	366	190310.55	31	30	21	20
AA	17	04-10-2006	422.94	USD	747-400	385	373	194442.64	31	31	21	21

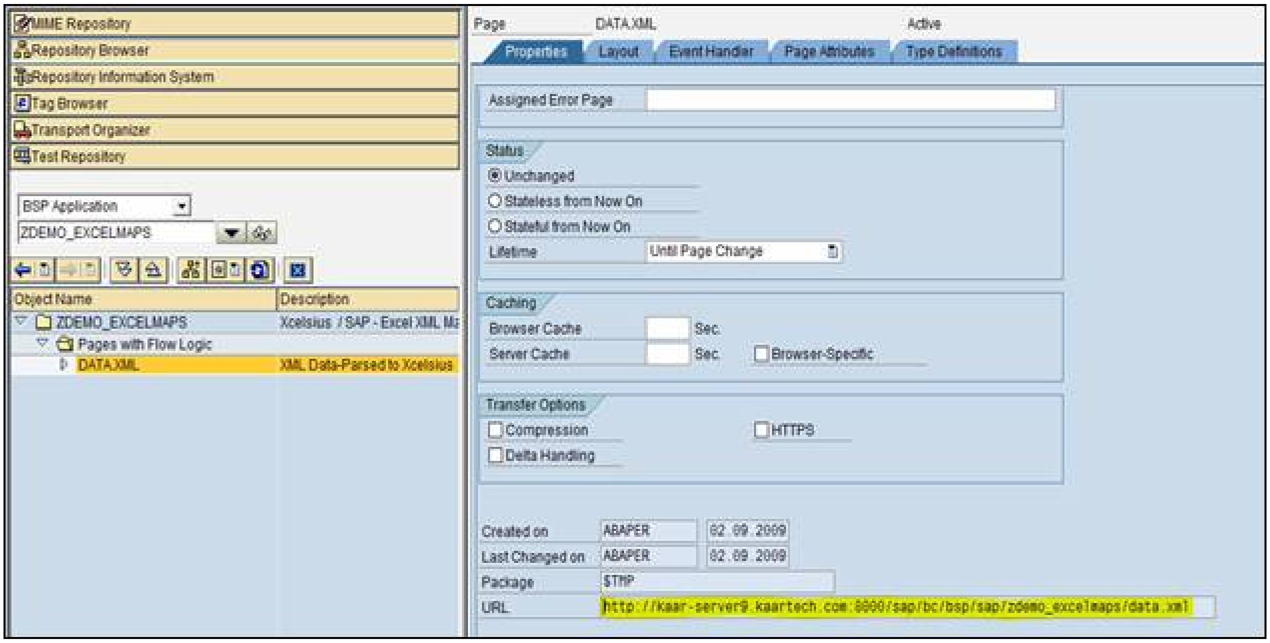
Now the BO-Xcelsius will be opened and imported with excel file which is saved in the previous step. The data will be feed into the excel sheet of the Xcelsius and the report is designed by using several components available.



CARRID	CONNID	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F	SEATSOCC_F
AA	17	14-06-2006	422.94	USD	747-400	385	373	192099.55	31	31	21	19
AA	**17**	**12-07-2006**	**422.94**	**USD**	**747-400**	**385**	**369**	**193131.52**	**31**	**30**	**21**	**20**
AA	17	09-08-2006	422.94	USD	747-400	385	367	189206.66	31	28	21	21
AA	17	06-09-2006	422.94	USD	747-400	385	366	190310.55	31	30	21	20
AA	17	04-10-2006	422.94	USD	747-400	385	373	194442.64	31	31	21	21



Now the connectivity has to be made by opening the **Manage Data Connections - > Add - > Excel XML Maps**. Name the connection and mention the XML data URL. The view represented in BSP application for the XML file has the URL and will be represented.



BSP Application
ZDEMO_EXCELMAPS

Object Name	Description
ZDEMO_EXCELMAPS	Xcelsius / SAP - Excel XML Ma
Pages with Flow Logic	
DATA.XML	XML Data-Parsed to Xcelsius

Page DATA.XML Active

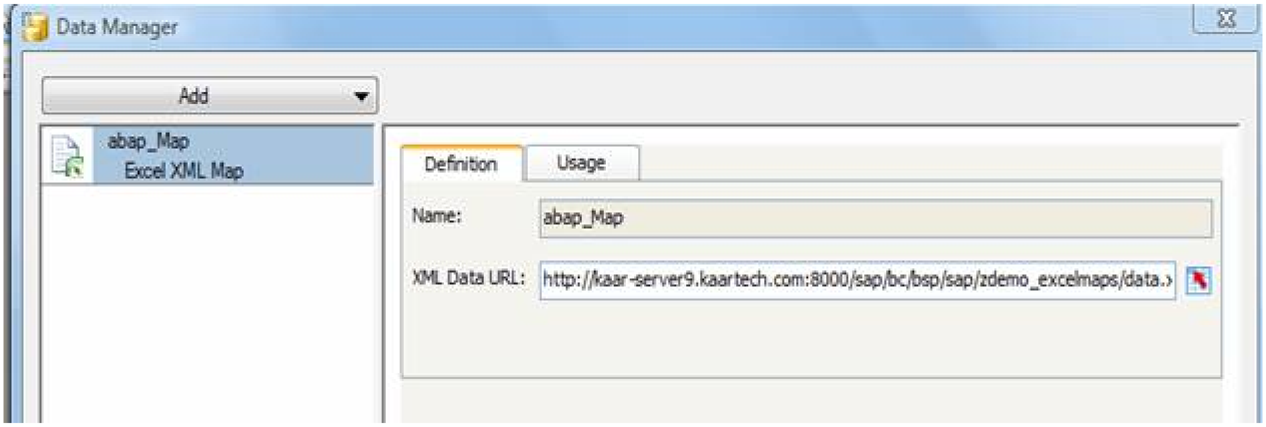
Assigned Error Page

Status
● Unchanged
○ Stateless from Now On
○ Stateful from Now On
Lifetime Until Page Change

Caching
Browser Cache Sec.
Server Cache Sec. □ Browser-Specific

Transfer Options
□ Compression □ HTTPS
□ Delta Handling

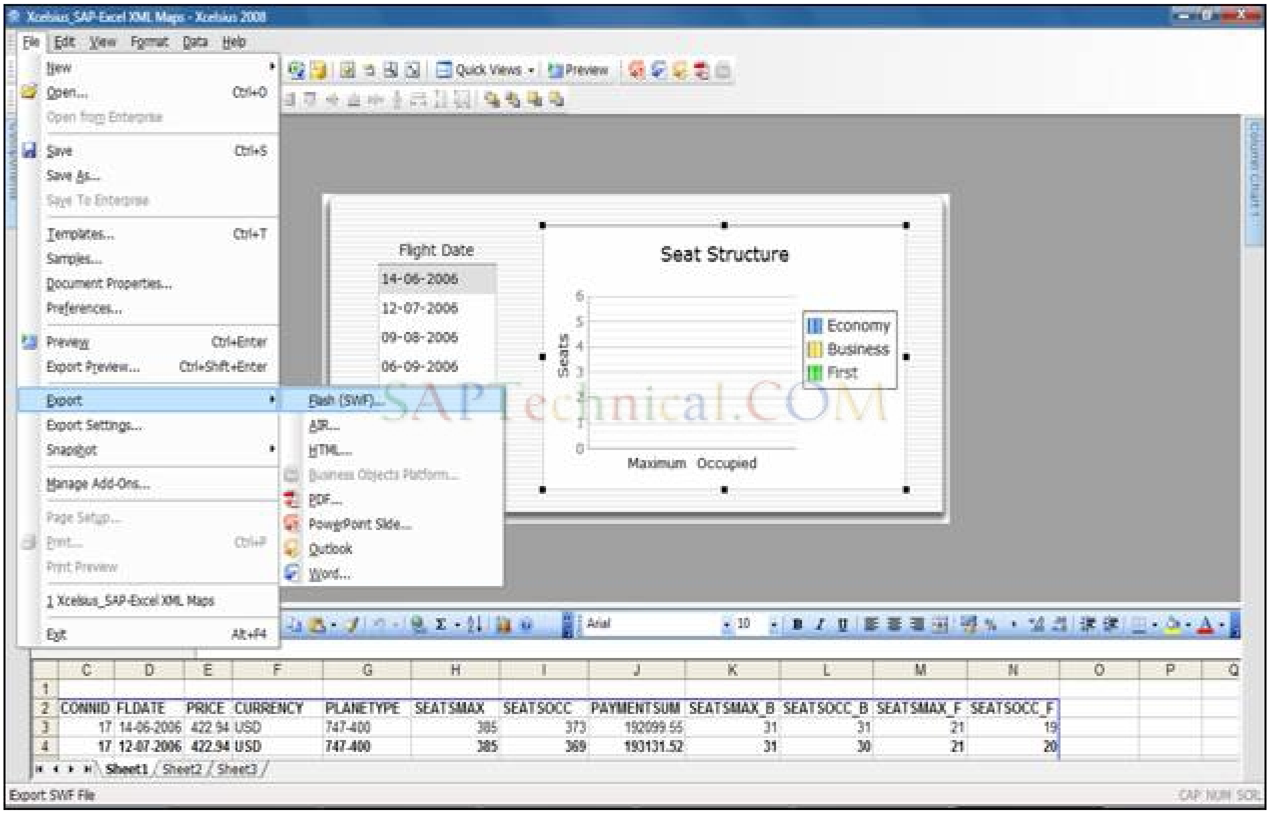
Created on ABAPER 02.09.2009
Last Changed on ABAPER 02.09.2009
Package $TMP
URL http://kaar-server9.kaartech.com:8000/sap/bc/bsp/sap/zdemo_excelmaps/data.xml



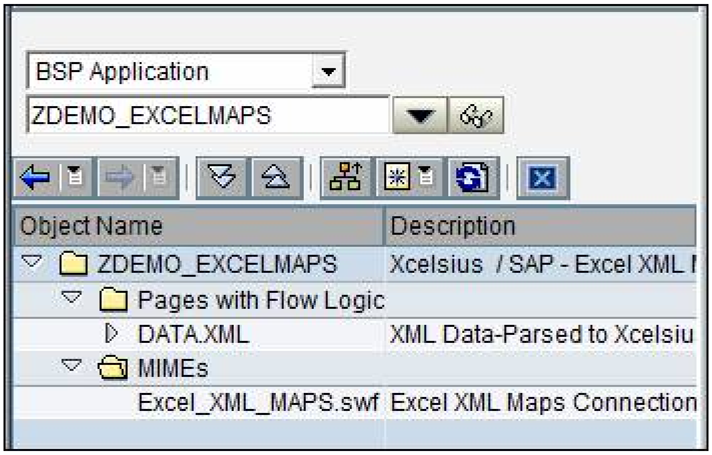
Name: abap_Map

XML Data URL: http://kaar-server9.kaartech.com:8000/sap/bc/bsp/sap/zdemo_excelmaps/data.x

The designed Xcelsius file will be saved in XLF format locally which will be then exported as SWF file. The SWF file will be imported as MIME object into the BSP application.



	C	D	E	F	G	H	I	J	K	L	M	N
1												
2	CONNID	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F	SEATSOCC_F
3	17	14-06-2006	422.94	USD	747-400	385	373	192099.55	31	31	21	19
4	17	12-07-2006	422.94	USD	747-400	385	369	193131.52	31	30	21	20



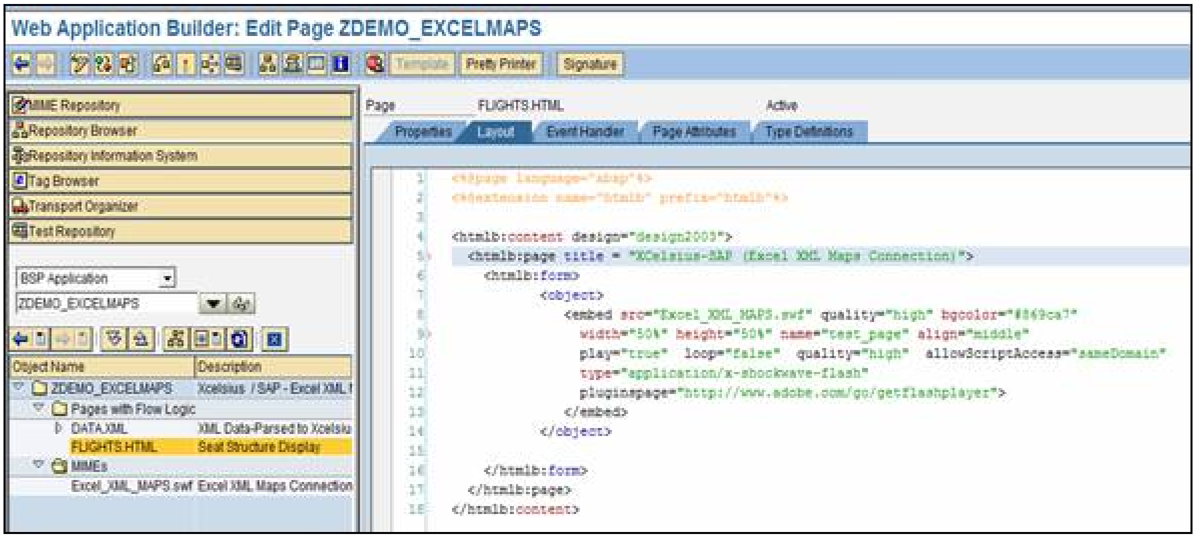
BSP Application

ZDEMO_EXCELMAPS

Object Name	Description
ZDEMO_EXCELMAPS	Xcelsius / SAP - Excel XML I
Pages with Flow Logic	
DATA.XML	XML Data-Parsed to Xcelsiu
MIMEs	
Excel_XML_MAPS.swf	Excel XML Maps Connection

Another **HTML** view will be created inside the same BSP application in order to display the **MIME** object. Apply the below code in the layout of **FLIGHTS.HTML** view.

```
<%@page language="abap"%>
<%@extension name="htmlb" prefix="htmlb"%>

<htmlb:content design="design2003">
  <htmlb:page title = "XCelsius-SAP (Excel XMl Maps Connection)">
    <htmlb:form>
           <object>
              <embed src="Excel_XML_MAPS.swf" quality="high"

                                          bgcolor="#869ca7" width="50%" height="50%"

                                          name="test_page" align="middle"     play="true"  loop="false"  quality="high"

                                          allowScriptAccess="sameDomain"

                                          type="application/x-shockwave-flash"

                pluginspage="http://www.adobe.com/go/getflashplayer">
              </embed>
           </object>

    </htmlb:form>
  </htmlb:page>
</htmlb:content>
```



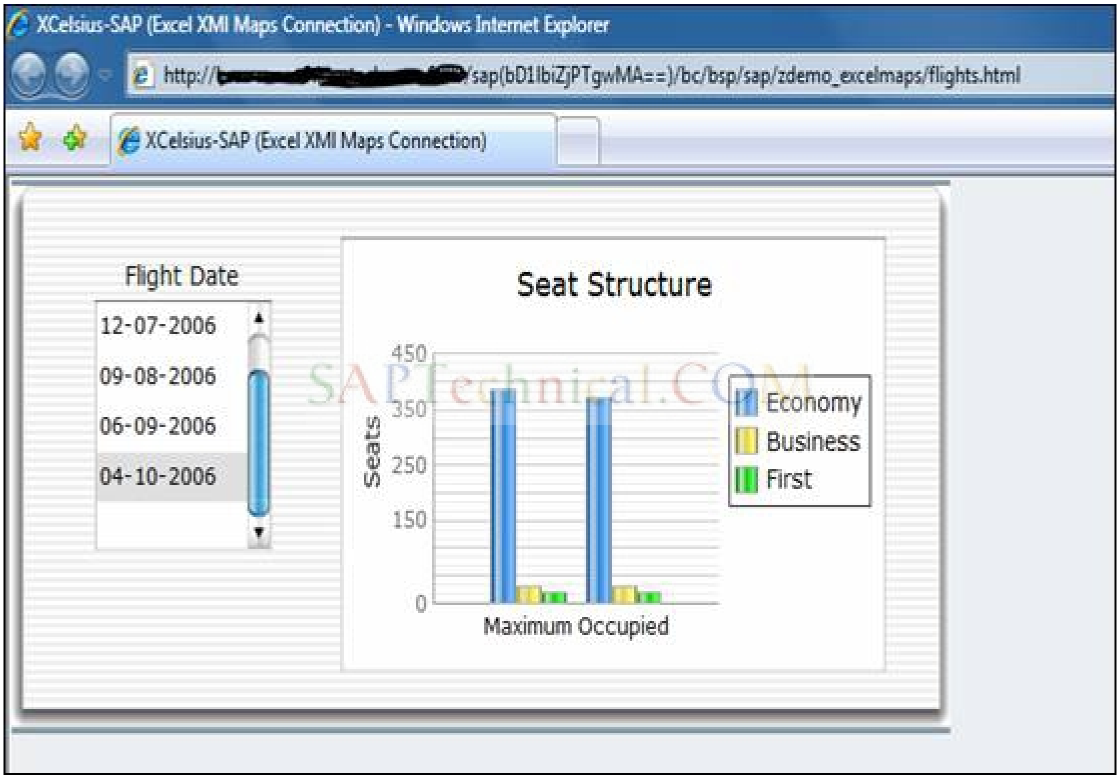
Page FLIGHTS.HTML Active

Object Name	Description
ZDEMO_EXCELMAPS	Xcelsius / SAP - Excel XML I
Pages with Flow Logic	
DATA.XML	XML Data-Parsed to Xcelsiu
FLIGHTS.HTML	Seat Structure Display
MIMEs	
Excel_XML_MAPS.swf	Excel XML Maps Connection

```
<%@page language="abap"%>
<%@extension name="htmlb" prefix="htmlb"%>

<htmlb:content design="design2003">
  <htmlb:page title = "XCelsius-SAP (Excel XML Maps Connection)">
    <htmlb:form>
            <object>
               <embed src="Excel_XML_MAPS.swf" quality="high" bgcolor="#869ca7"
                 width="50%" height="50%" name="test_page" align="middle"
                 play="true"  loop="false"  quality="high"  allowScriptAccess="sameDomain"
                 type="application/x-shockwave-flash"
                 pluginspage="http://www.adobe.com/go/getflashplayer">
               </embed>
            </object>

    </htmlb:form>
  </htmlb:page>
</htmlb:content>
```

Now test the **FLIGHTS.HTML** page. The **HTML** page first calls the Xcelsius – SWF file. The SWF file gets data from the XML page created as a view in the BSP application. You can easily have a view by debugging the application on keeping the break point in the appropriate places. For example when a refresh button is placed in the Xcelsius file and whenever it is clicked, the data gets refreshed and acts based on the current data.

I think the blog on Excel XML maps connection gave a basic idea of integrating SAP with the BO. Not only R/3 data can be presented as a report or dashboard, but also the BW data can also be presented by representing the BW Query or BW objects into the customized FM's and can be operated as the same as that of operating the R/3 data through the XML file. Below shows the sample application which is consuming data from BW query.



XCelsius-SAP (Excel XMl Maps Connection) - Windows Internet Explorer
http://[redacted]/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_excelmaps/flights.html
XCelsius-SAP (Excel XMl Maps Connection)
Flight Date
12-07-2006
09-08-2006
06-09-2006
04-10-2006
Seat Structure
Seats
450
350
250
150
0
Maximum
Occupied
Economy
Business
First

Integrating SAP data (R/3 or BW Data) with the BO-Xcelsius (Business Objects) using BSP

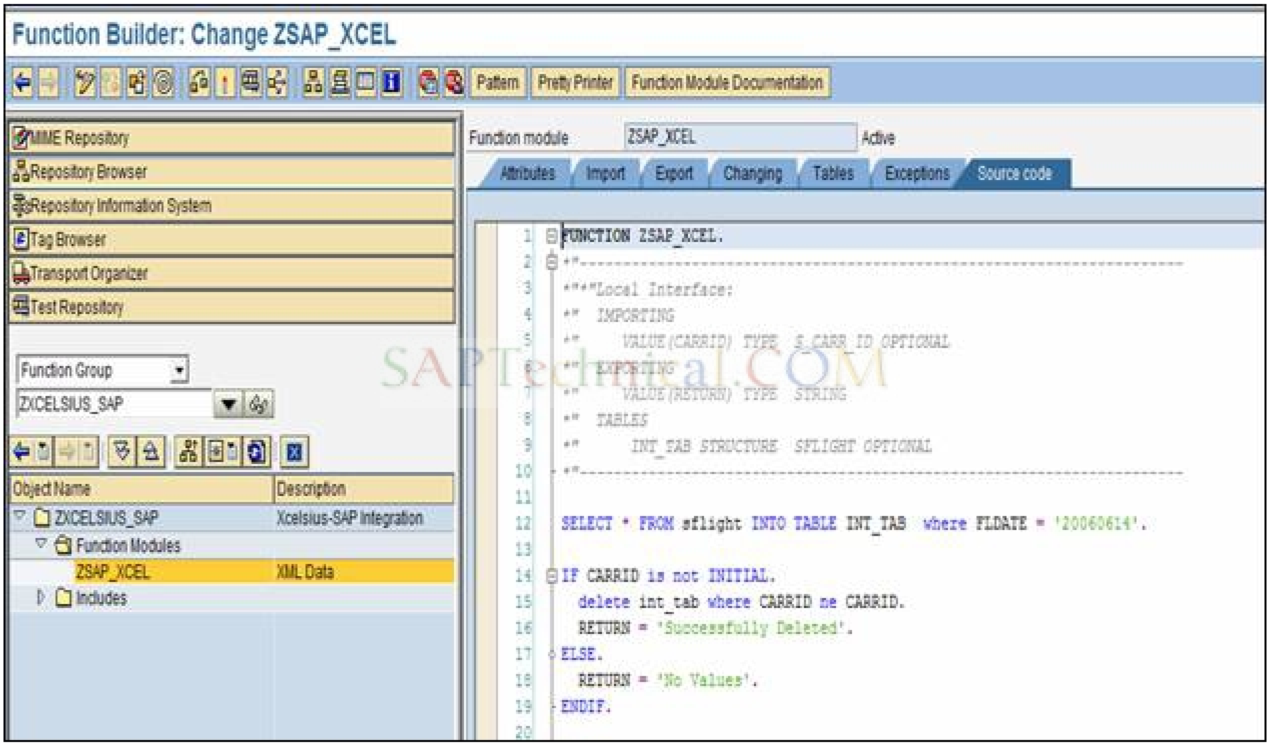
Part 2: SAP BSP + BO-Xcelsius (web Service Connection)

As we discussed earlier about the SAP and BO integration using BSP and Xcelsius, here I have used another connection of Xcelsius to integrate with the SAP BSP.

This experiment is intended to describe the integration of SAP with BO-Xcelsius using web service. The R/3 or BW data will be consumed as web service so that the service will be used with the data connectivity of Xcelsius. The data gets refreshed every moment based on the user action through the interface where we are going to use SAP BSP as the tool.

The first step is to create the FM, where the logical operation of the report has to perform. We can either use R/3 tables, Classes, methods etc or BW Query as the source.

Create FM "**ZSAP_XCEL**", with the following code as shown below.

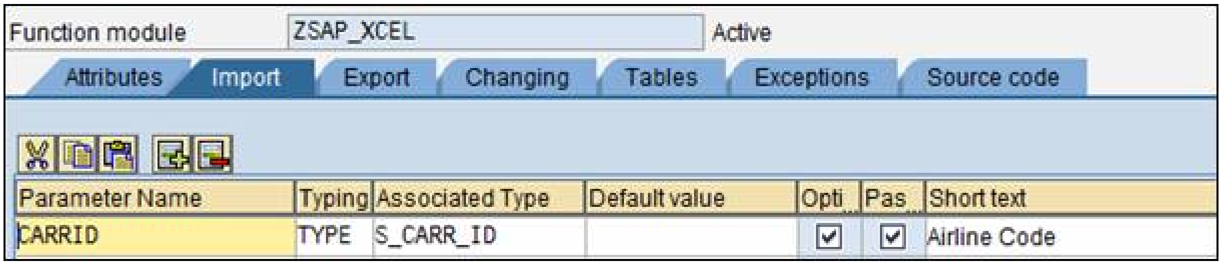


Function Group: ZXCELSIUS_SAP

Object Name	Description
ZXCELSIUS_SAP	Xcelsius-SAP Integration
Function Modules	
ZSAP_XCEL	XML Data
Includes	

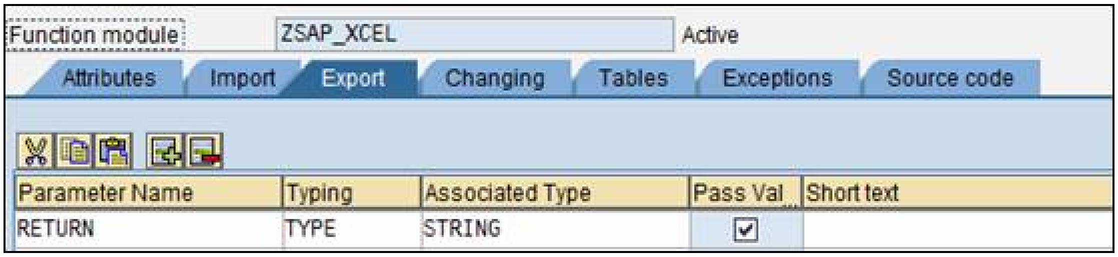
Function module: ZSAP_XCEL Active

```
FUNCTION ZSAP_XCEL.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(CARRID) TYPE  S_CARR_ID OPTIONAL
*"  EXPORTING
*"     VALUE(RETURN) TYPE  STRING
*"  TABLES
*"      INT_TAB STRUCTURE  SFLIGHT OPTIONAL
*"----------------------------------------------------------------------

SELECT * FROM sflight INTO TABLE INT_TAB  where FLDATE = '20060614'.

IF CARRID is not INITIAL.
  delete int_tab where CARRID ne CARRID.
  RETURN = 'Successfully Deleted'.
ELSE.
  RETURN = 'No Values'.
ENDIF.
```



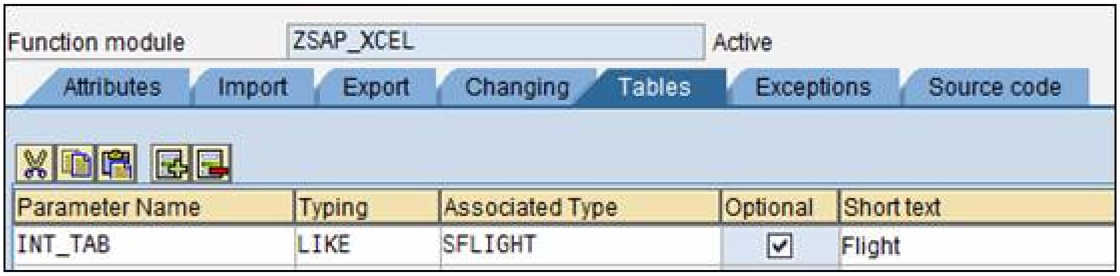
Function module: ZSAP_XCEL Active

Parameter Name	Typing	Associated Type	Default value	Opti	Pas	Short text
CARRID	TYPE	S_CARR_ID		☑	☑	Airline Code



Function module ZSAP_XCEL Active

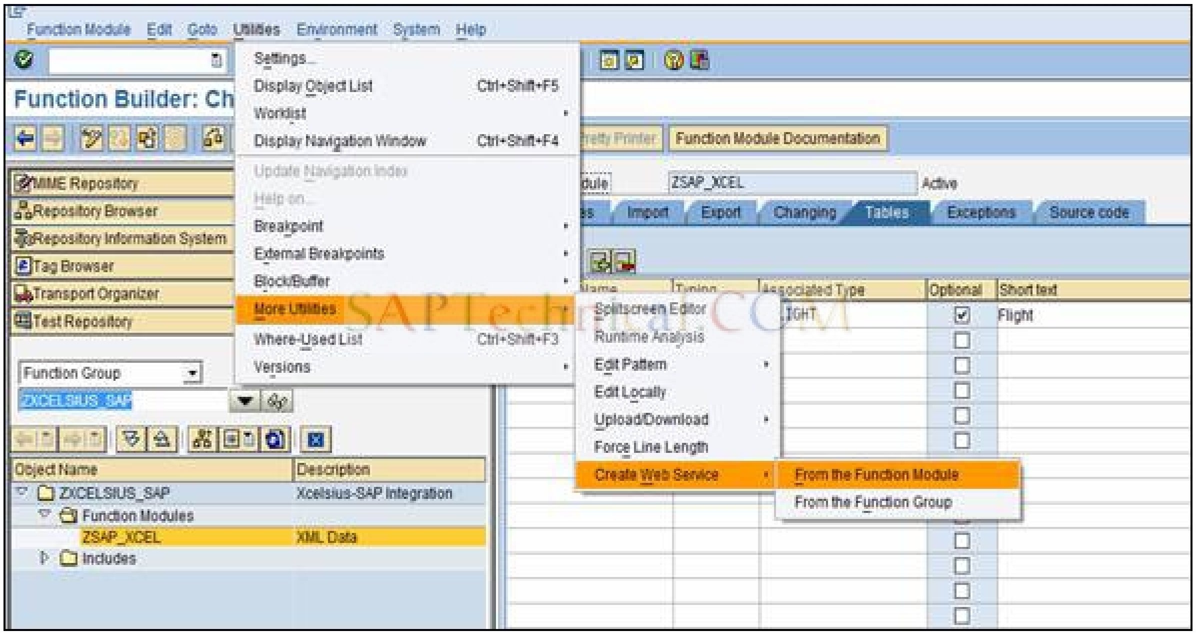
Parameter Name	Typing	Associated Type	Pass Val...	Short text
RETURN	TYPE	STRING	☑	

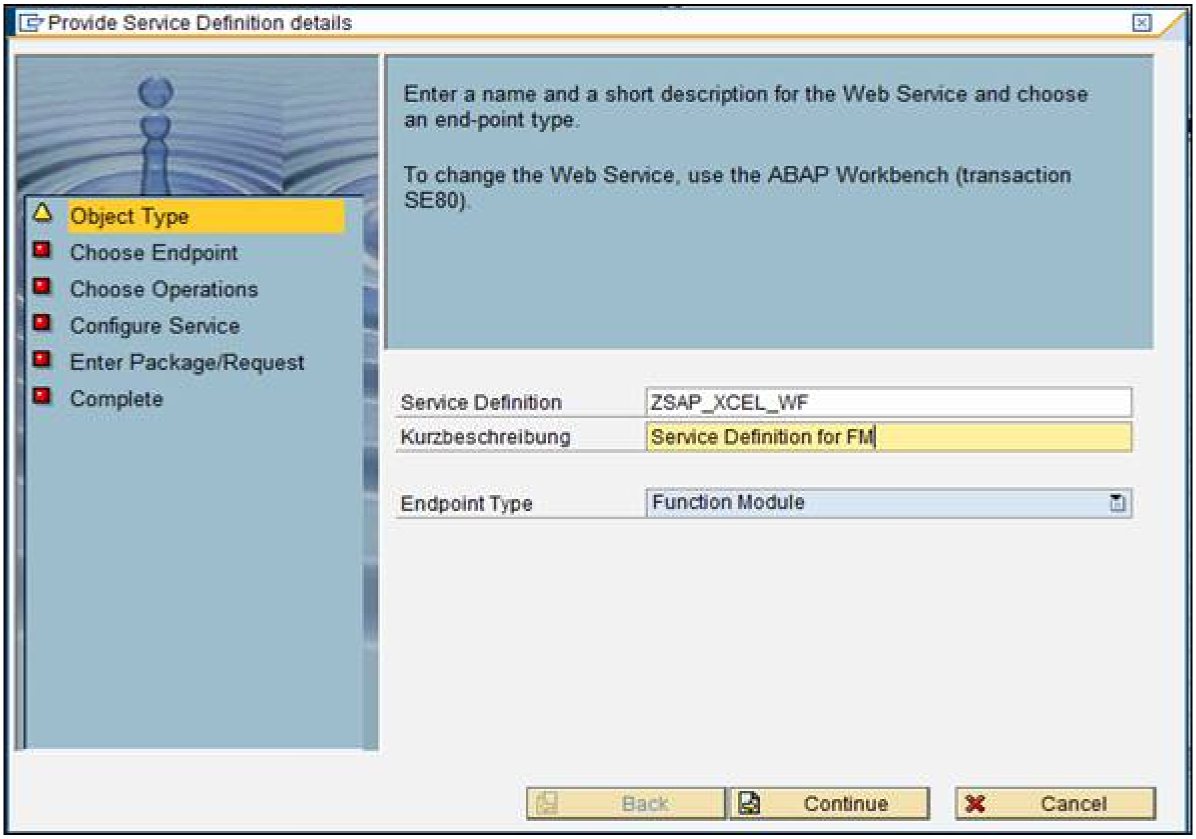


Function module ZSAP_XCEL Active

Parameter Name	Typing	Associated Type	Optional	Short text
INT_TAB	LIKE	SFLIGHT	☑	Flight

Now the FM has to be published as a web service. Follow the below steps which is mentioned in the screens.

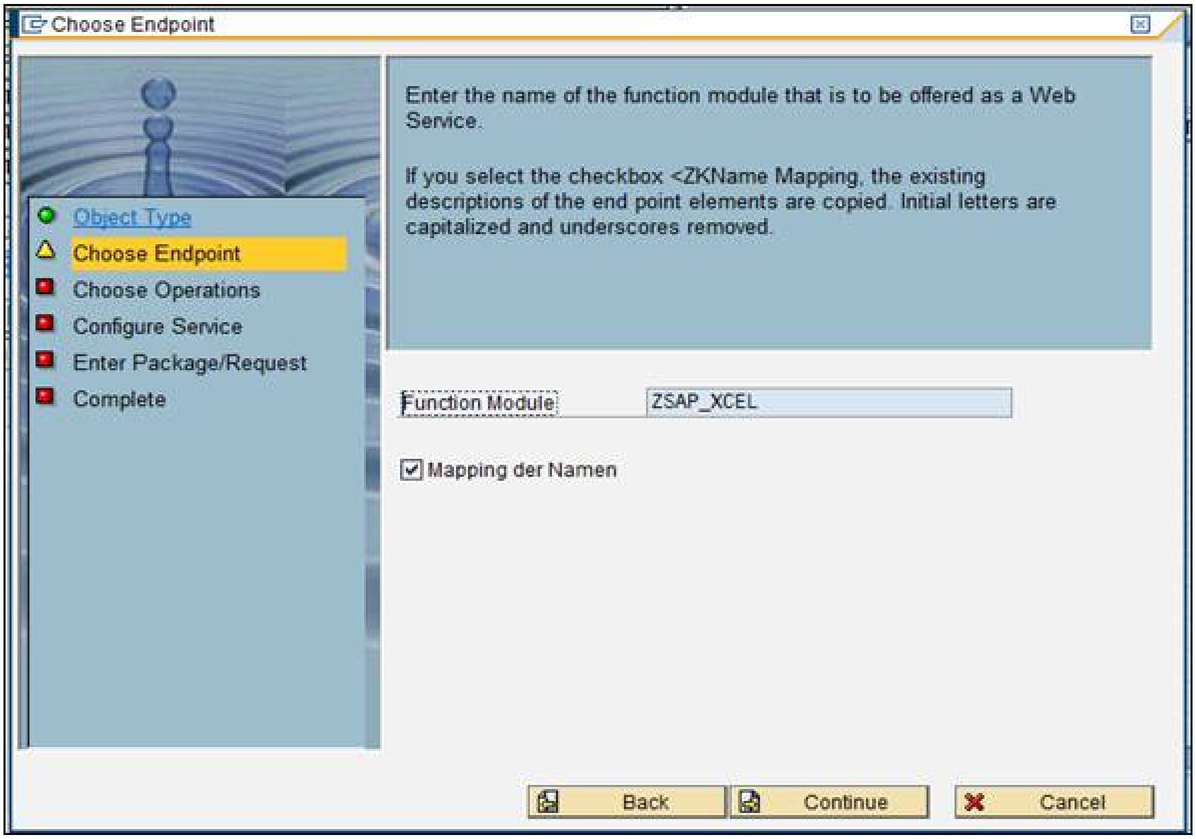




Enter a name and a short description for the Web Service and choose an end-point type.

To change the Web Service, use the ABAP Workbench (transaction SE80).

Service Definition	ZSAP_XCEL_WF
Kurzbeschreibung	Service Definition for FM
Endpoint Type	Function Module

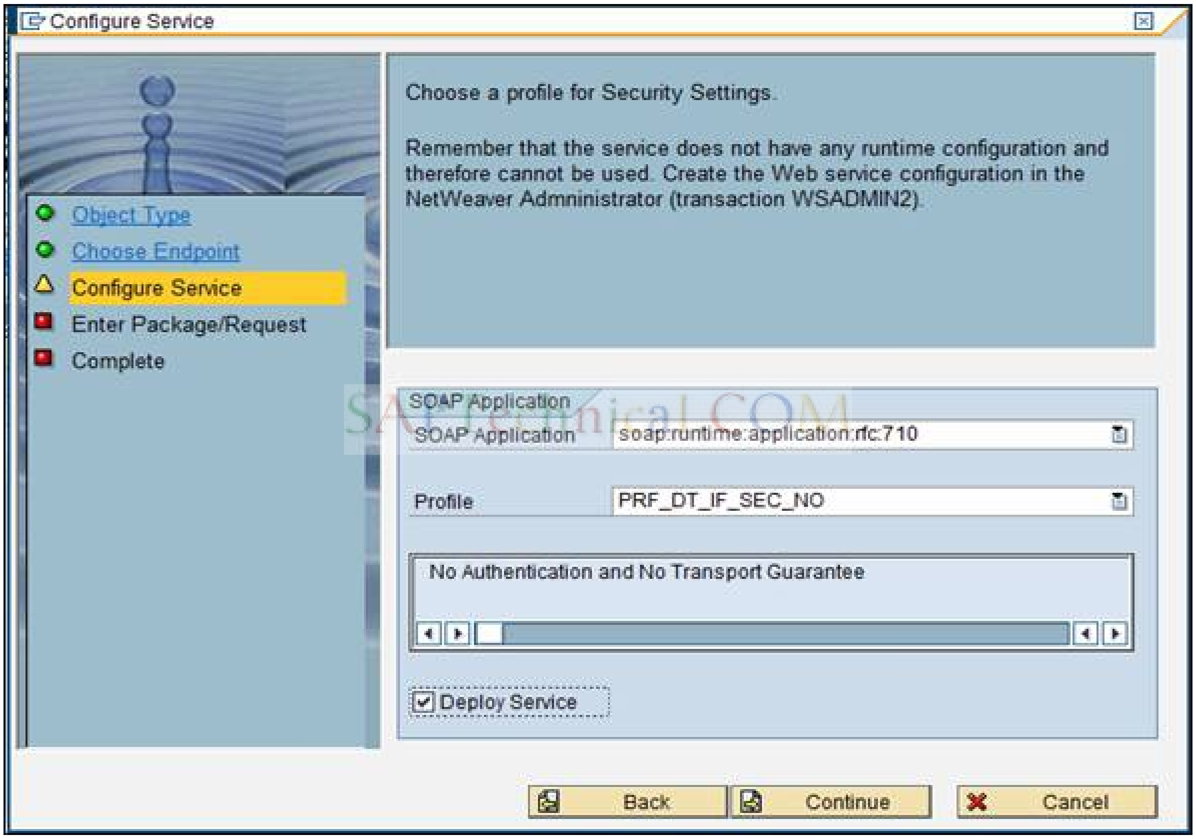


Enter the name of the function module that is to be offered as a Web Service.

If you select the checkbox <ZKName Mapping, the existing descriptions of the end point elements are copied. Initial letters are capitalized and underscores removed.

Function Module	ZSAP_XCEL

☑ Mapping der Namen



Choose a profile for Security Settings.

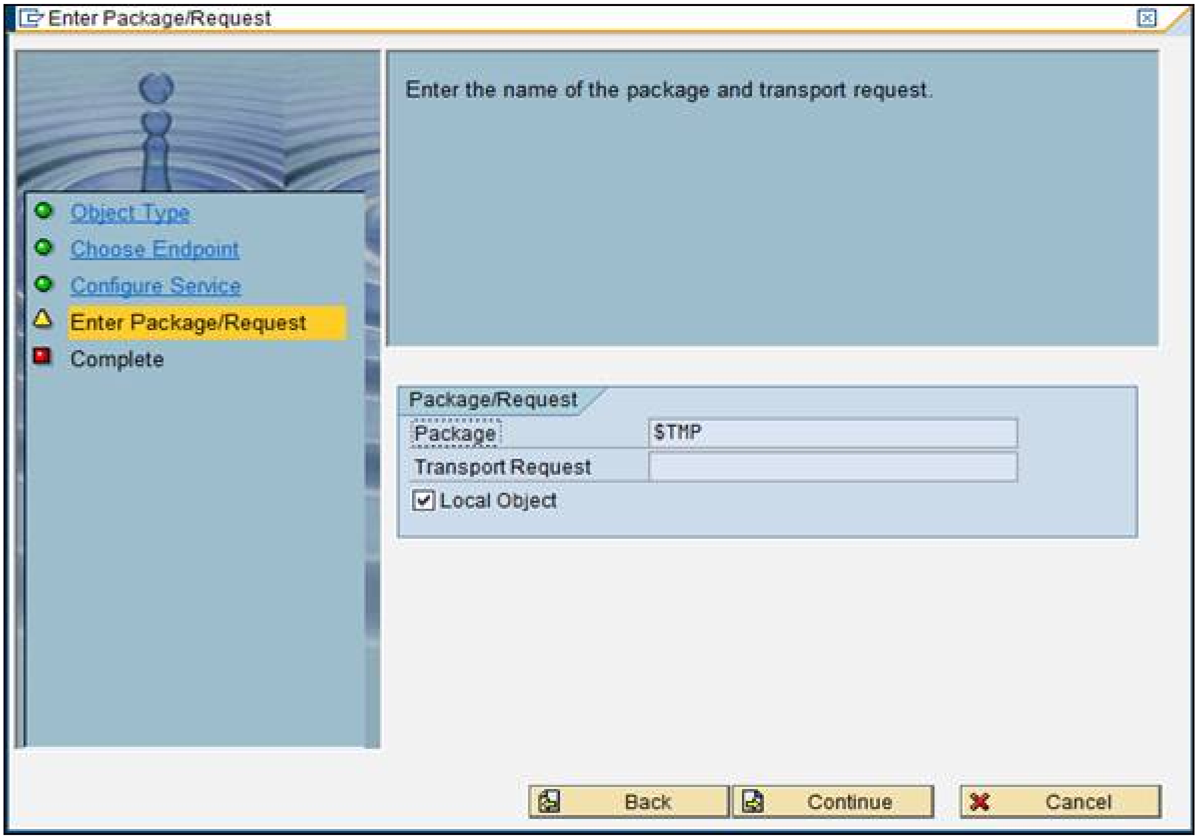
Remember that the service does not have any runtime configuration and therefore cannot be used. Create the Web service configuration in the NetWeaver Admninistrator (transaction WSADMIN2).

SOAP Application

SOAP Application	soap:runtime:application:rfc:710
Profile	PRF_DT_IF_SEC_NO

No Authentication and No Transport Guarantee

☑ Deploy Service



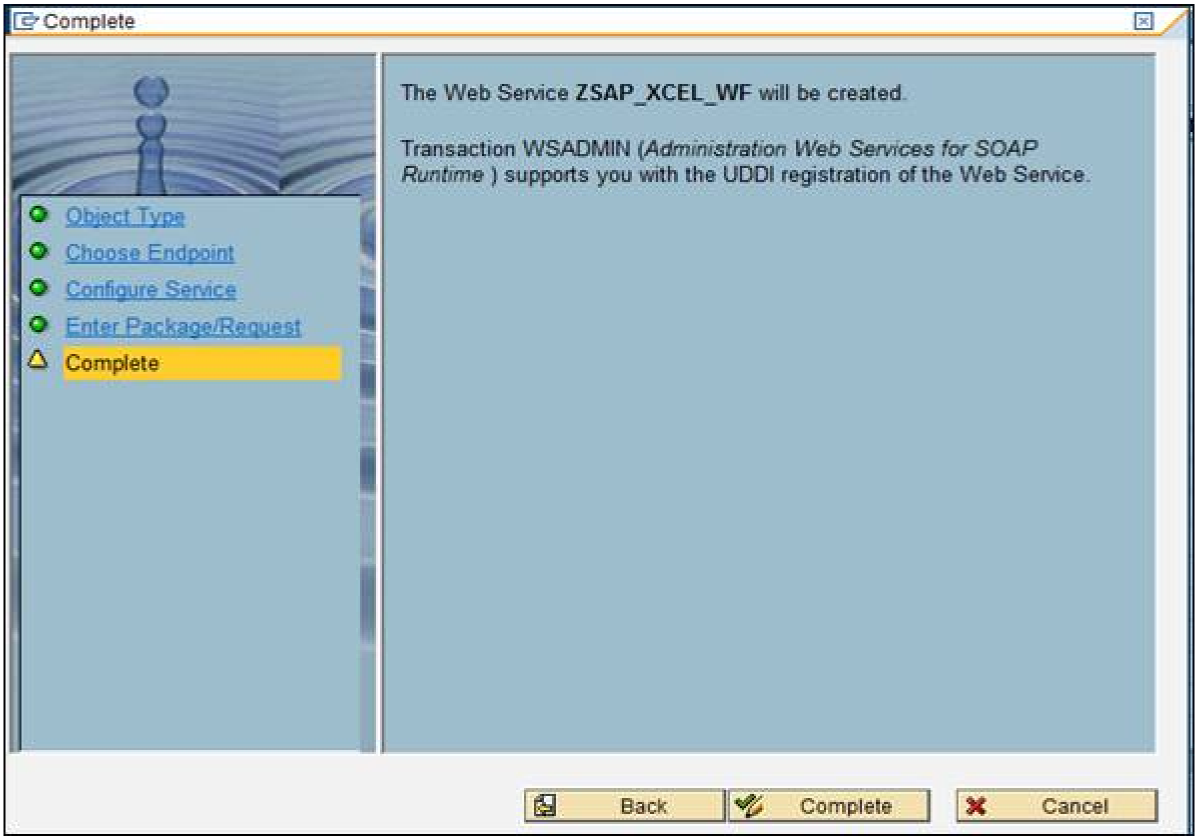
Enter Package/Request

Enter the name of the package and transport request.

Package/Request

Package	$TMP
Transport Request	

☑ Local Object

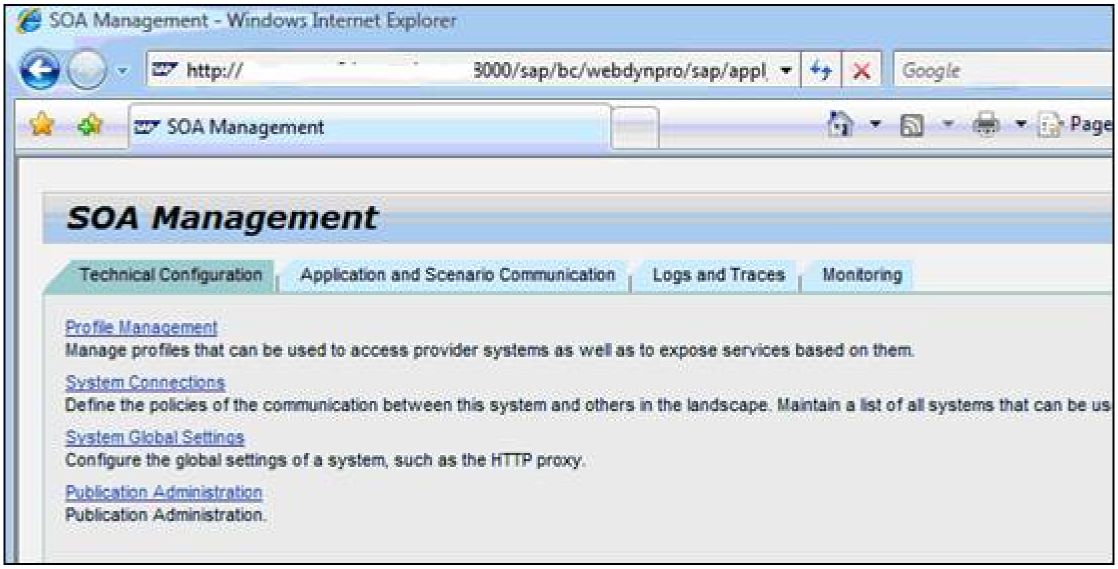


Complete

The Web Service **ZSAP_XCEL_WF** will be created.

Transaction WSADMIN (*Administration Web Services for SOAP Runtime*) supports you with the UDDI registration of the Web Service.

Once the above steps are done, open the SOA MANAGEMENT through SOAMANAGER transaction and follow the below screens stepwise.

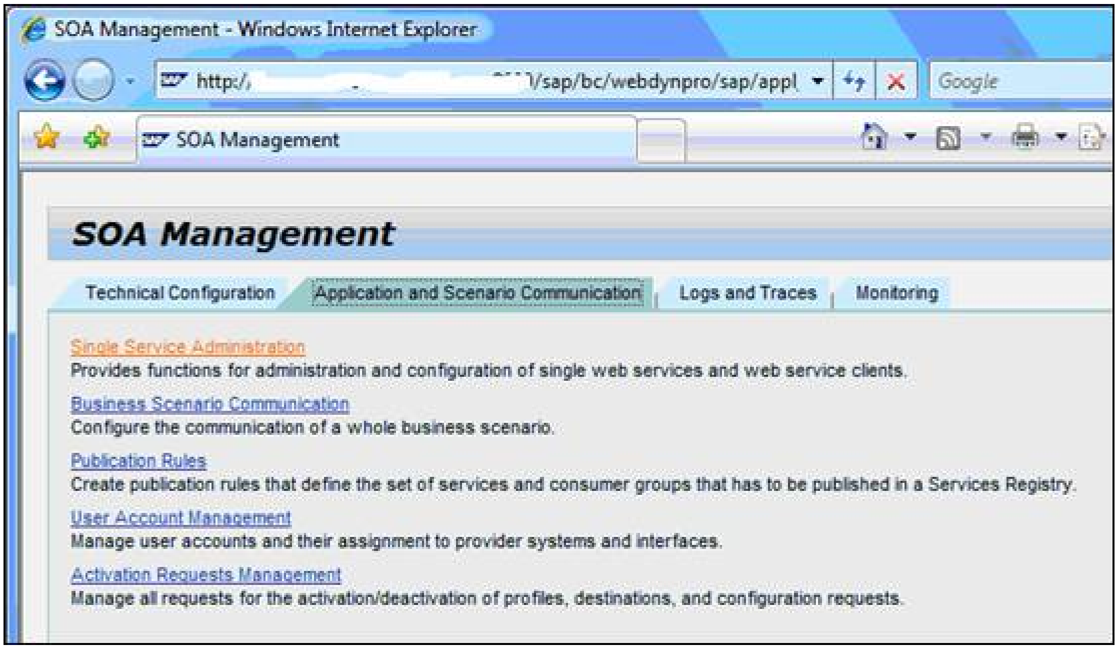

SOA Management - Windows Internet Explorer
http:// [redacted] 3000/sap/bc/webdynpro/sap/appl
Google
SOA Management
Page
SOA Management
Technical Configuration
Application and Scenario Communication
Logs and Traces
Monitoring

Profile Management
Manage profiles that can be used to access provider systems as well as to expose services based on them.

System Connections
Define the policies of the communication between this system and others in the landscape. Maintain a list of all systems that can be us

System Global Settings
Configure the global settings of a system, such as the HTTP proxy.

Publication Administration
Publication Administration.


SOA Management - Windows Internet Explorer
http:/ [redacted] /sap/bc/webdynpro/sap/appl
Google
SOA Management
SOA Management
Technical Configuration
Application and Scenario Communication
Logs and Traces
Monitoring

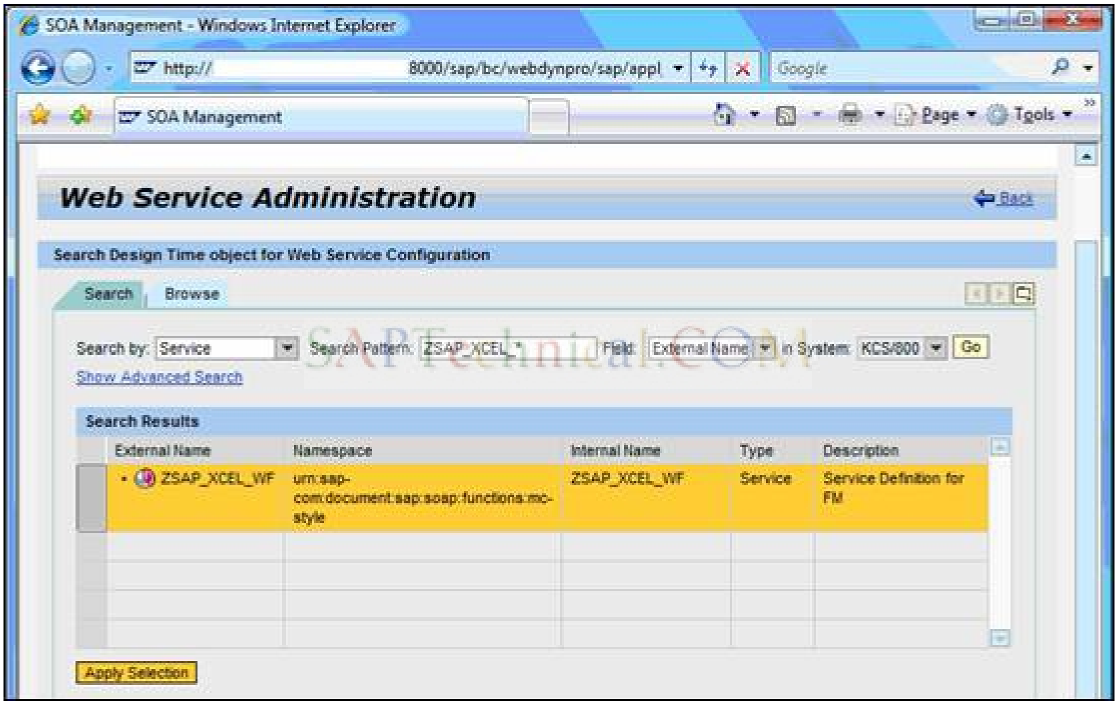
Single Service Administration
Provides functions for administration and configuration of single web services and web service clients.

Business Scenario Communication
Configure the communication of a whole business scenario.

Publication Rules
Create publication rules that define the set of services and consumer groups that has to be published in a Services Registry.

User Account Management
Manage user accounts and their assignment to provider systems and interfaces.

Activation Requests Management
Manage all requests for the activation/deactivation of profiles, destinations, and configuration requests.


SOA Management - Windows Internet Explorer

http://   8000/sap/bc/webdynpro/sap/appl

SOA Management

Web Service Administration

Back

Search Design Time object for Web Service Configuration

Search | Browse

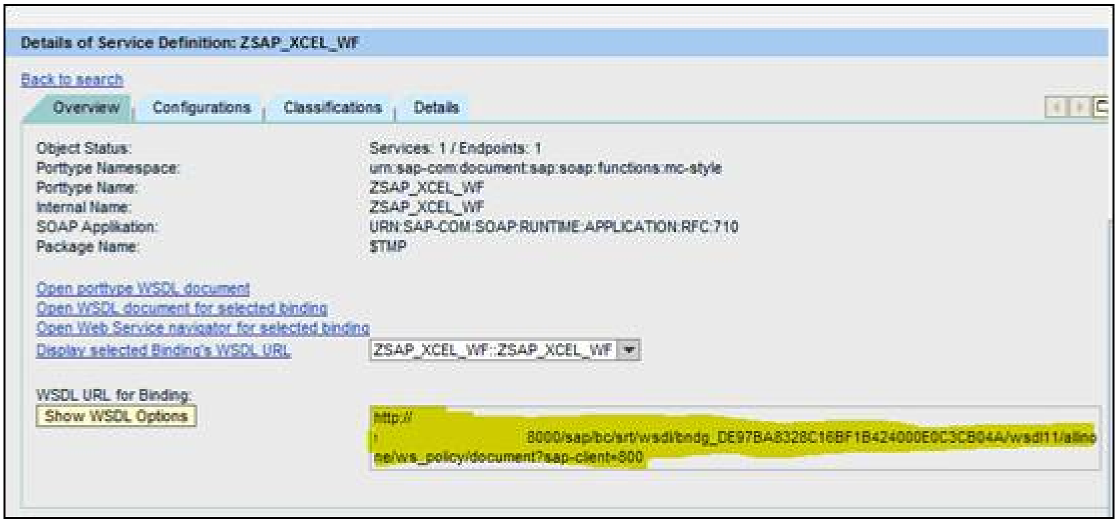
Search by: Service ▾ Search Pattern: ZSAP_XCEL_* Field: External Name ▾ in System: KCS/800 ▾ Go

Show Advanced Search

Search Results

External Name	Namespace	Internal Name	Type	Description
ZSAP_XCEL_WF	urn:sap-com:document:sap:soap:functions:mc-style	ZSAP_XCEL_WF	Service	Service Definition for FM

Apply Selection


Details of Service Definition: ZSAP_XCEL_WF

Back to search

Overview | Configurations | Classifications | Details

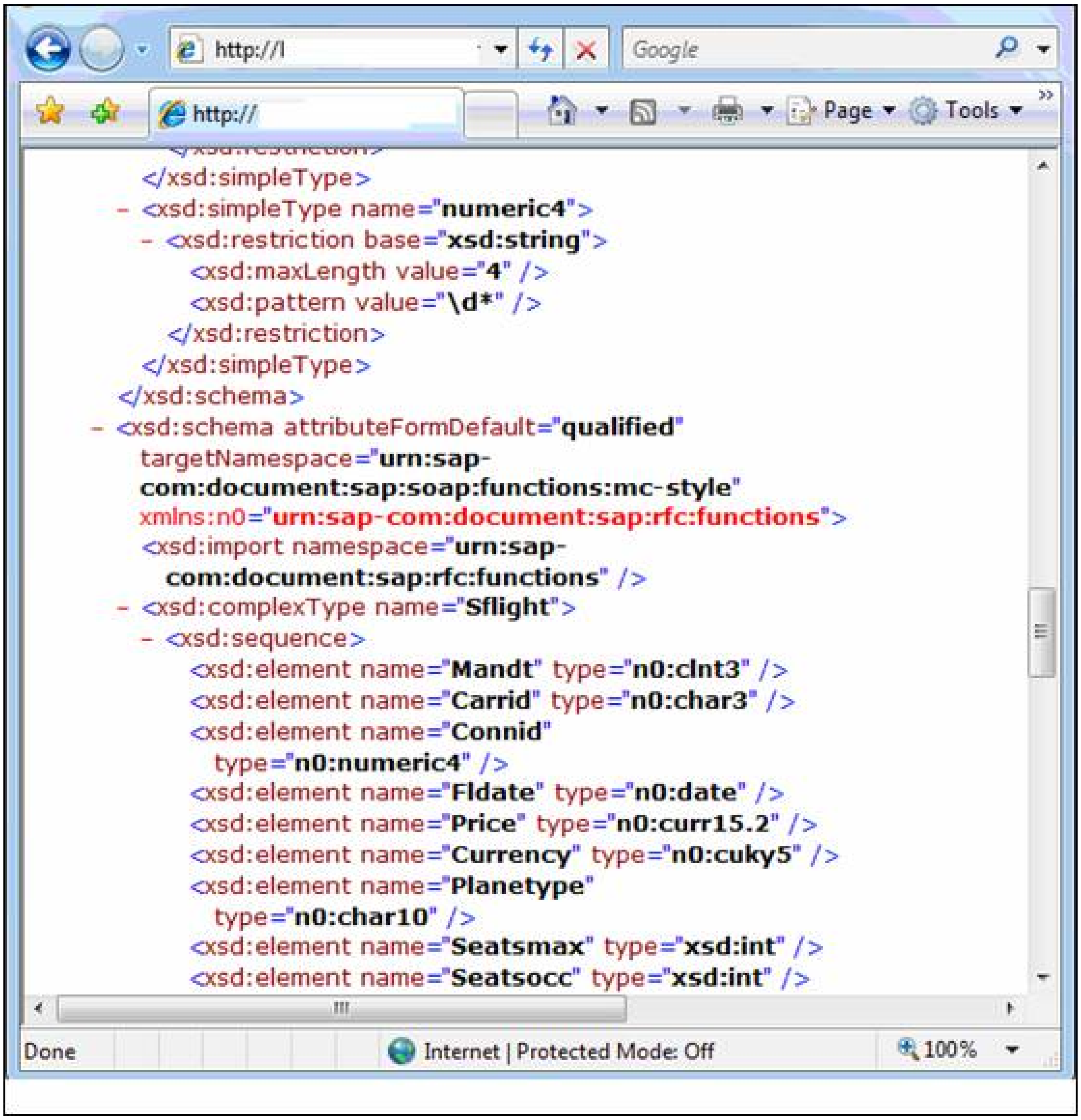
Object Status:	Services: 1 / Endpoints: 1
Porttype Namespace:	urn:sap-com:document:sap:soap:functions:mc-style
Porttype Name:	ZSAP_XCEL_WF
Internal Name:	ZSAP_XCEL_WF
SOAP Applikation:	URN:SAP-COM:SOAP:RUNTIME:APPLICATION:RFC:710
Package Name:	$TMP

Open porttype WSDL document
Open WSDL document for selected binding
Open Web Service navigator for selected binding
Display selected Binding's WSDL URL   ZSAP_XCEL_WF::ZSAP_XCEL_WF ▾

WSDL URL for Binding:
Show WSDL Options

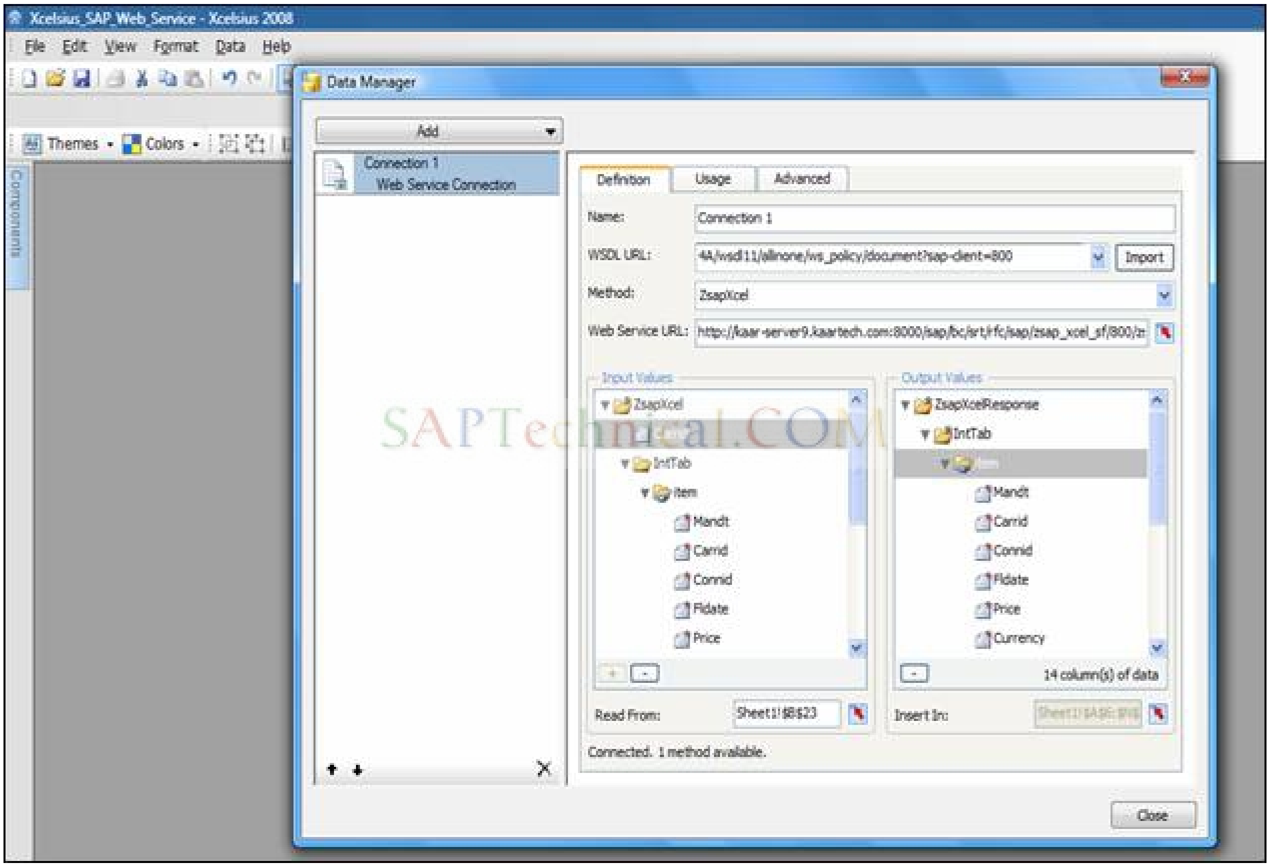
http://
:   8000/sap/bc/srt/wsdl/bndg_DE97BA8328C16BF1B424000E0C3CB04A/wsdl11/allinone/ws_policy/document?sap-client=800

We can test the data of the FM as XML file through the web service URL by copying the link which is highlighted above and place it in IE.



```
        </xsd:restriction>
      </xsd:simpleType>
    - <xsd:simpleType name="numeric4">
      - <xsd:restriction base="xsd:string">
          <xsd:maxLength value="4" />
          <xsd:pattern value="\d*" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:schema>
  - <xsd:schema attributeFormDefault="qualified"
      targetNamespace="urn:sap-
      com:document:sap:soap:functions:mc-style"
      xmlns:n0="urn:sap-com:document:sap:rfc:functions">
      <xsd:import namespace="urn:sap-
        com:document:sap:rfc:functions" />
    - <xsd:complexType name="Sflight">
      - <xsd:sequence>
          <xsd:element name="Mandt" type="n0:clnt3" />
          <xsd:element name="Carrid" type="n0:char3" />
          <xsd:element name="Connid"
            type="n0:numeric4" />
          <xsd:element name="Fldate" type="n0:date" />
          <xsd:element name="Price" type="n0:curr15.2" />
          <xsd:element name="Currency" type="n0:cuky5" />
          <xsd:element name="Planetype"
            type="n0:char10" />
          <xsd:element name="Seatsmax" type="xsd:int" />
          <xsd:element name="Seatsocc" type="xsd:int" />
```

Now the report has to be designed in BO-Xcelsius based on the data from web service which will be consumed as WSDL URL.



Name: Connection 1

WSDL URL: 4A/wsdl11/allinone/ws_policy/document?sap-client=800

Method: ZsapXcel

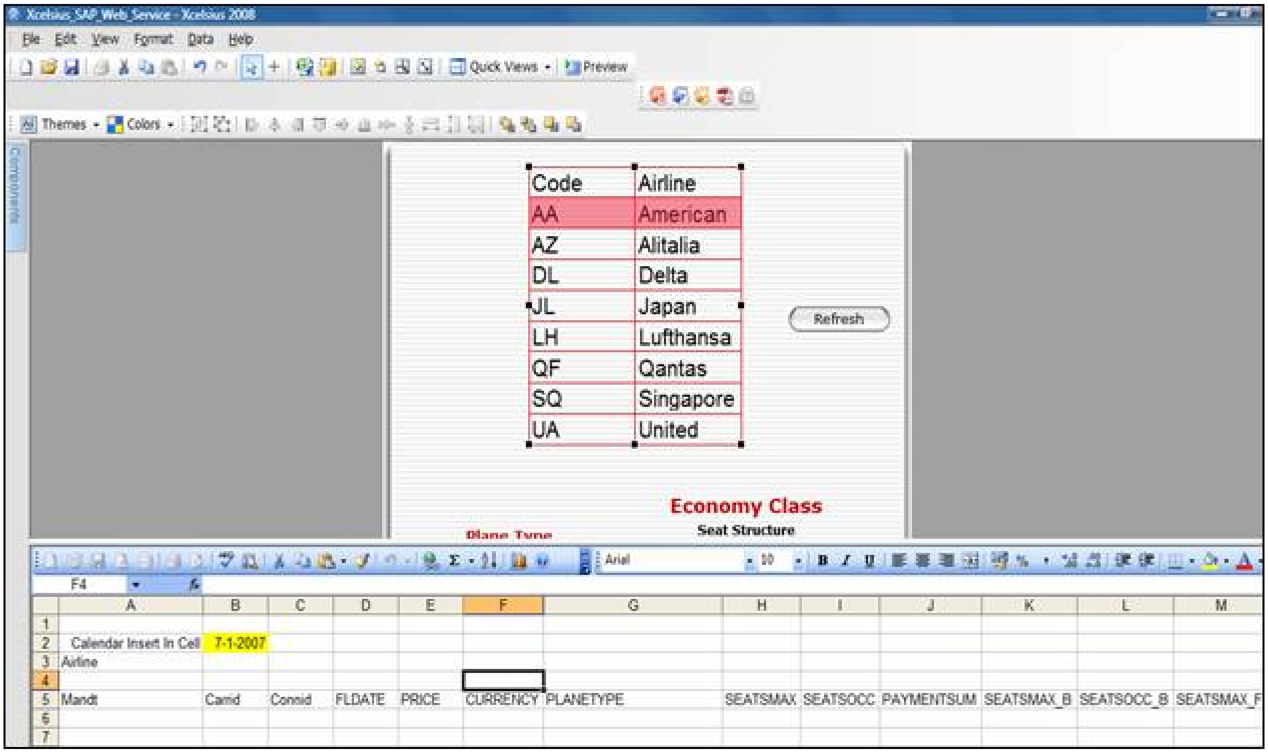
Web Service URL: http://kaar-server9.kaartech.com:8000/sap/bc/srt/rfc/sap/zsap_xcel_sf/800/z

Read From: Sheet1!$B$23

Insert In: [illegible]

Connected. 1 method available.

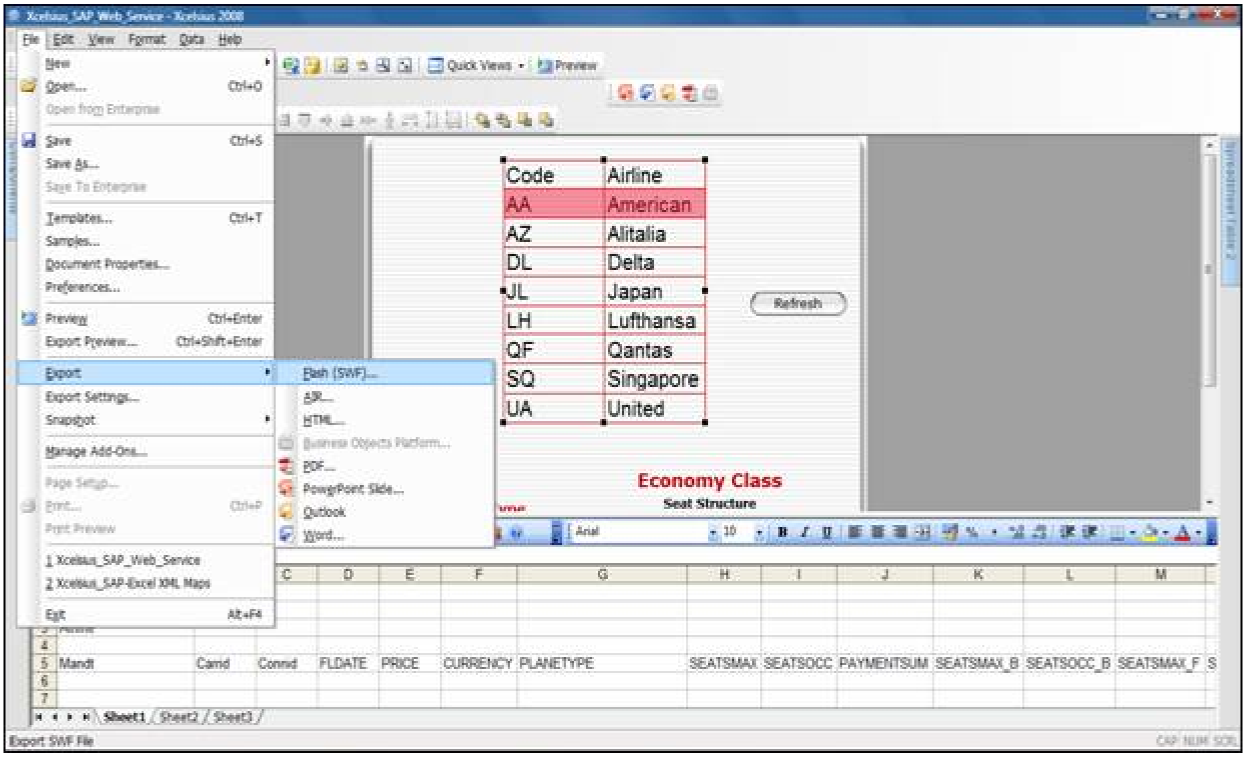
Now the report has to be designed based on the functionality / users requirement of the report. The Input (Read from) and Output (Insert In) cells have to be mapped to the Xcelsius-Excel sheet during the design phase.



Code	Airline
AA	American
AZ	Alitalia
DL	Delta
JL	Japan
LH	Lufthansa
QF	Qantas
SQ	Singapore
UA	United

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	Calendar Insert In Cell	7-1-2007											
3	Airline												
4													
5	Mandt	Carrid	Connid	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F
6													
7													

The designed Xcelsius file will be saved in XLF format locally which will be then exported as SWF file.

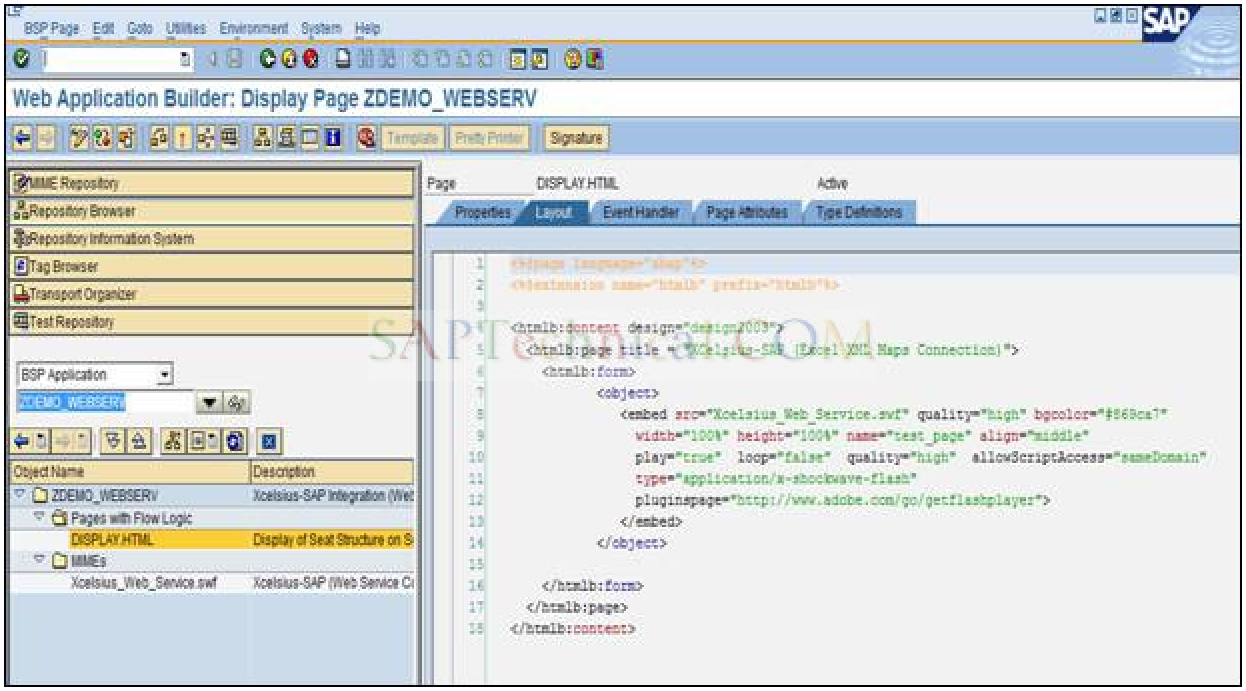


Code	Airline
AA	American
AZ	Alitalia
DL	Delta
JL	Japan
LH	Lufthansa
QF	Qantas
SQ	Singapore
UA	United

	A	B	C	D	E	F	G	H	I	J	K	L	M
5	Mandt		Carrid	Connid	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F

Now the created SWF file has to be presented to the dashboard by having SAP BSP as the web interface.

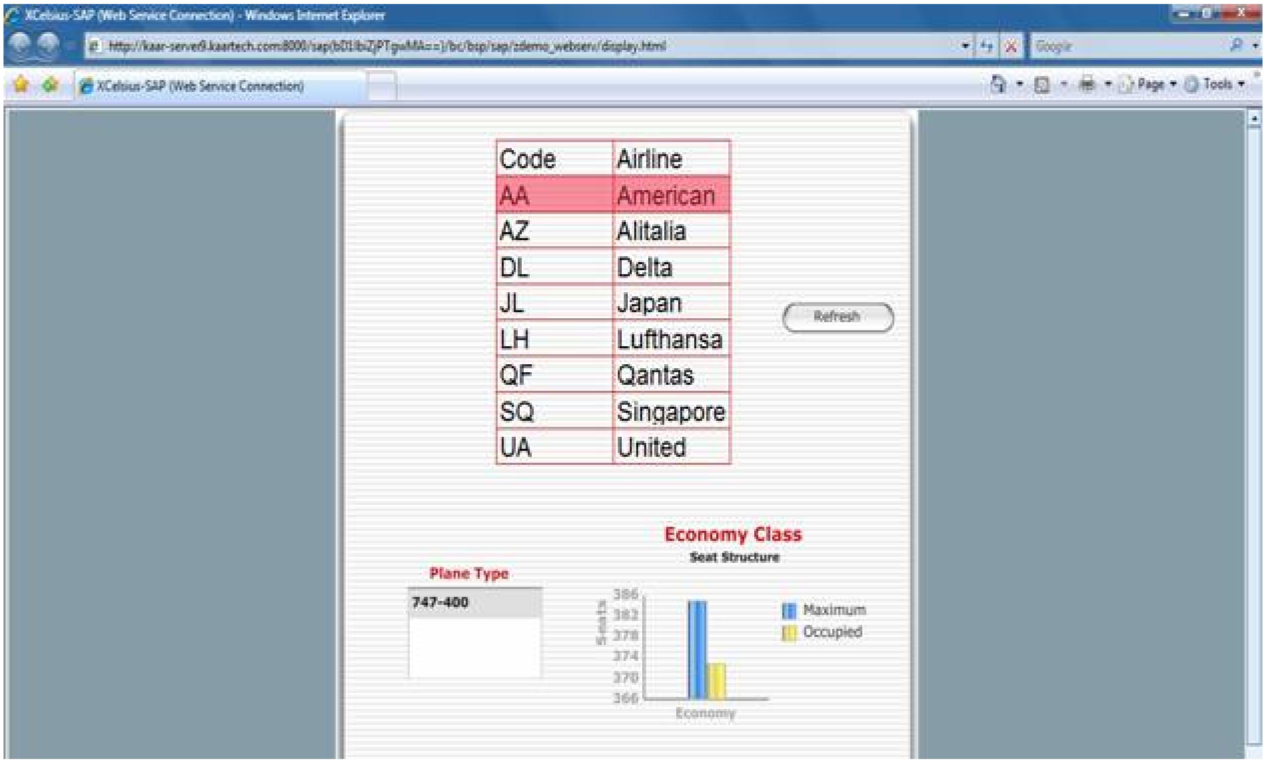
Create a BSP Application ZDEMO_WEBSERV with a HTML View-DISPLAY.HTML with the following code. Import the SWF file into the BSP application in the MIME folder.


BSP Page Edit Goto Utilities Environment System Help
Web Application Builder: Display Page ZDEMO_WEBSERV
Template Pretty Printer Signature
MIME Repository
Repository Browser
Repository Information System
Tag Browser
Transport Organizer
Test Repository
BSP Application
ZDEMO_WEBSERV
Page DISPLAY.HTML Active
Properties Layout Event Handler Page Attributes Type Definitions

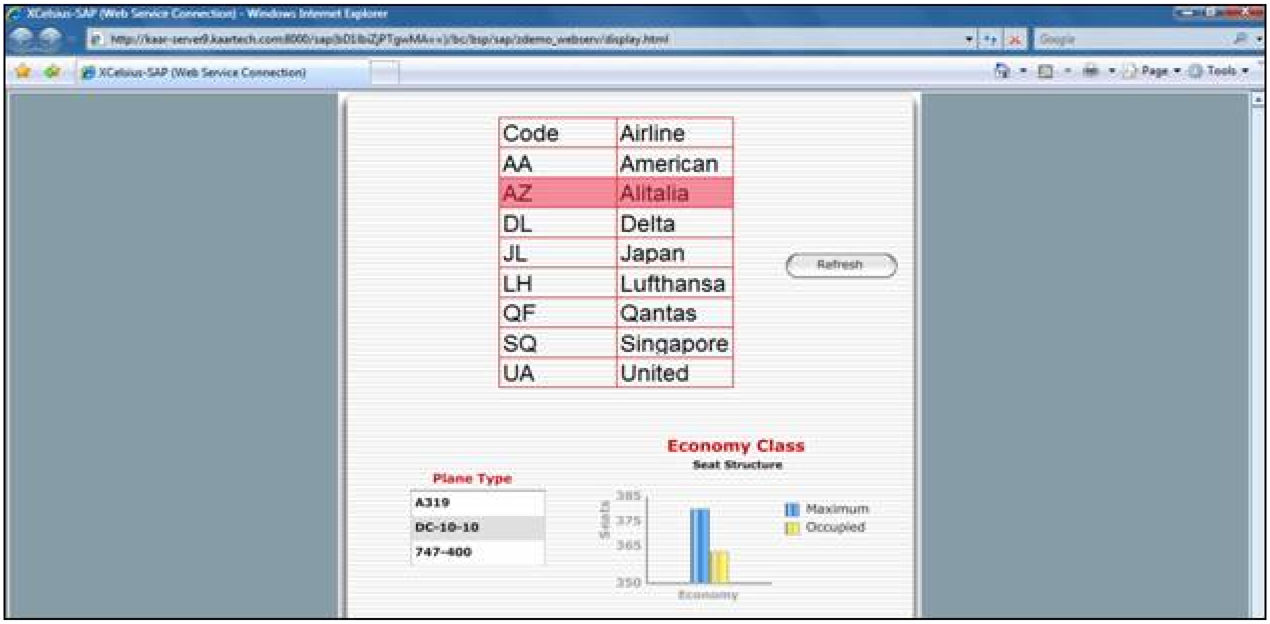
Object Name	Description
ZDEMO_WEBSERV	Xcelsius-SAP Integration (Wet
Pages with Flow Logic	
DISPLAY.HTML	Display of Seat Structure on S
MIMEs	
Xcelsius_Web_Service.swf	Xcelsius-SAP (Web Service Co

```
<%@page language="abap"%>
<%@extension name="htmlb" prefix="htmlb"%>

<htmlb:content design="design2003">
  <htmlb:page title = "Xcelsius-SAP (Excel XML Maps Connection)">
    <htmlb:form>
           <object>
              <embed src="Xcelsius_Web_Service.swf" quality="high" bgcolor="#869ca7"
                width="100%" height="100%" name="test_page" align="middle"
                play="true"  loop="false"  quality="high"  allowScriptAccess="sameDomain"
                type="application/x-shockwave-flash"
                pluginspage="http://www.adobe.com/go/getflashplayer">
              </embed>
           </object>

    </htmlb:form>
  </htmlb:page>
</htmlb:content>
```

Now test the DISPLAY.HTML and interact with Xcelsius file by giving input and output will be based on the functionality of the web service.


XCelsius-SAP (Web Service Connection) - Windows Internet Explorer
http://kaar-server9.kaartech.com:8000/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_webserv/display.html
Google
XCelsius-SAP (Web Service Connection)
Page Tools
Refresh
Economy Class
Seat Structure
Plane Type
747-400
Seats
386
382
378
374
370
366
Economy
Maximum
Occupied

Code	Airline
AA	American
AZ	Alitalia
DL	Delta
JL	Japan
LH	Lufthansa
QF	Qantas
SQ	Singapore
UA	United


XCelsius-SAP (Web Service Connection) - Windows Internet Explorer
http://kaar-server9.kaartech.com:8000/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_webserv/display.html
XCelsius-SAP (Web Service Connection)
Page
Tools
Refresh

Code	Airline
AA	American
AZ	Alitalia
DL	Delta
JL	Japan
LH	Lufthansa
QF	Qantas
SQ	Singapore
UA	United

Plane Type
A319
DC-10-10
747-400
Economy Class
Seat Structure
Seats
385
375
365
350
Economy
Maximum
Occupied

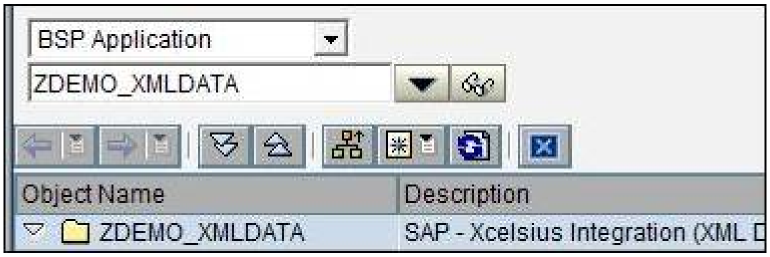
Here the R/3 data is consumed as a FM and used as the source. In the previous part also R/3 data is used as the source. BW data can also be used as the source data in the previous and for this experiment also.

Integrating SAP data (R/3 or BW Data) with the BO-Xcelsius (Business Objects) using BSP

Part 3: SAP BSP + BO-Xcelsius (XML Data Connection)

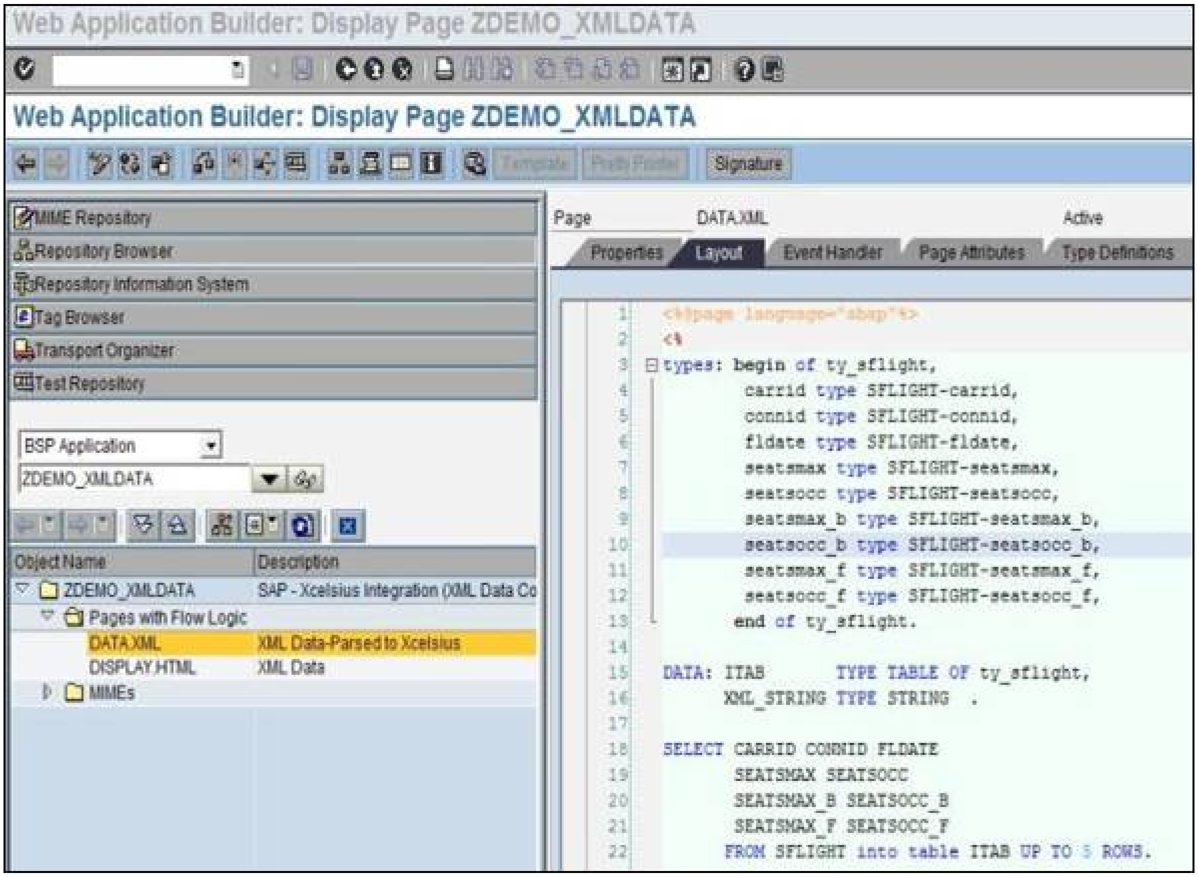
In this presentation the third connection of Xcelsius – XML data is used to integrate with SAP BSP, so that the data will be presented in the dashboard.

Create a sample BSP application “ZDEMO_XMLDATA”

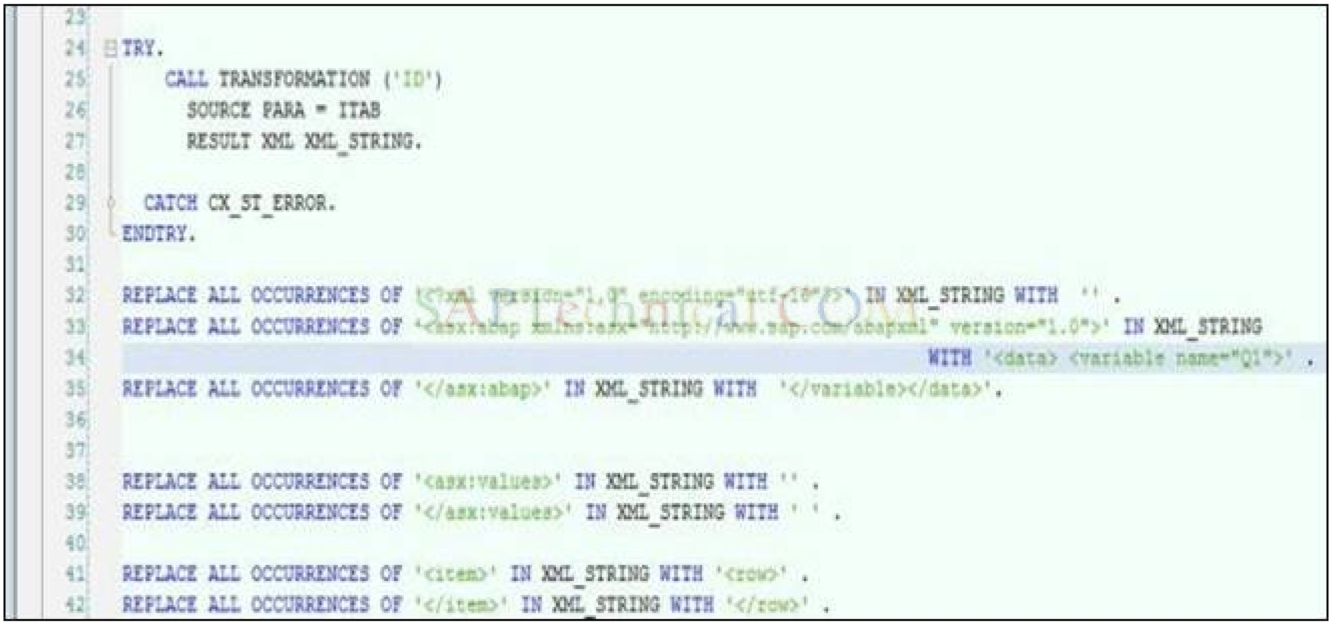


Object Name	Description
ZDEMO_XMLDATA	SAP - Xcelsius Integration (XML D

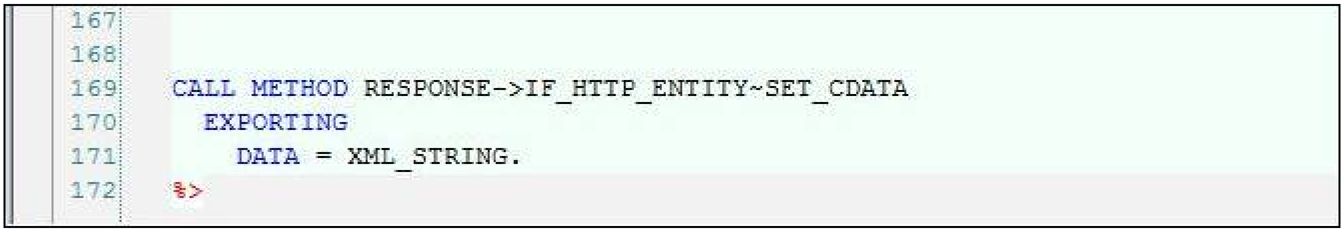
Create a XML view in the BSP application based on the PWFL model. Embed the below code shown in the following screens. The code is mainly projected to generate the XML file which is going to act as a source for the SWF file in a particular format. The CALL **TRANSFORMATION ('ID')** will generate the XML file using the data in a format based on the system / server settings. The pre defined format has to be achieved based on replacing some tags with other values as shown in the below screens.



```
<%@page language="abap"%>
<%
types: begin of ty_sflight,
        carrid type SFLIGHT-carrid,
        connid type SFLIGHT-connid,
        fldate type SFLIGHT-fldate,
        seatsmax type SFLIGHT-seatsmax,
        seatsocc type SFLIGHT-seatsocc,
        seatsmax_b type SFLIGHT-seatsmax_b,
        seatsocc_b type SFLIGHT-seatsocc_b,
        seatsmax_f type SFLIGHT-seatsmax_f,
        seatsocc_f type SFLIGHT-seatsocc_f,
       end of ty_sflight.

DATA: ITAB       TYPE TABLE OF ty_sflight,
      XML_STRING TYPE STRING  .

SELECT CARRID CONNID FLDATE
       SEATSMAX SEATSOCC
       SEATSMAX_B SEATSOCC_B
       SEATSMAX_F SEATSOCC_F
      FROM SFLIGHT into table ITAB UP TO 5 ROWS.
```

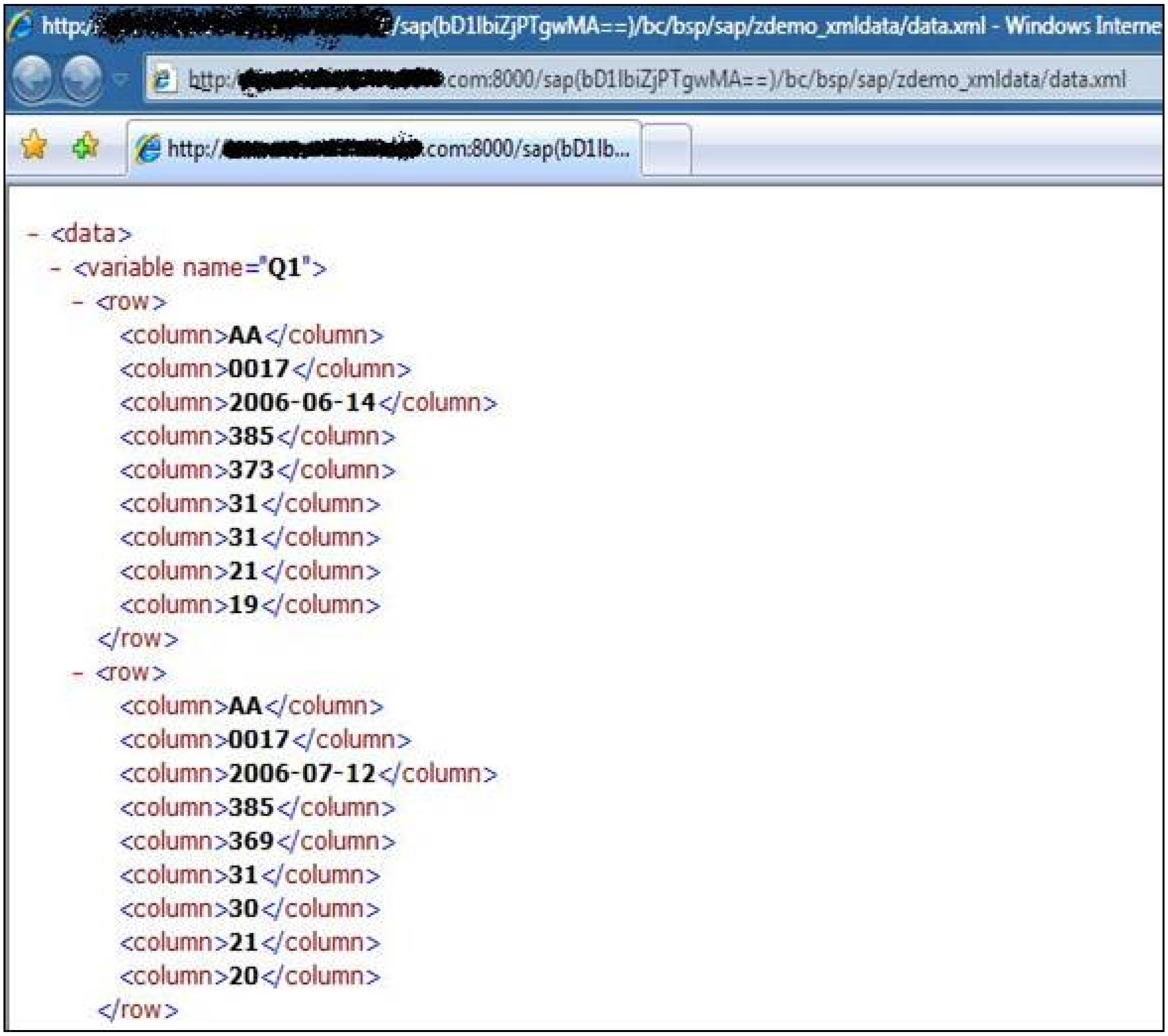


```

TRY.
    CALL TRANSFORMATION ('ID')
      SOURCE PARA = ITAB
      RESULT XML XML_STRING.

  CATCH CX_ST_ERROR.
ENDTRY.

REPLACE ALL OCCURRENCES OF '<?xml version="1.0" encoding="utf-16"?>' IN XML_STRING WITH  '' .
REPLACE ALL OCCURRENCES OF '<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">' IN XML_STRING
                                                                        WITH '<data> <variable name="Q1">' .
REPLACE ALL OCCURRENCES OF '</asx:abap>' IN XML_STRING WITH  '</variable></data>'.


REPLACE ALL OCCURRENCES OF '<asx:values>' IN XML_STRING WITH '' .
REPLACE ALL OCCURRENCES OF '</asx:values>' IN XML_STRING WITH ' ' .

REPLACE ALL OCCURRENCES OF '<item>' IN XML_STRING WITH '<row>' .
REPLACE ALL OCCURRENCES OF '</item>' IN XML_STRING WITH '</row>' .
```



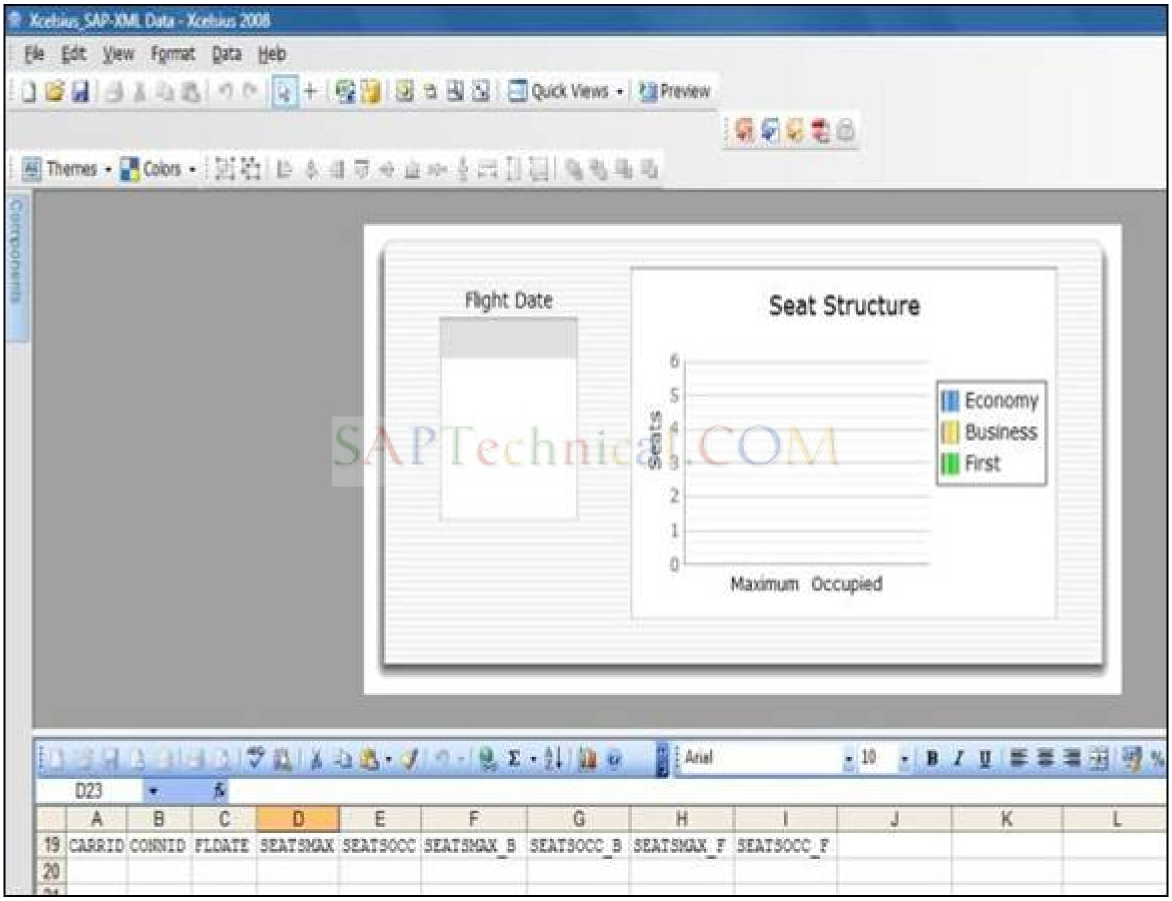
```

CALL METHOD RESPONSE->IF_HTTP_ENTITY~SET_CDATA
  EXPORTING
    DATA = XML_STRING.
%>
```

The below screen is the predefined format with data, variable, row and column tags generated while testing the XML view. The generated XML has to be parsed to the Xcelsius file. Sometimes the target might not be achieved if the format of the XML file is not proper.


http://[redacted]/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_xmldata/data.xml - Windows Interne
http://[redacted].com:8000/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_xmldata/data.xml
http://[redacted].com:8000/sap(bD1lb...

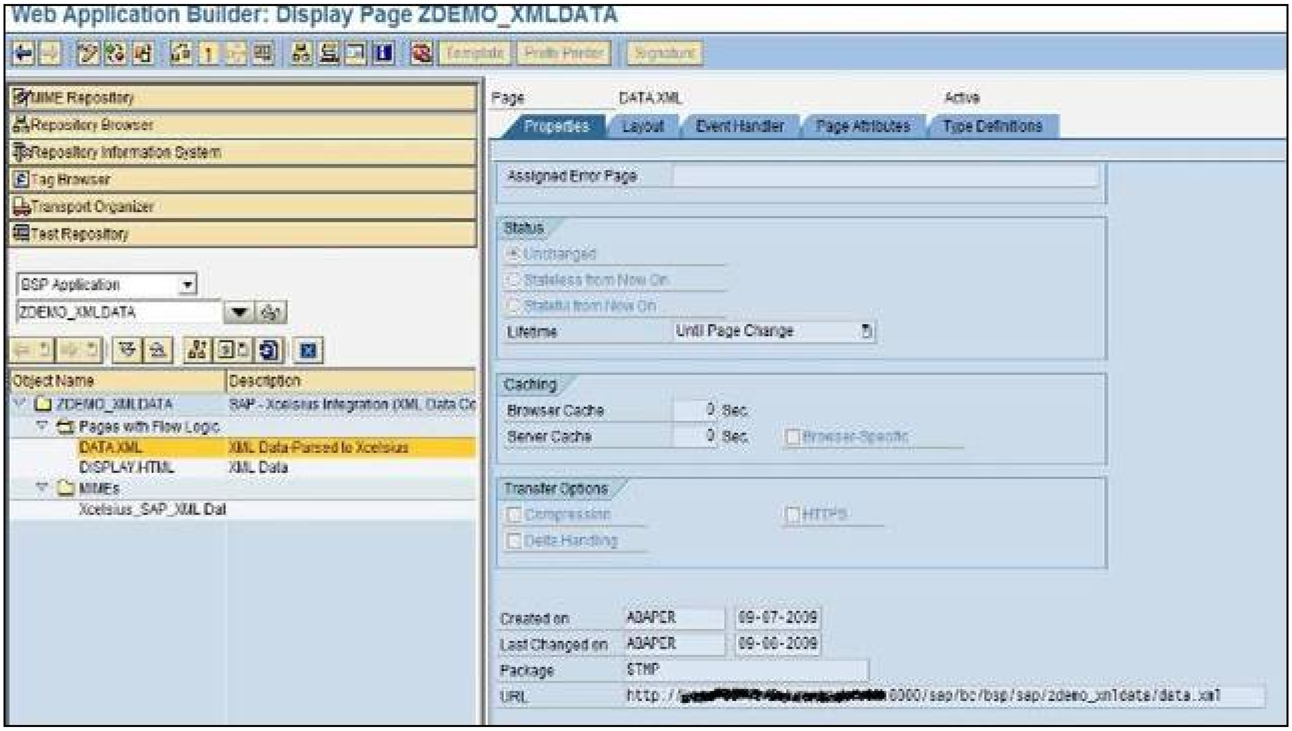
```
- <data>
  - <variable name="Q1">
    - <row>
        <column>AA</column>
        <column>0017</column>
        <column>2006-06-14</column>
        <column>385</column>
        <column>373</column>
        <column>31</column>
        <column>31</column>
        <column>21</column>
        <column>19</column>
      </row>
    - <row>
        <column>AA</column>
        <column>0017</column>
        <column>2006-07-12</column>
        <column>385</column>
        <column>369</column>
        <column>31</column>
        <column>30</column>
        <column>21</column>
        <column>20</column>
      </row>
```

Open the Xcelsius 2008 and design the canvas using a list box and a column chart. The list box will be holding the date of the flight and the column chart will represent the maximum and occupied seats of the particular class on the selected date.



	A	B	C	D	E	F	G	H	I	J	K	L
19	CARRID	CONNID	FLDATE	SEATSMAX	SEATSOCC	SEATSMAX_B	SEATSOCC_B	SEATSMAX_F	SEATSOCC_F			
20												

Now the data connectivity has to be assigned to the Xcelsius file. The XML path will be taken from the properties of the XML view as shown below.



Web Application Builder: Display Page ZDEMO_XMLDATA

Object Name	Description
ZDEMO_XMLDATA	SAP - Xcelsius Integration (XML Data Co
Pages with Flow Logic	
DATA.XML	XML Data-Parsed to Xcelsius
DISPLAY.HTML	XML Data
MIMEs	
Xcelsius_SAP_XML Dat	

Page DATA.XML Active

Assigned Error Page

Status

○ Unchanged

○ Stateless from Now On

○ Stateful from Now On

Lifetime: Until Page Change

Caching

Browser Cache: 0 Sec.

Server Cache: 0 Sec. □ Browser-Specific

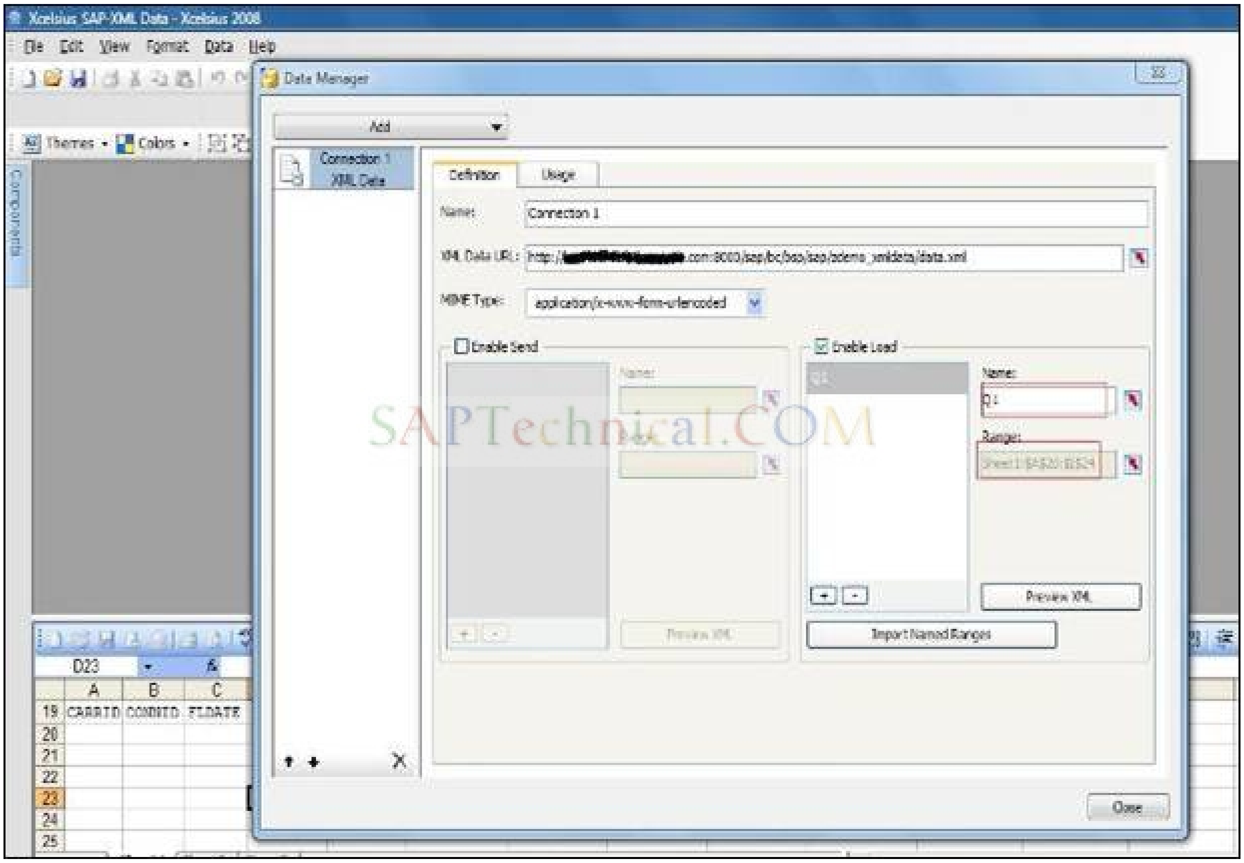
Transfer Options

□ Compression □ HTTPS

□ Delta Handling

Created on	ADAPER	09-07-2009
Last Changed on	ADAPER	09-08-2009
Package	$TMP	
URL	http://[redacted]:8000/sap/bc/bsp/sap/zdemo_xmldata/data.xml	

The XML data URL will be the copied URL from the above screen. The URL can also be dynamically changed by keeping tags on the server name. eq. **http://<webserver>.com:<port>/sap(namesapce)**



Xcelsius SAP-XML Data - Xcelsius 2008

Data Manager

Connection 1 XML Data

Name: Connection 1

XML Data URL: http://[redacted].com:8000/sap/bc/bsp/sap/zdemo_xmldata/data.xml

MIME Type: application/x-www-form-urlencoded

□ Enable Send

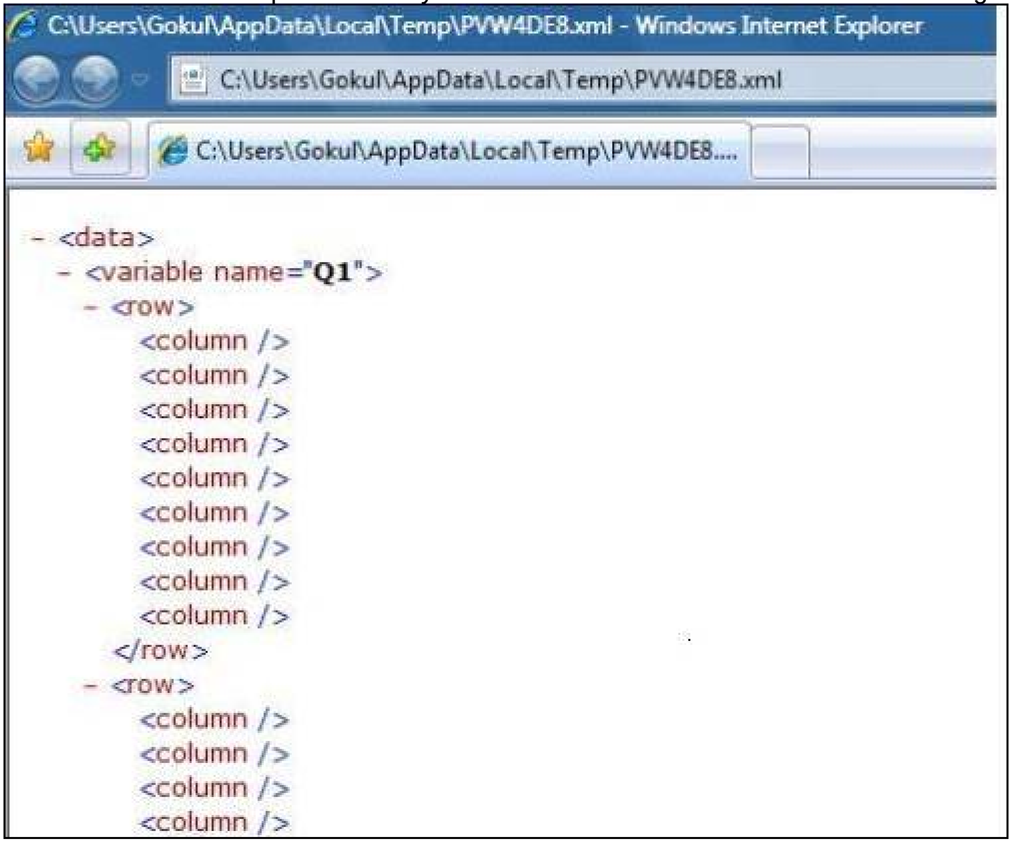
☑ Enable Load

Name: Q1

Range: Sheet1!$A$20:$I$24

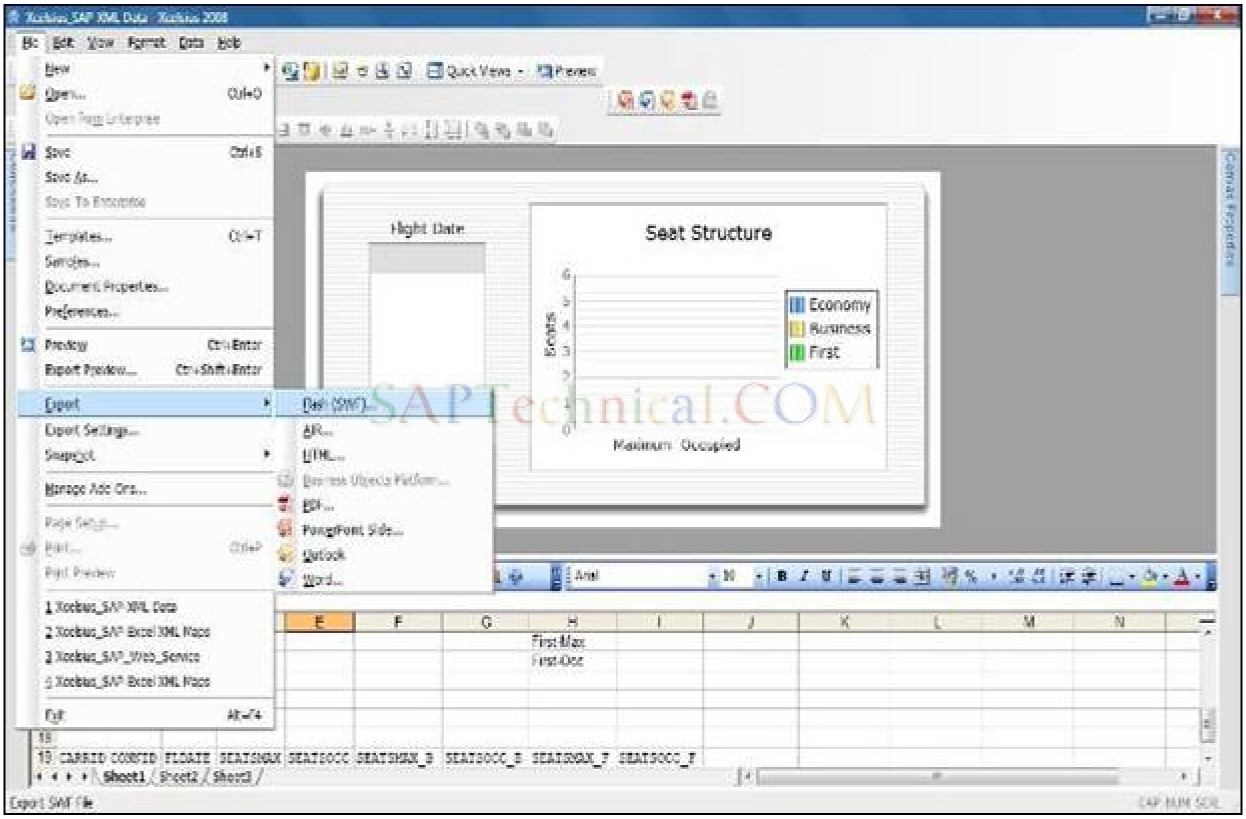
	A	B	C
19	CARRID	CONNID	FLDATE
20			
21			
22			
23			
24			
25			

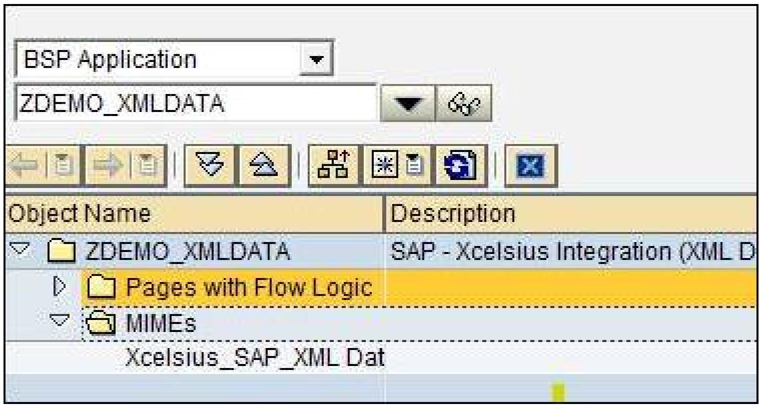
Now the data from the XML has to be mapped to a series of range. Mark check on “Enable Load” and click on “+”. Name the series as “Q1” which we used the variable name in the XML file and select the ranges based on the data. Click on preview XML and compare the XML file generated by Xcelsius preview and the XML generated while testing XML view from BSP application. The format should be the same, so that the data can be parsed easily between the BSP and SWF file which is going to be generated.


C:\Users\Gokul\AppData\Local\Temp\PVW4DE8.xml - Windows Internet Explorer
C:\Users\Gokul\AppData\Local\Temp\PVW4DE8.xml
C:\Users\Gokul\AppData\Local\Temp\PVW4DE8....

```
- <data>
  - <variable name="Q1">
    - <row>
        <column />
        <column />
        <column />
        <column />
        <column />
        <column />
        <column />
        <column />
        <column />
      </row>
    - <row>
        <column />
        <column />
        <column />
        <column />
```

Now export the Xcelsius file as a flash file.



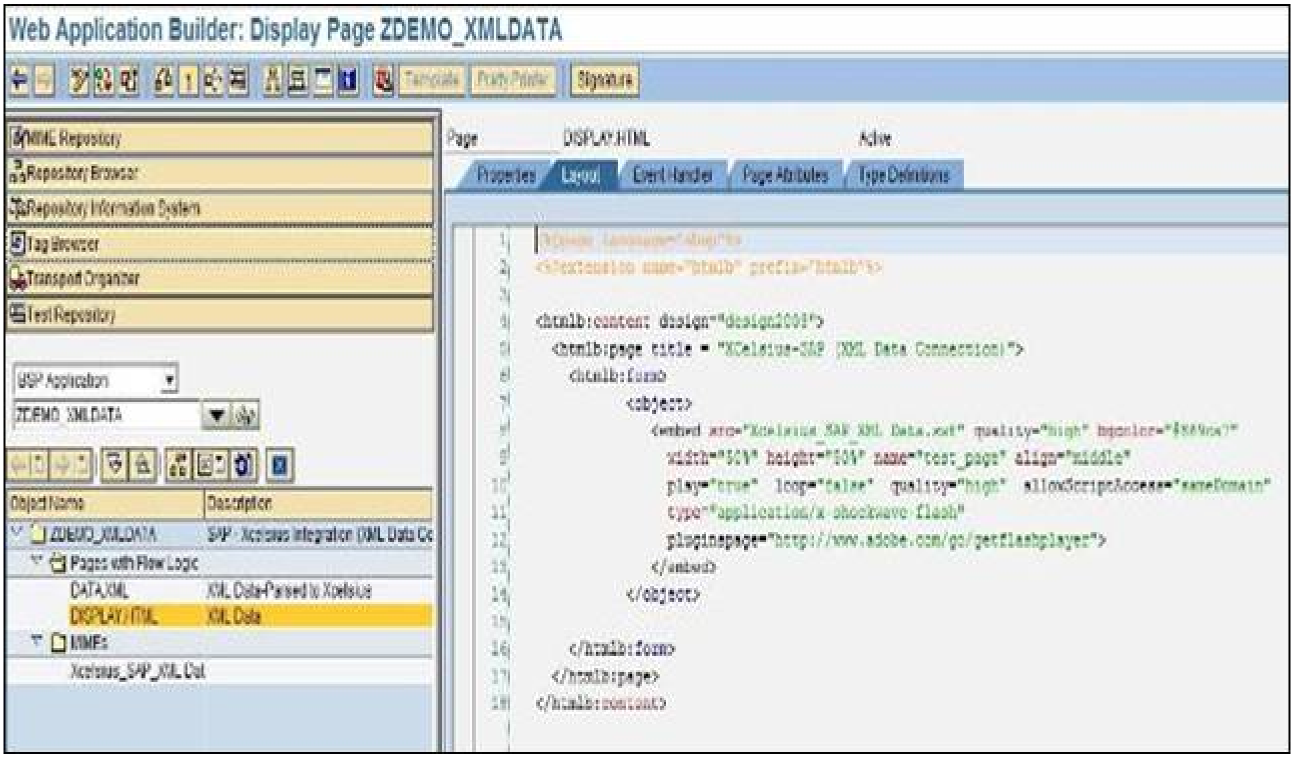


BSP Application

ZDEMO_XMLDATA

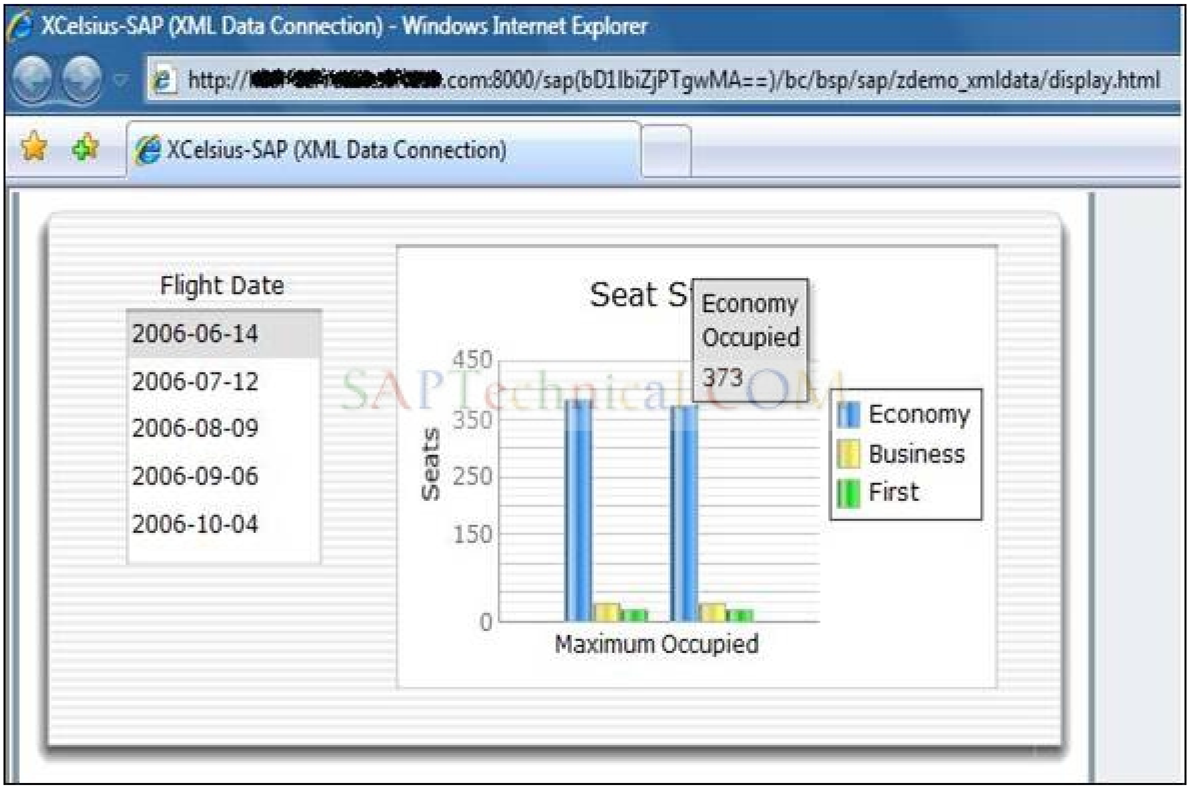
Object Name	Description
ZDEMO_XMLDATA	SAP - Xcelsius Integration (XML D
Pages with Flow Logic	
MIMEs	
Xcelsius_SAP_XML Dat	

Now the BSP application will be enhanced with an HTML page which will be holding SWF file and will be displayed during the run time. Place the below code shown in the screen on the layout of HTML view.



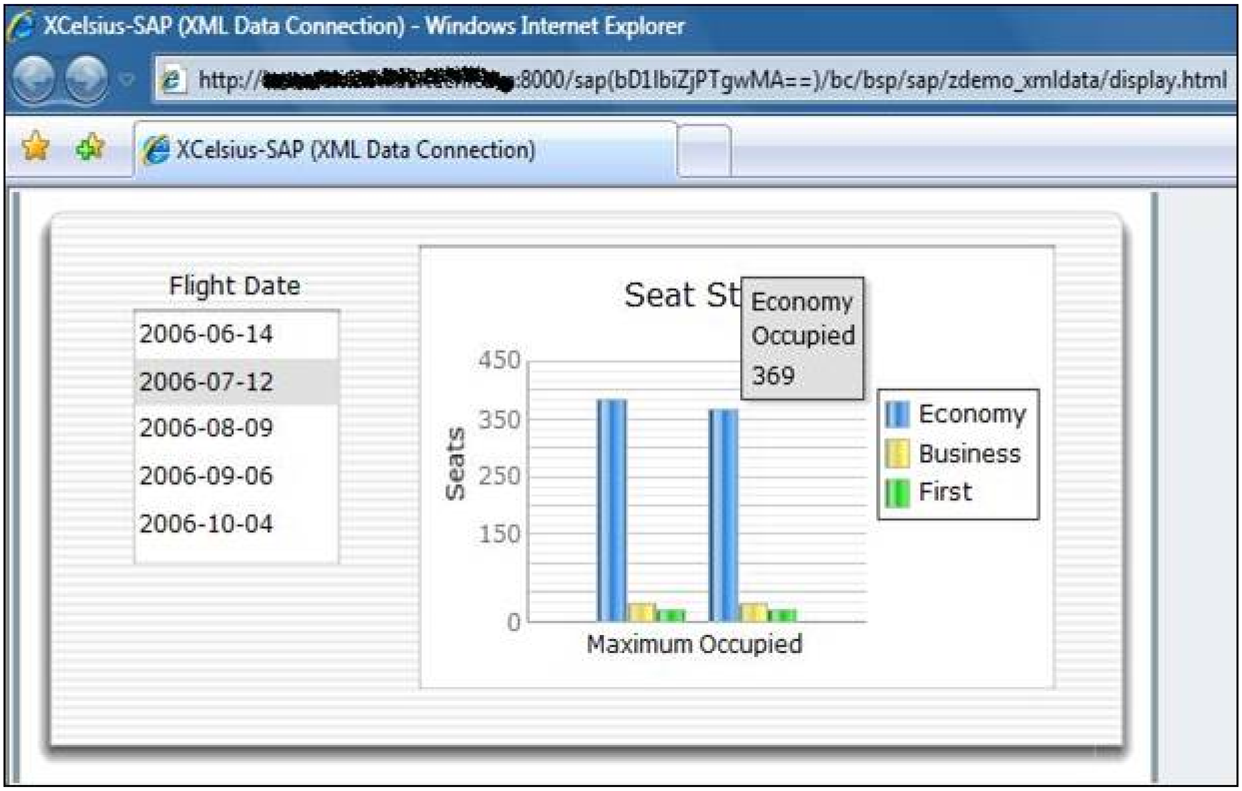
```
<%@page language="abap"%>
<%@extension name="htmlb" prefix="htmlb"%>

<htmlb:content design="design2003">
  <htmlb:page title = "XCelsius-SAP (XML Data Connection)">
    <htmlb:form>
          <object>
            <embed src="Xcelsius_SAP_XML Data.swf" quality="high" bgcolor="#869ca7"
              width="50%" height="50%" name="test_page" align="middle"
              play="true"  loop="false"  quality="high"  allowScriptAccess="sameDomain"
              type="application/x-shockwave-flash"
              pluginspage="http://www.adobe.com/go/getflashplayer">
            </embed>
          </object>

    </htmlb:form>
  </htmlb:page>
</htmlb:content>
```

Now test the HTML view and SWF file will be generated. Based on the different selections on flight date the data will be changed as shown in the below screens.



XCelsius-SAP (XML Data Connection) - Windows Internet Explorer

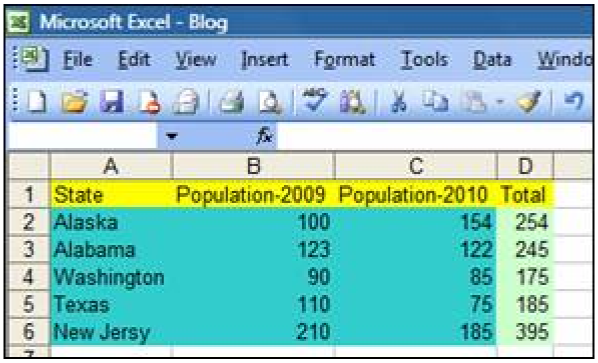
http://[redacted].com:8000/sap(bD1IbiZjPTgwMA==)/bc/bsp/sap/zdemo_xmldata/display.html


XCelsius-SAP (XML Data Connection) - Windows Internet Explorer
http://[redacted]:8000/sap(bD1lbiZjPTgwMA==)/bc/bsp/sap/zdemo_xmldata/display.html
XCelsius-SAP (XML Data Connection)
Flight Date
2006-06-14
2006-07-12
2006-08-09
2006-09-06
2006-10-04
Seat St
Economy
Occupied
369
Seats
450
350
250
150
0
Maximum Occupied
Economy
Business
First

Integrating BO-Xcelsius (SWF) file with BO-Crystal Reports

This document is intended to display formatted report (Crystal Report) where the Xcelsius file will be embedded within the Crystal report. Only Crystal 2008 will support embedding Flash file. Further the Xcelsius file (SWF) present within the crystal report can be operated within the same page. Here an excel sheet will be acting as the database for the crystal report. Database for the crystal Report can be anything like SAP R/3 or BW data, etc.

Create an Excel database with below data and save it in local disk.

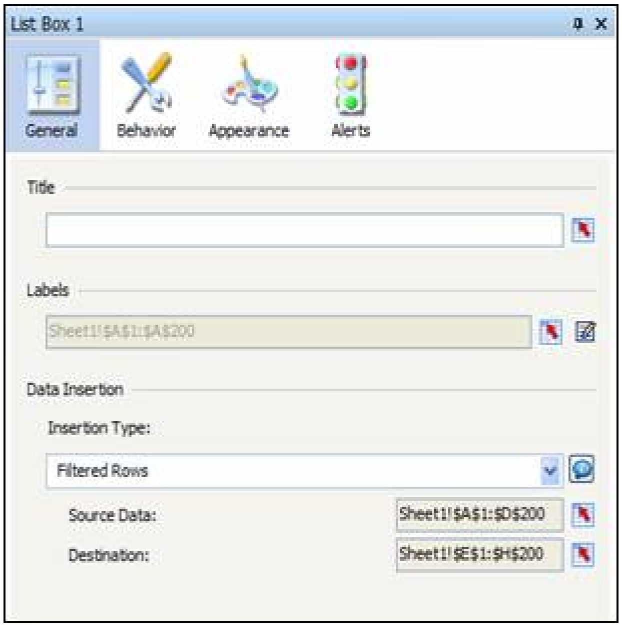


	A	B	C	D
1	State	Population-2009	Population-2010	Total
2	Alaska	100	154	254
3	Alabama	123	122	245
4	Washington	90	85	175
5	Texas	110	75	185
6	New Jersy	210	185	395

Now the Xcelsius file has to be designed and SWF file will be generated. Now open the Xcelsius and place a List Box, Column chart and a Gauge in the Canvas. The List box will be displaying all the States, Column Chart will be used to display the Population on 2009 and 2010 of the selected state; and Gauge will be pointing the total population of the selected state.

Select the List Box and apply the below properties.

- Map the Label ranges by selecting the “A” column.
- Select Insertion type as “Filtered Rows”
- Map the Source Data ranges by selecting “A”, “B”, “C” and “D” columns.
- Map the Destination ranges by selecting “E”, “F”, “G” and “H” columns.



Title

Labels

Sheet1!$A$1:$A$200

Data Insertion

Insertion Type:

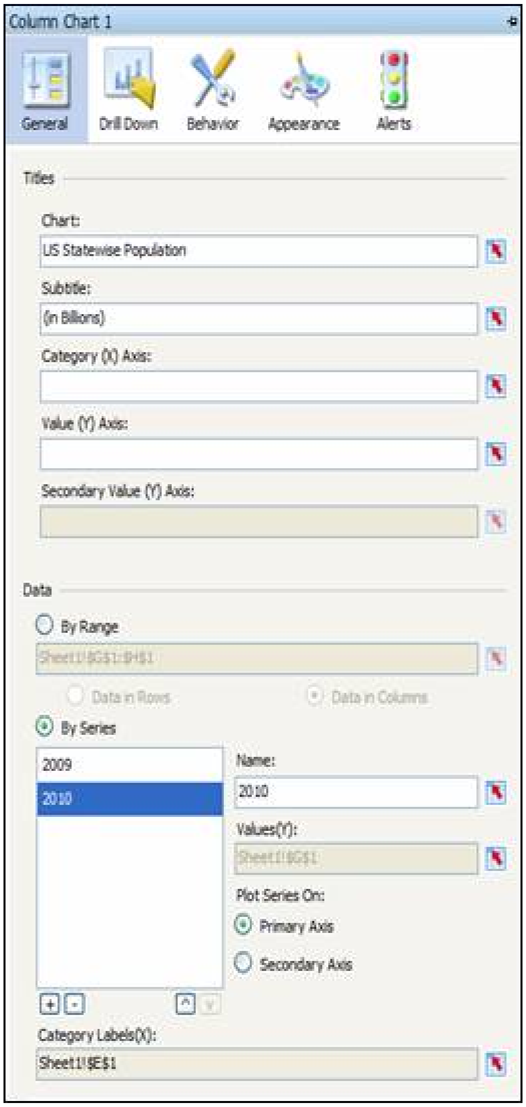
Filtered Rows

Source Data: Sheet1!$A$1:$D$200

Destination: Sheet1!$E$1:$H$200

Select the Column Chart and apply the below properties.

- Create 2 Series and Name it as “2009” and “2010”
- Select 2009 series and Map cell “F1” to values(Y)
- Select 2010 series and Map cell “G1” to values(Y)
- Map the Category Labels (X) with cell “E1”



Titles

Chart: US Statewise Population

Subtitle: (in Billions)

Category (X) Axis:

Value (Y) Axis:

Secondary Value (Y) Axis:

Data

○ By Range

Sheet1!$G$1:$H$1

○ Data in Rows ● Data in Columns

● By Series

2009

2010

Name: 2010

Values(Y): Sheet1!$G$1

Plot Series On:

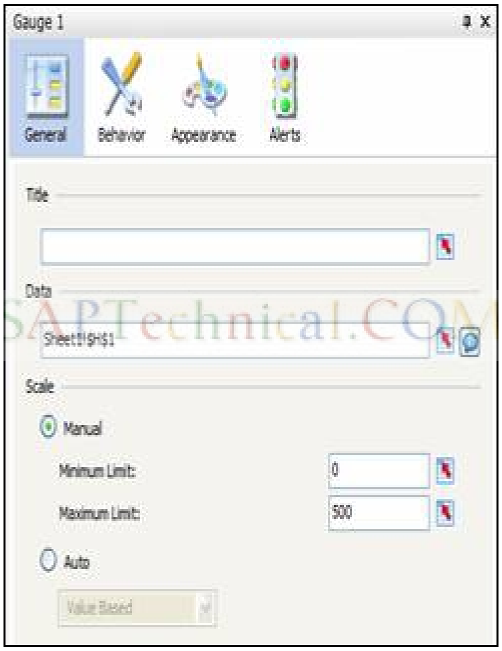
● Primary Axis

○ Secondary Axis

Category Labels(X): Sheet1!$E$1

Select the Gauge and follow the below properties.

- Map the Data with cell “H1”
- Change Maximum Limit to “500”
- In the Alerts tab, Check “Enable Alerts” and change “As Percent of Target” to 500



Title

Data

Sheet1!$H$1

Scale

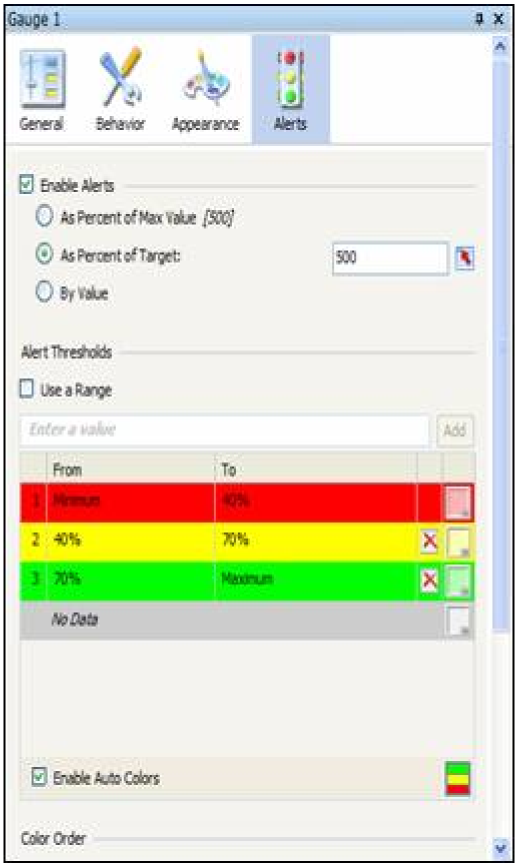
● Manual

Minimum Limit: 0

Maximum Limit: 500

○ Auto

Value Based



☑ Enable Alerts

○ As Percent of Max Value *[500]*

● As Percent of Target: 500

○ By Value

Alert Thresholds

☐ Use a Range

Enter a value Add

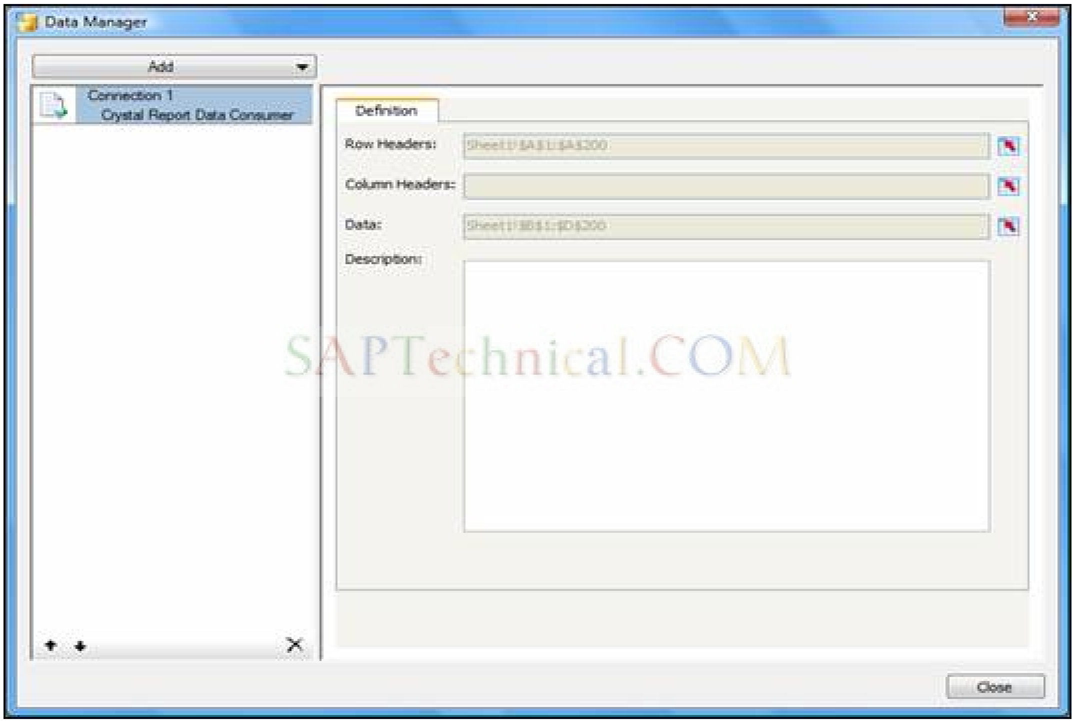
	From	To
1	Minimum	40%
2	40%	70%
3	70%	Maximum
	No Data	

☑ Enable Auto Colors

Color Order

Open Data Manger and add a “Crystal Report Data Consumer” connection and map the cells as below.

- Map the Row Header ranges by selecting the “A” column.
- Map the Data ranges by selecting the “B” “C” and “D” columns.



Row Headers: Sheet1!$A$1:$A$200

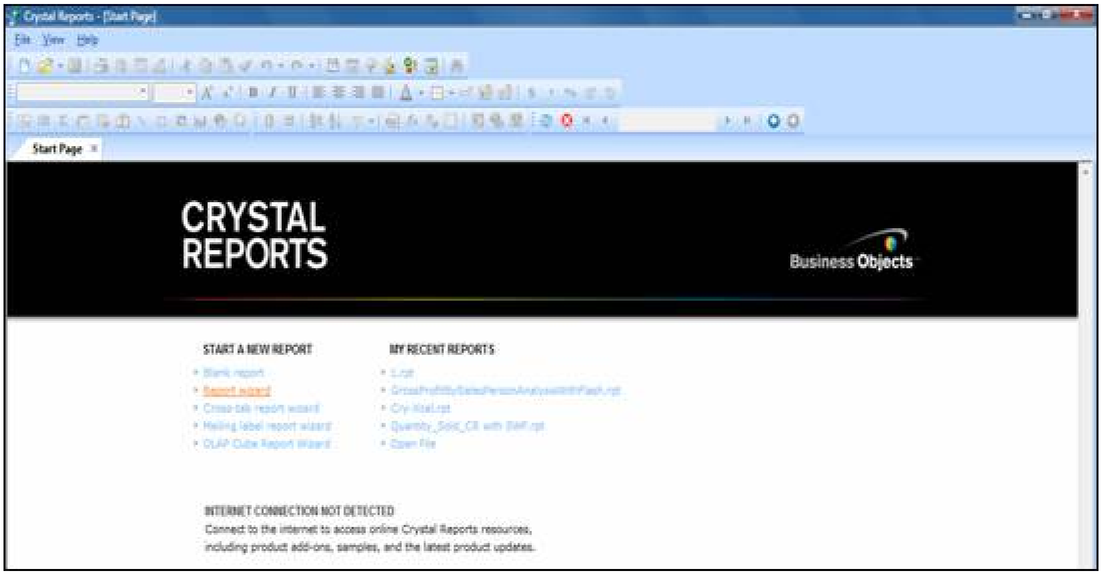
Column Headers:

Data: Sheet1!$B$1:$D$200

Description:

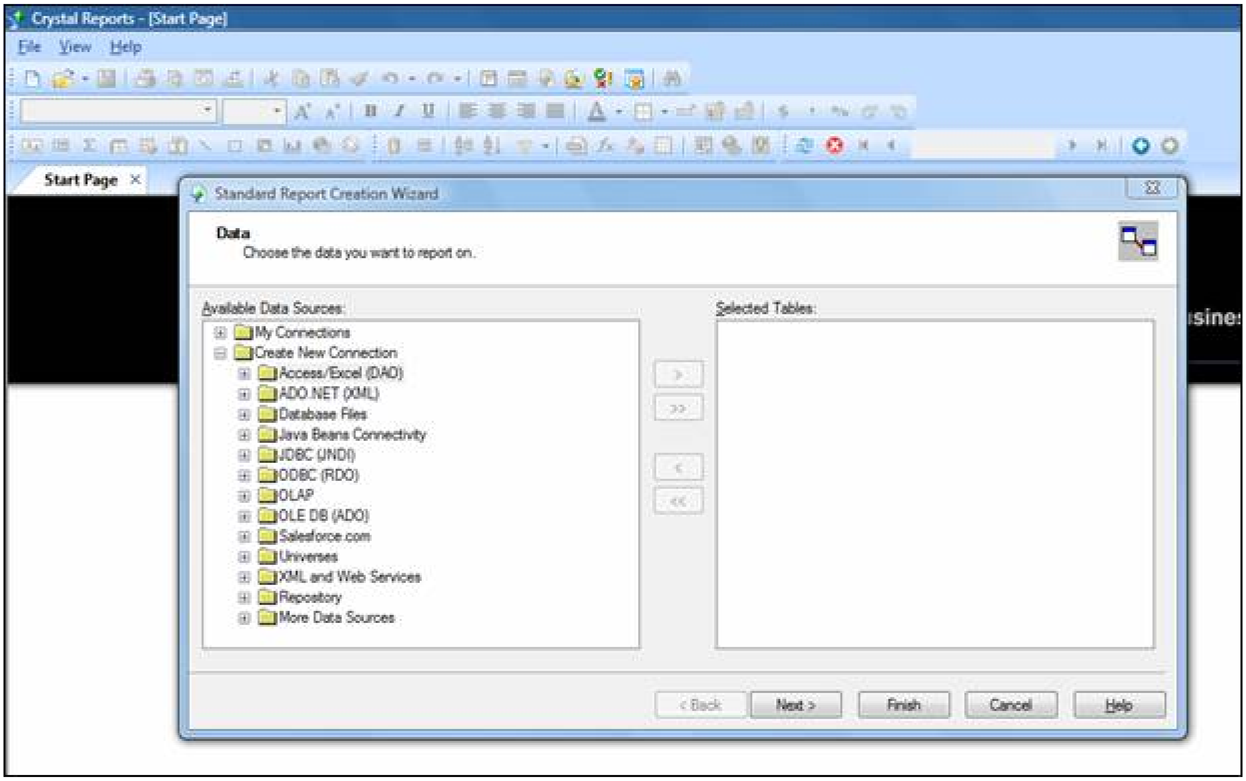
Save the Xcelsius file (.XLF) and export the Xcelsius file as SWF.

Open Crystal Reports 2008 and select Report Wizard.

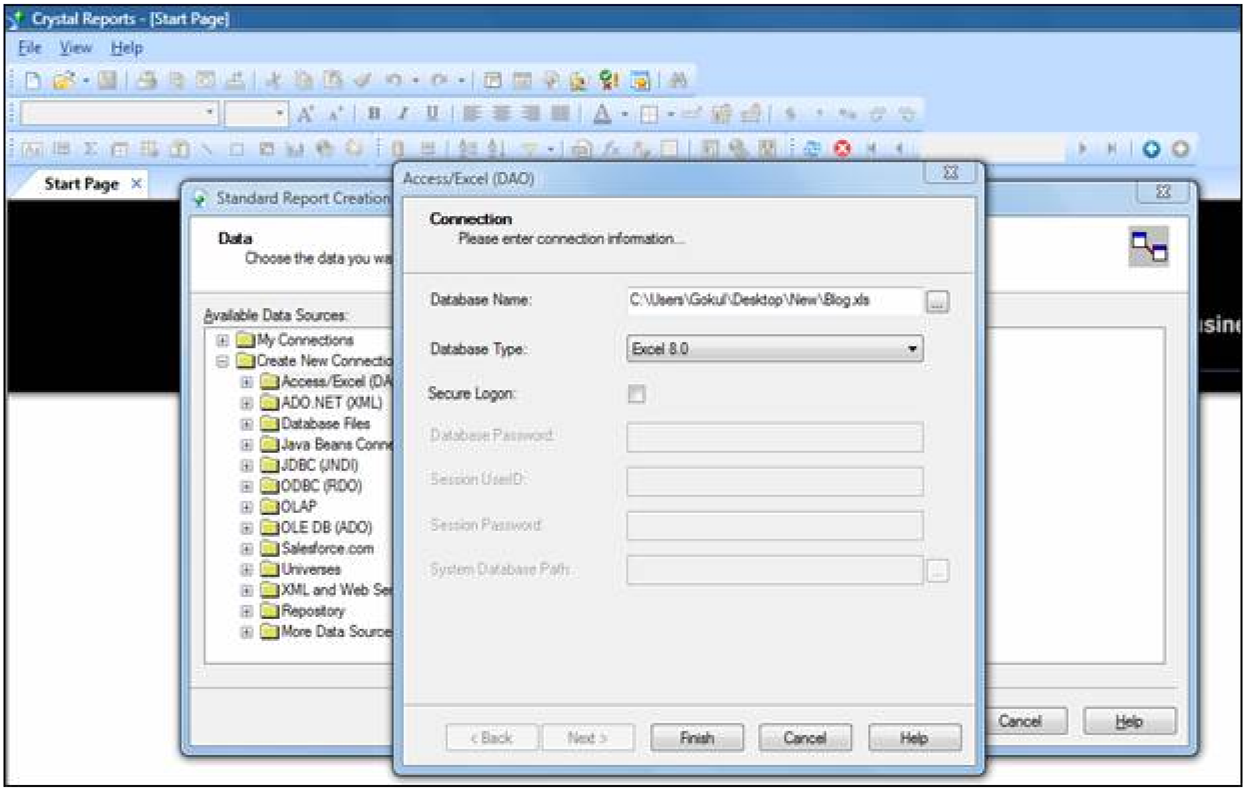


INTERNET CONNECTION NOT DETECTED
Connect to the internet to access online Crystal Reports resources, including product add-ons, samples, and the latest product updates.

Now a new connection has to be opened by keeping the Excel file which we mentioned above will be acting as the database.



We will select “Access/Excel” as the new connection and mention the path of the excel file by changing the Database Type as “Excel 8.0



Database Name: C:\Users\Gokul\Desktop\New\Blog.xls

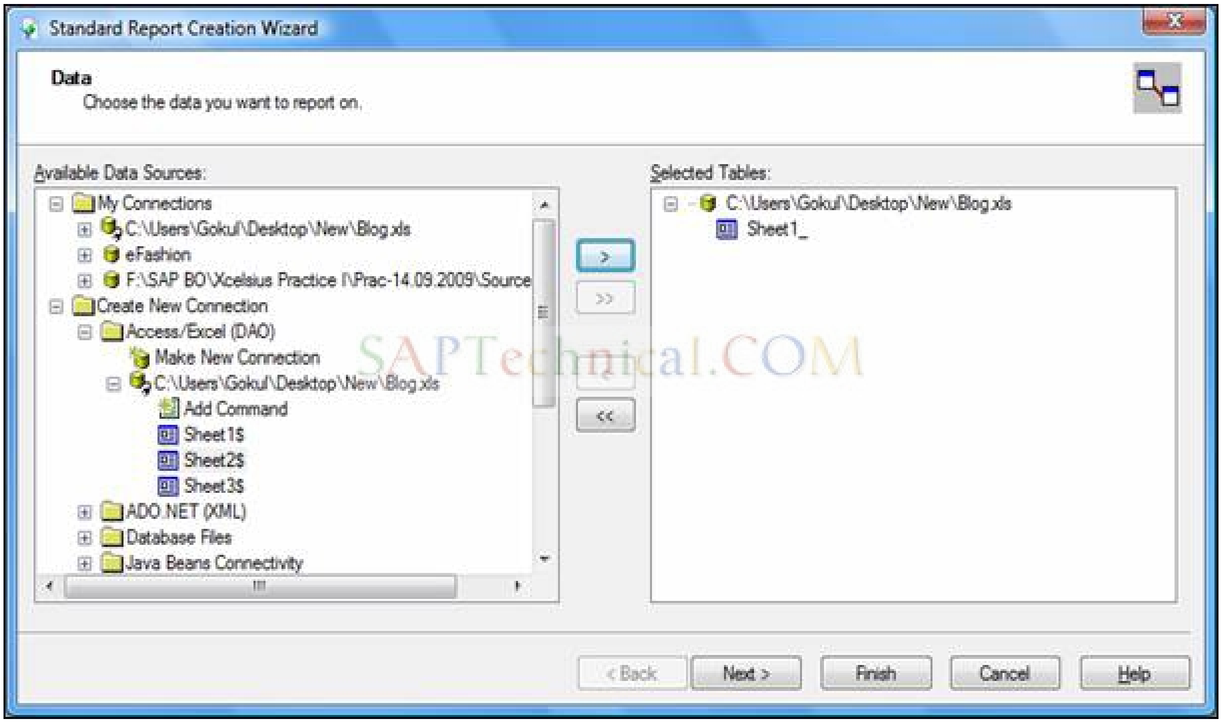
Database Type: Excel 8.0

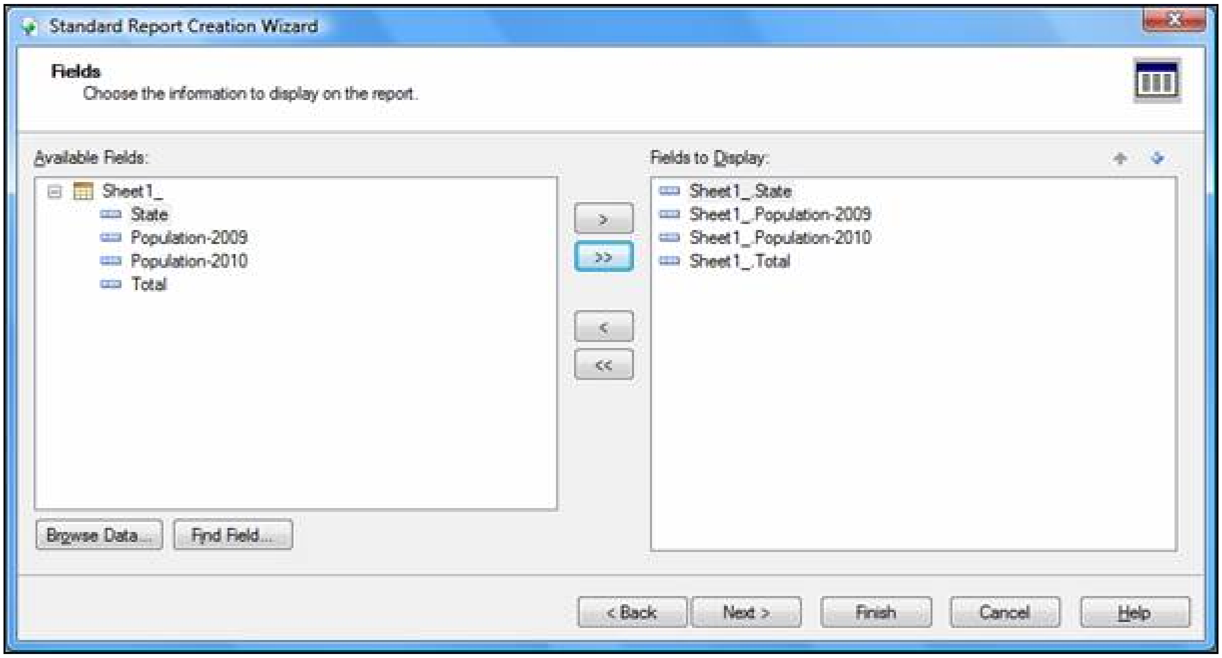
Secure Logon: □

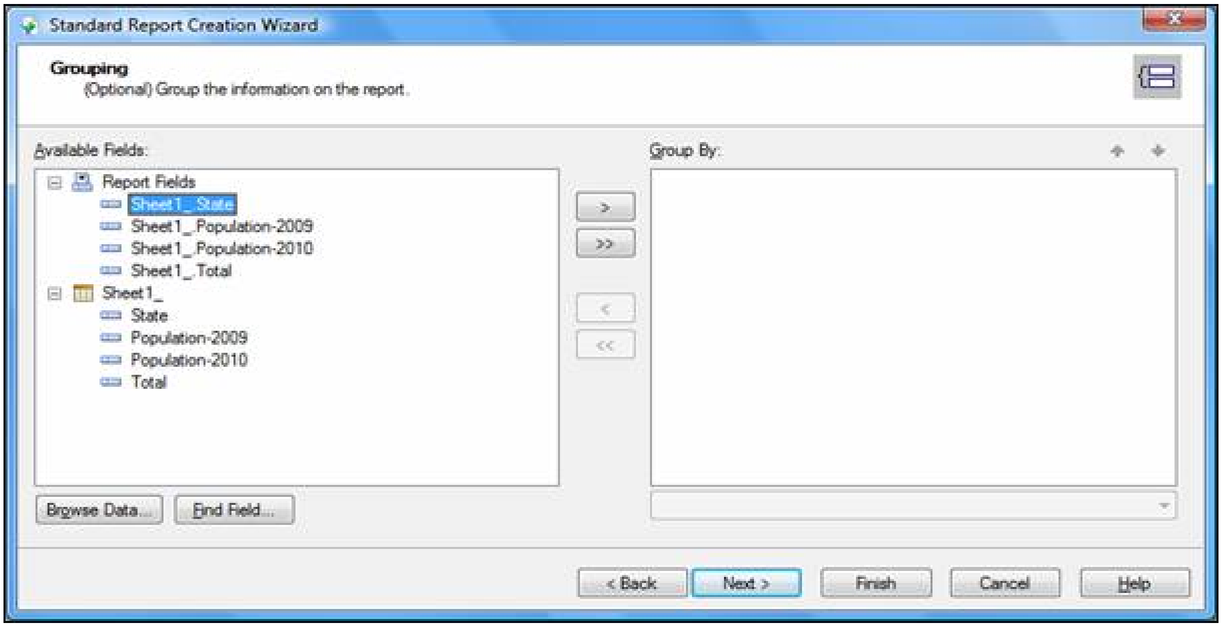
By clicking on “Finish” several properties can be assigned in the following steps by clicking on “Next”, like

- What are the fields to be displayed in the report
- Template to be used
- Selecting fields if summary is needed and etc

Follow the below screen and assign the properties.







Standard Report Creation Wizard

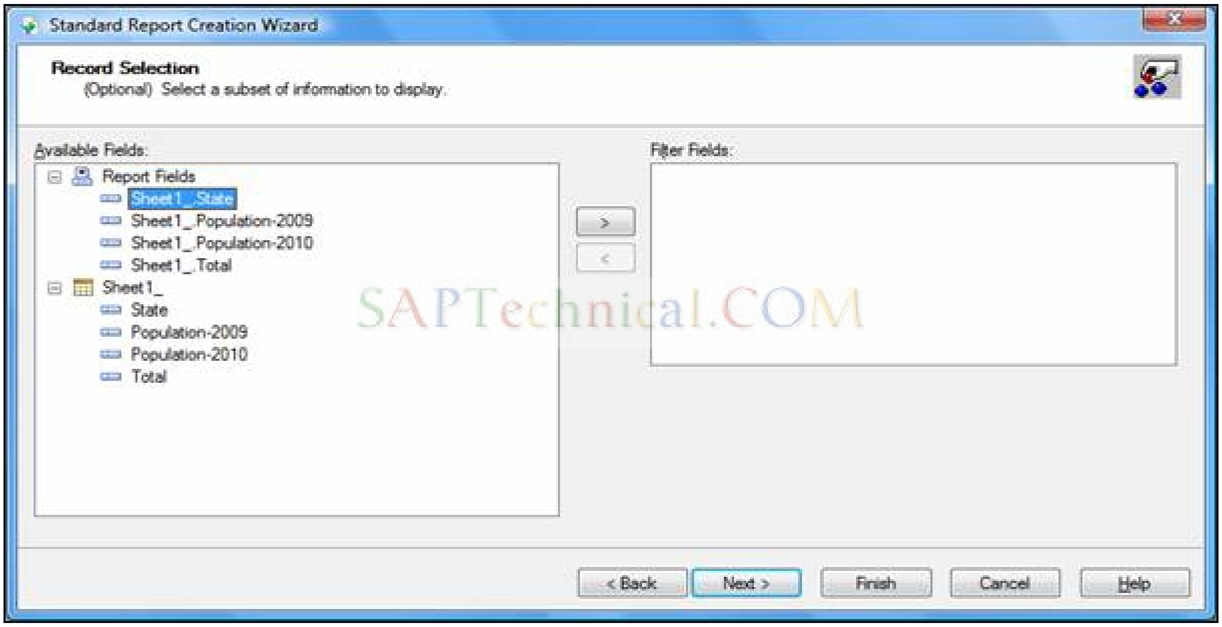
Grouping

(Optional) Group the information on the report.

Available Fields:

- Report Fields
  - Sheet1_.State
  - Sheet1_.Population-2009
  - Sheet1_.Population-2010
  - Sheet1_.Total
- Sheet1_
  - State
  - Population-2009
  - Population-2010
  - Total

Group By:

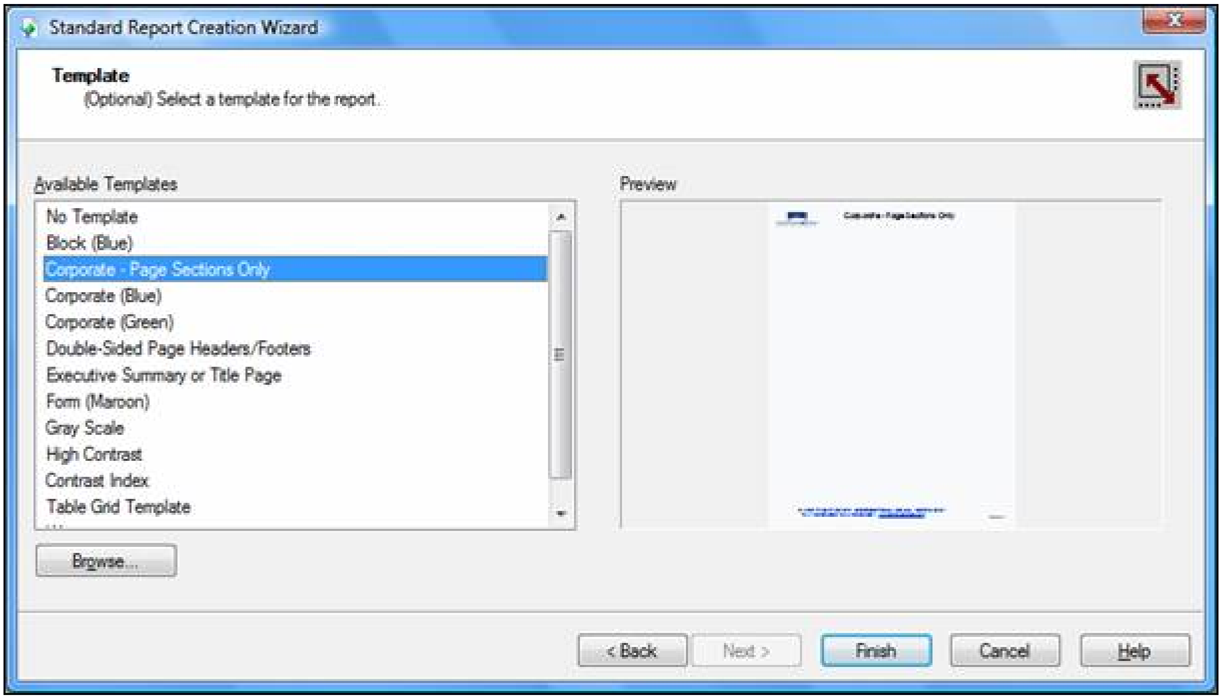

Standard Report Creation Wizard

Record Selection
(Optional) Select a subset of information to display.

Available Fields:

- Report Fields
  - Sheet1_.State
  - Sheet1_.Population-2009
  - Sheet1_.Population-2010
  - Sheet1_.Total
- Sheet1_
  - State
  - Population-2009
  - Population-2010
  - Total

Filter Fields:


Standard Report Creation Wizard

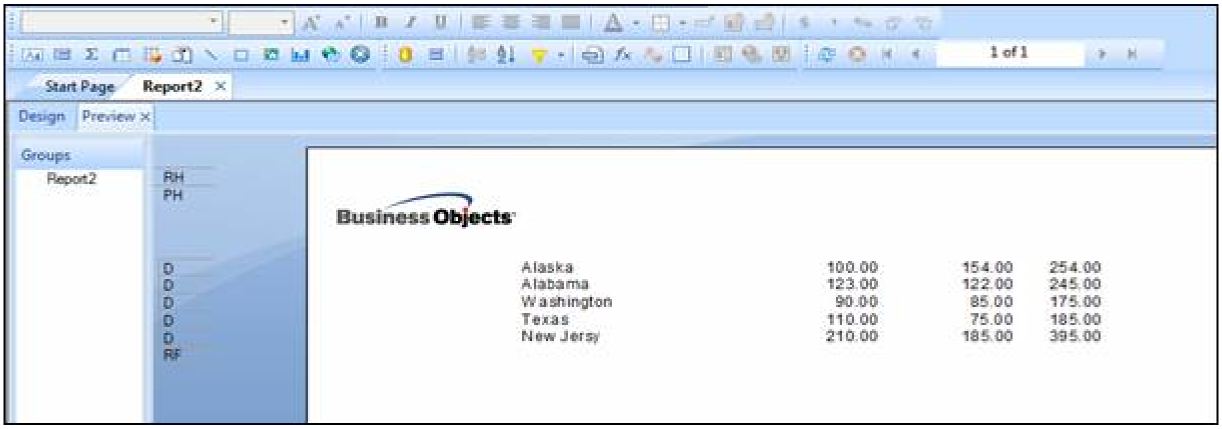
Template
(Optional) Select a template for the report.

Available Templates

- No Template
- Block (Blue)
- Corporate - Page Sections Only
- Corporate (Blue)
- Corporate (Green)
- Double-Sided Page Headers/Footers
- Executive Summary or Title Page
- Form (Maroon)
- Gray Scale
- High Contrast
- Contrast Index
- Table Grid Template

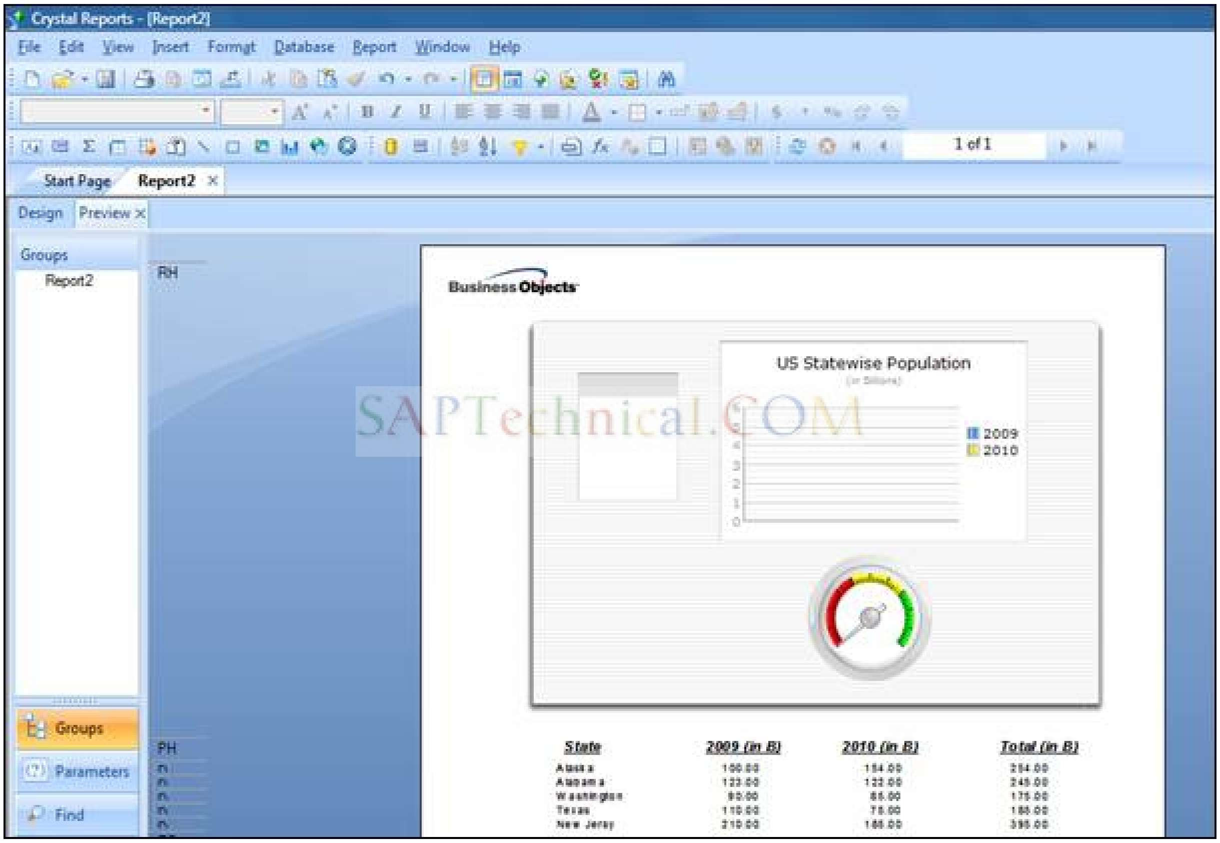
Preview

While Clicking in “Finish” following screen displayed on Preview mode.



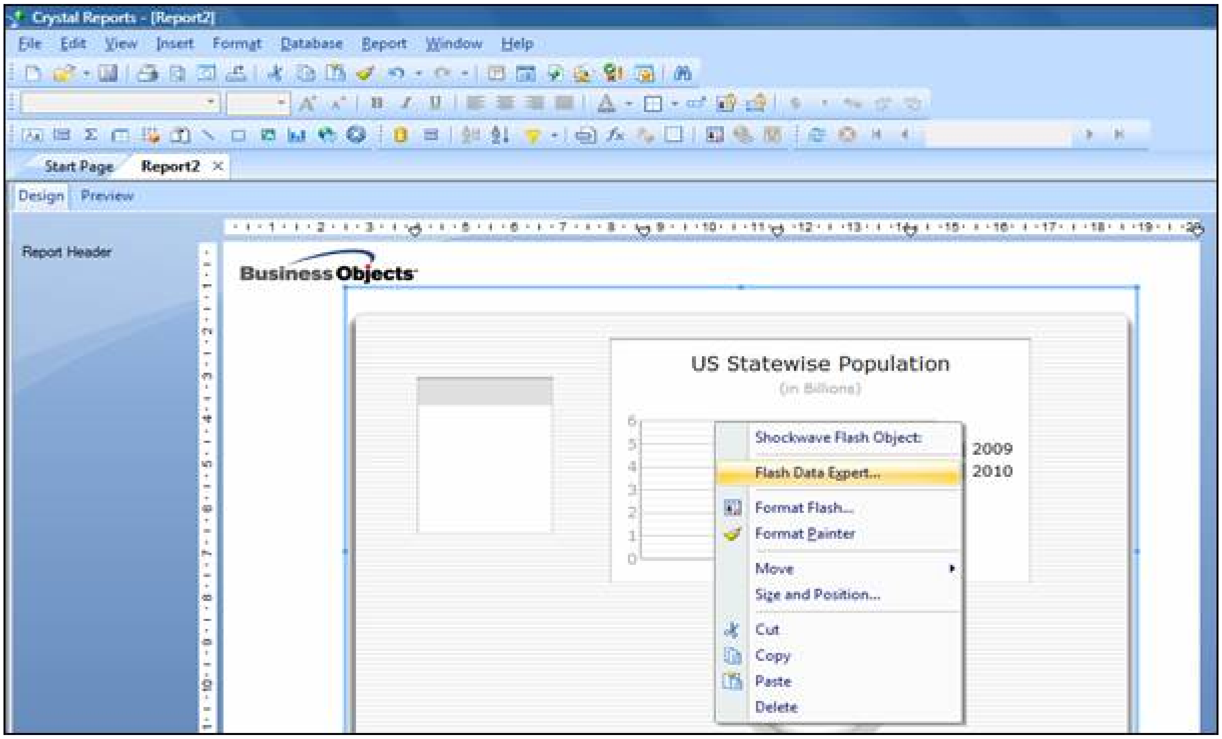
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00

Now the Xcelsius (SWF) file has to be integrated and based on the users need, design has to be done in the formatted report. Go to design mode and click on **INSERT ->** FLASH and select the SWF file and place it in the crystal report.

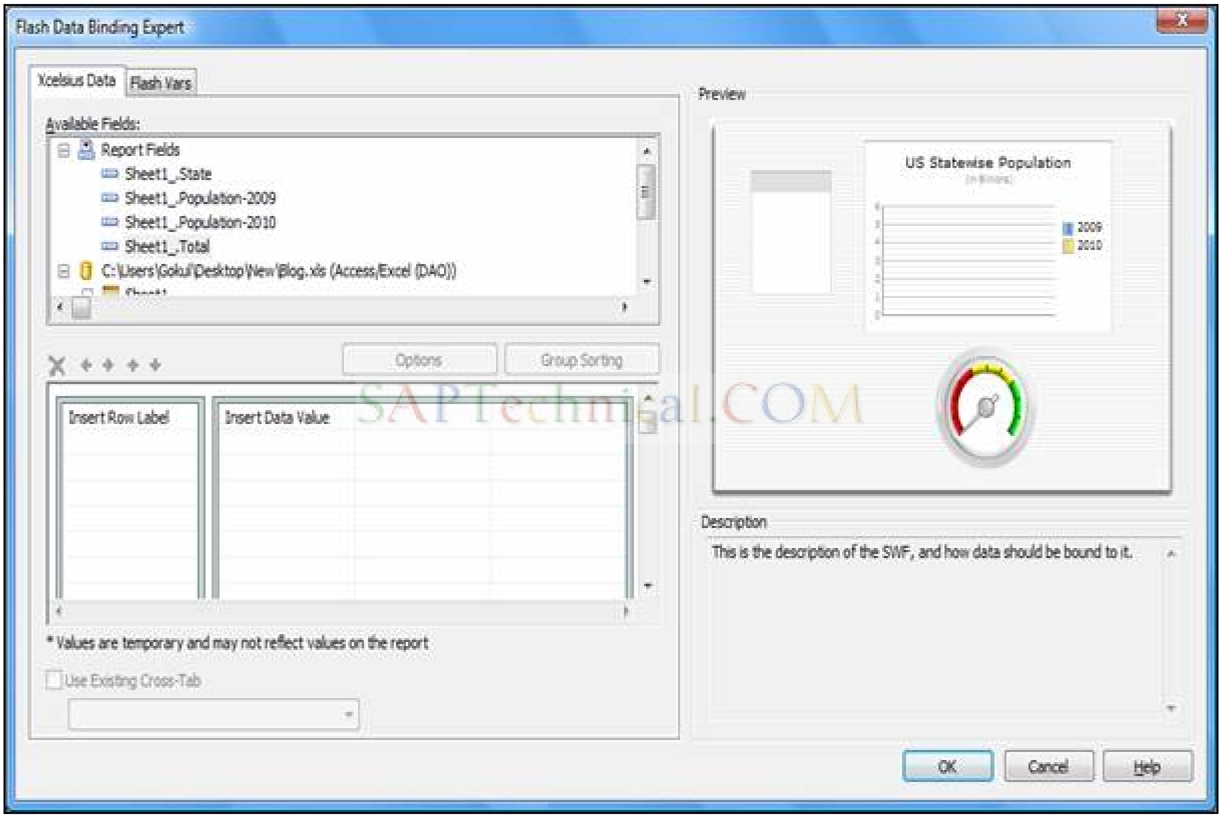


State	2009 (in B)	2010 (in B)	Total (in B)
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00

Now the values has to be mapped to which will be from the database which we are using for the crystal report. Right click on the SWF file and select “Flash Data Expert”.



Now the values have to be mapped for the SWF file.

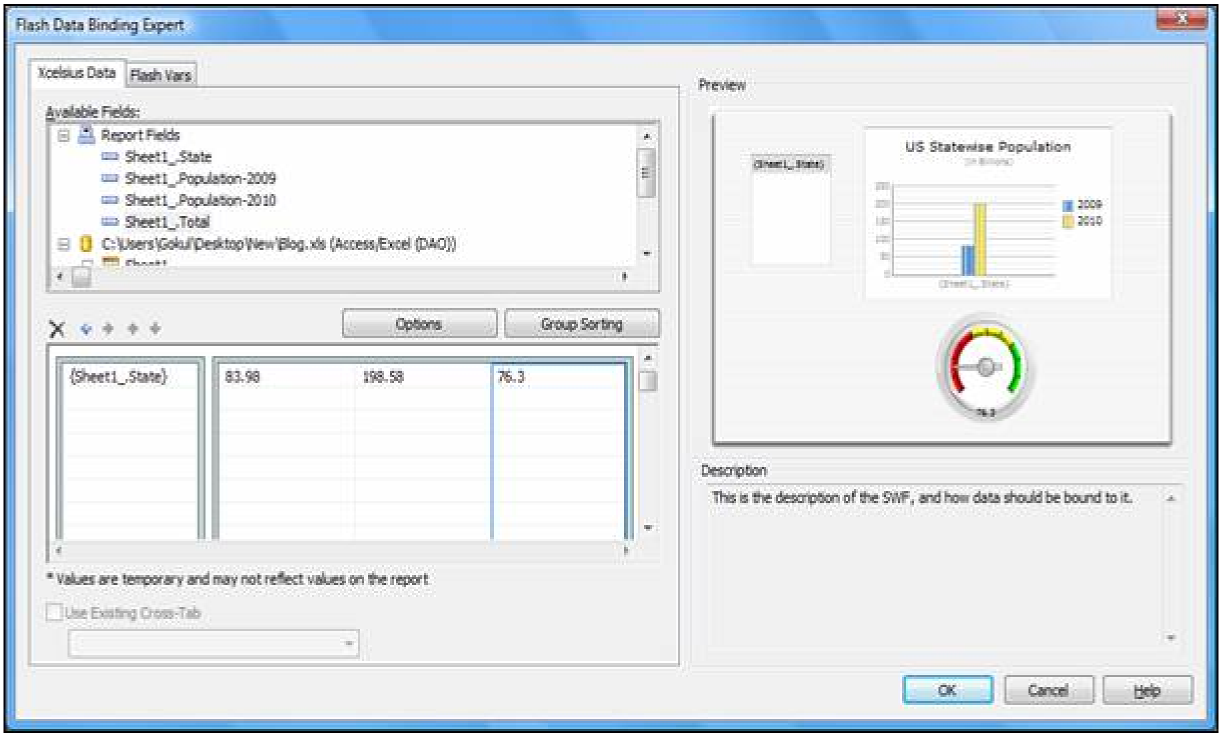


* Values are temporary and may not reflect values on the report

This is the description of the SWF, and how data should be bound to it.

- Drag and Drop Sheet1_.State field to Insert row Label

- Drag and Drop Sheet1_.Population-2009, Sheet1_.Population-2010 and total fields to Insert Data value fields.



Available Fields:

- Report Fields
  - Sheet1_.State
  - Sheet1_.Population-2009
  - Sheet1_.Population-2010
  - Sheet1_.Total
- C:\Users\Gokul\Desktop\New\Blog.xls (Access/Excel (DAO))

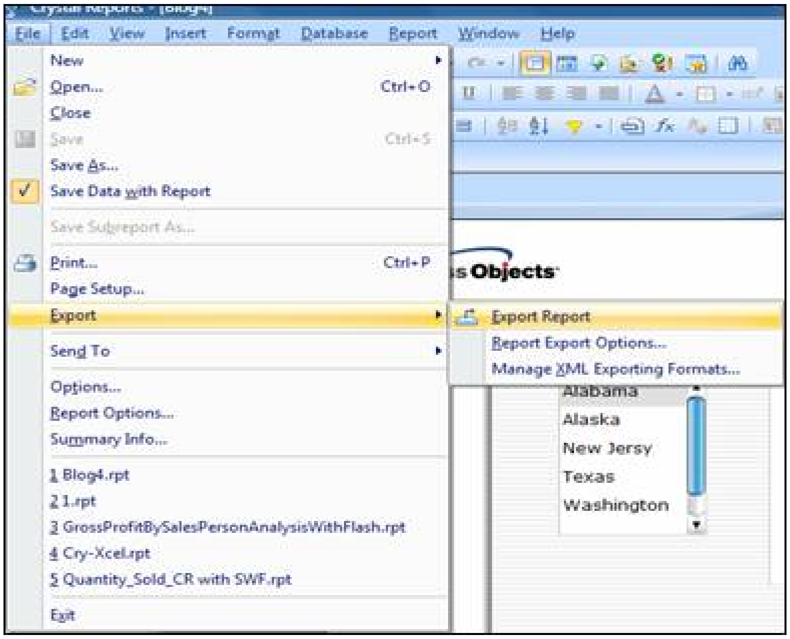
{Sheet1_.State}	83.98	198.58	76.3

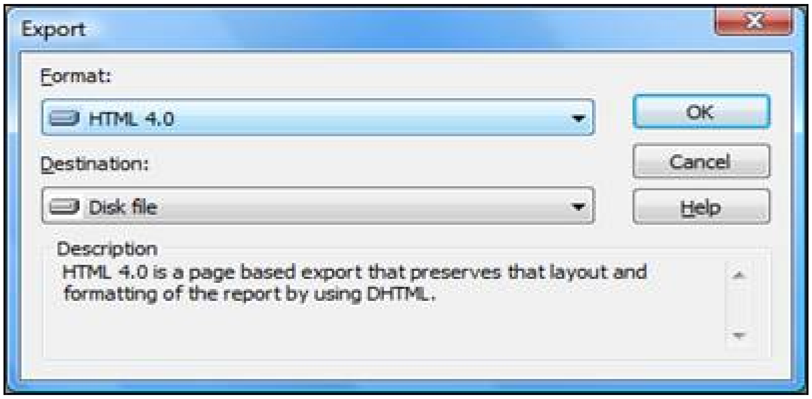
* Values are temporary and may not reflect values on the report

□ Use Existing Cross-Tab

Description: This is the description of the SWF, and how data should be bound to it.

Now by clicking on “Ok” save the crystal report. The crystal report can be exported to “PDF” or “HTML” so that the report can be visualized interactively.



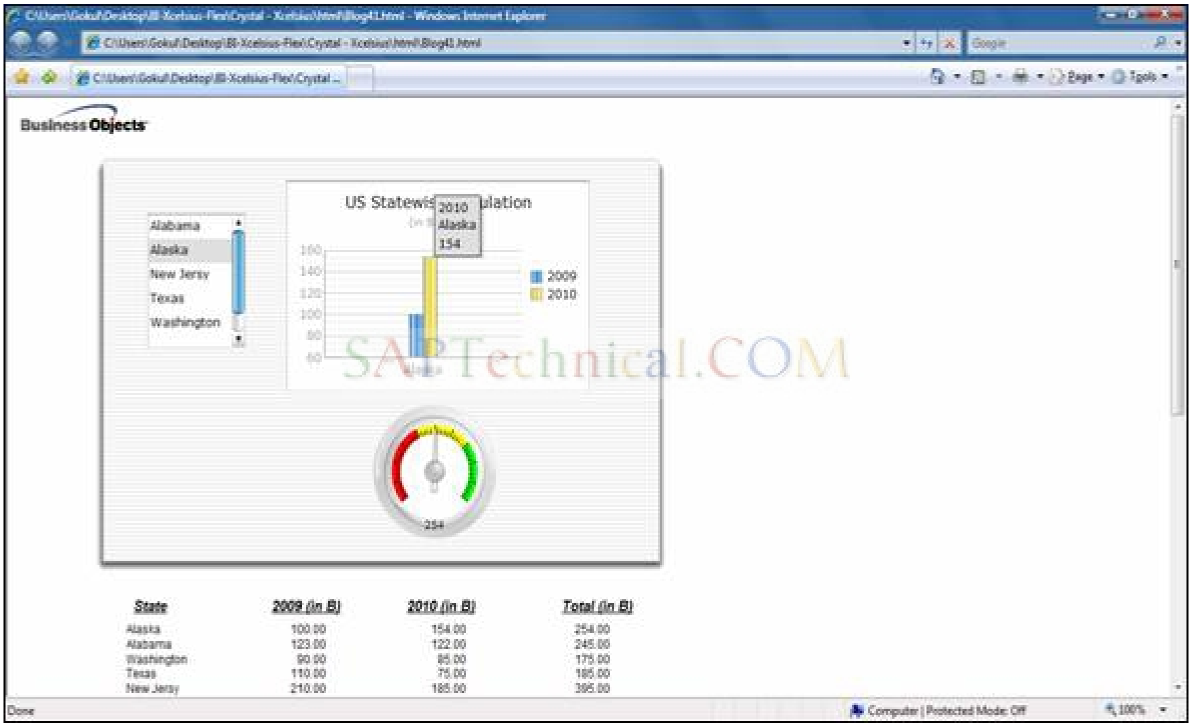


Format: HTML 4.0

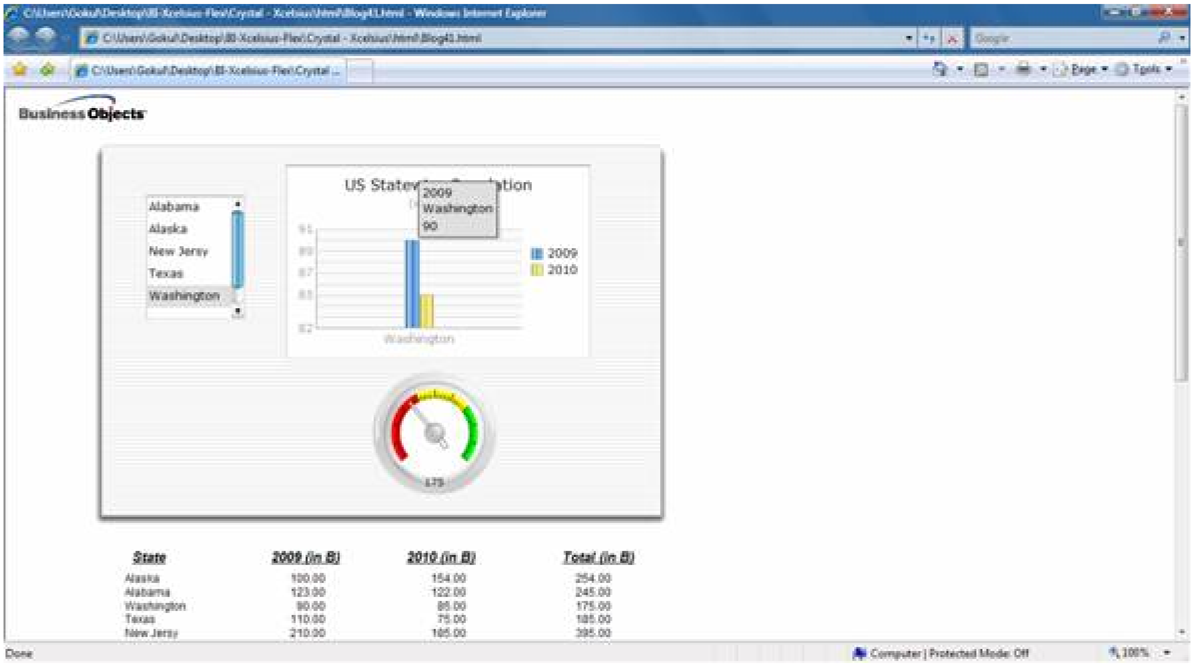
Destination: Disk file

Description
HTML 4.0 is a page based export that preserves that layout and formatting of the report by using DHTML.

From the above screen mention the path and export the report in HTML 4.0. Open the HTML file from the exported path and the report will generate as below.


C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html - Windows Internet Explorer
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html
Google
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal ...
Page
Tools
Business Objects
Alabama
Alaska
New Jersy
Texas
Washington
US Statewise Population
2010
Alaska
154
160
140
120
100
80
60
2009
2010
SAPTechnical.COM
254
Done
Computer | Protected Mode: Off
100%

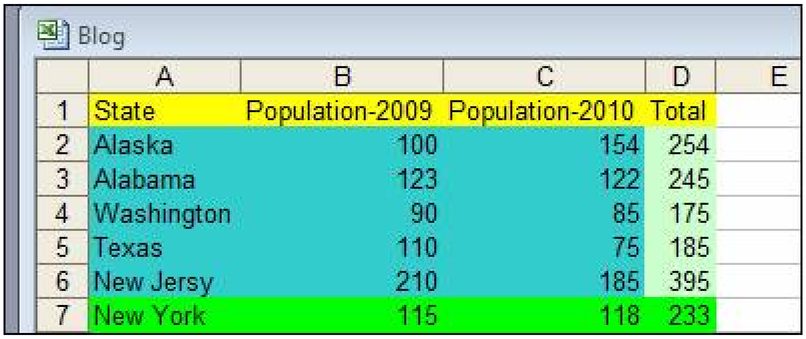
State	2009 (in B)	2010 (in B)	Total (in B)
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00



C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html - Windows Internet Explorer
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal ...
Page
Tools
Business Objects
Alabama
Alaska
New Jersy
Texas
Washington
US State ... tion
2009
Washington
90
91
89
87
85
82
Washington
2009
2010
175
Done
Computer | Protected Mode: Off
100%

State	2009 (in B)	2010 (in B)	Total (in B)
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00

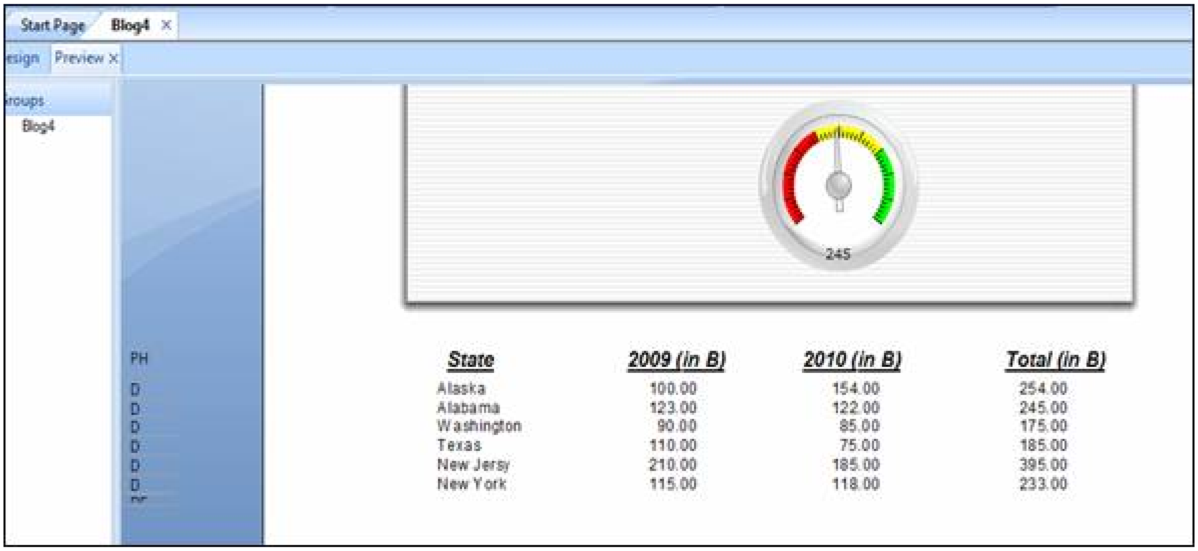
When the database (Excel Sheet) is updated with more records, by clicking on “Refresh Data” icon 🔄 in crystal report; the data will be updated and affected in the crystal report. The Xcelsius file will also be affected and displayed in the same manner as of crystal report.



Blog

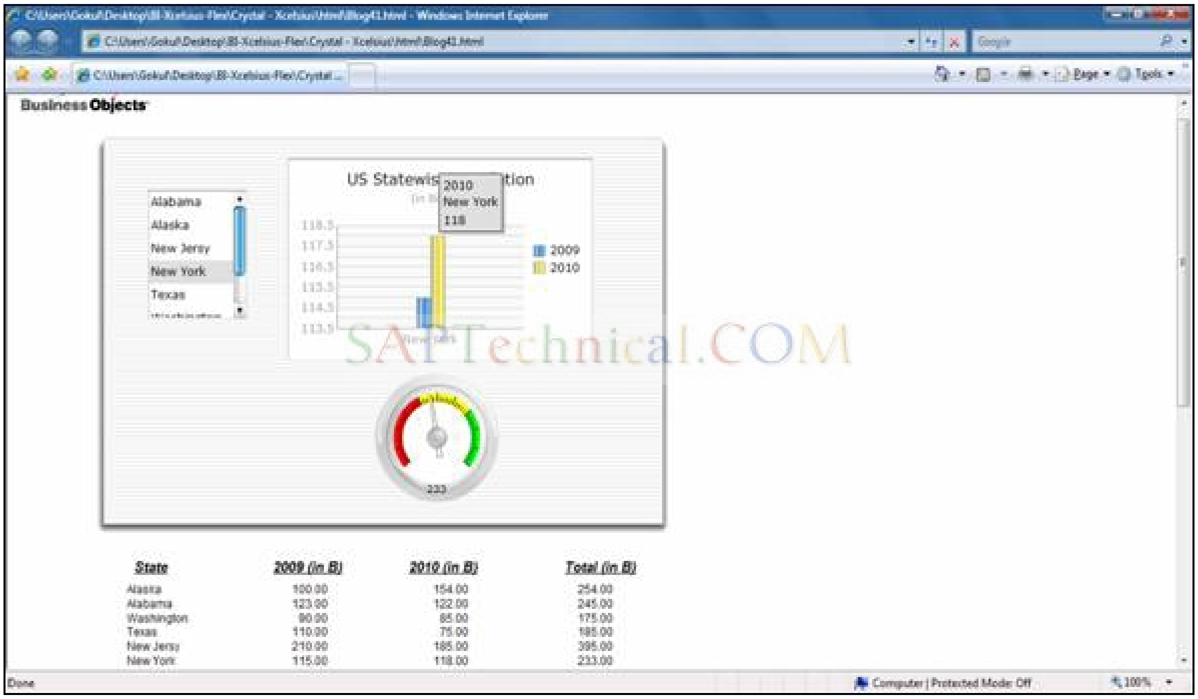
	A	B	C	D	E
1	State	Population-2009	Population-2010	Total	
2	Alaska	100	154	254	
3	Alabama	123	122	245	
4	Washington	90	85	175	
5	Texas	110	75	185	
6	New Jersy	210	185	395	
7	New York	115	118	233	

By refreshing Crystal report



State	2009 (in B)	2010 (in B)	Total (in B)
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00
New York	115.00	118.00	233.00

Save the Crystal report and export the same in HTML 4.0 format.


C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html - Windows Internet Explorer
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal - Xcelsius\html\Blog41.html
Google
C:\Users\Gokul\Desktop\BI-Xcelsius-Flex\Crystal ...
Page
Tools
Business Objects
Alabama
Alaska
New Jersy
New York
Texas
US Statewis
tion
2010
New York
118
118.5
117.5
116.5
115.5
114.5
113.5
New York
2009
2010
233
Done
Computer | Protected Mode: Off
100%

State	2009 (in B)	2010 (in B)	Total (in B)
Alaska	100.00	154.00	254.00
Alabama	123.00	122.00	245.00
Washington	90.00	85.00	175.00
Texas	110.00	75.00	185.00
New Jersy	210.00	185.00	395.00
New York	115.00	118.00	233.00