How to Add Columns to Tables in Standard Supply Network Collaboration (SNC) Web Interface

Applies to:

SAP Supply Network Collaboration (SNC) 5.1 For more information, visit the Supply Chain Management homepage.

Summary

This document details the procedure of adding columns to tables in standard SNC Web UI. It is intended for SAP technical consultants. It is expected that the reader have knowledge of ABAP, especially BADIs.

Author: Ashwin Bhat

Company: L&T Infotech, Mumbai, India

Created on: 9 September 2009

Author Bio

Ashwin Bhat is a SAP NetWeaver consultant working with L&T Infotech, Mumbai, India. His SAP experience of over five years spans various technologies such as BW, ABAP, WebDynpro for Java and Process Integration (XI3.0, PI7.0, and PI7.1).

Table of Contents

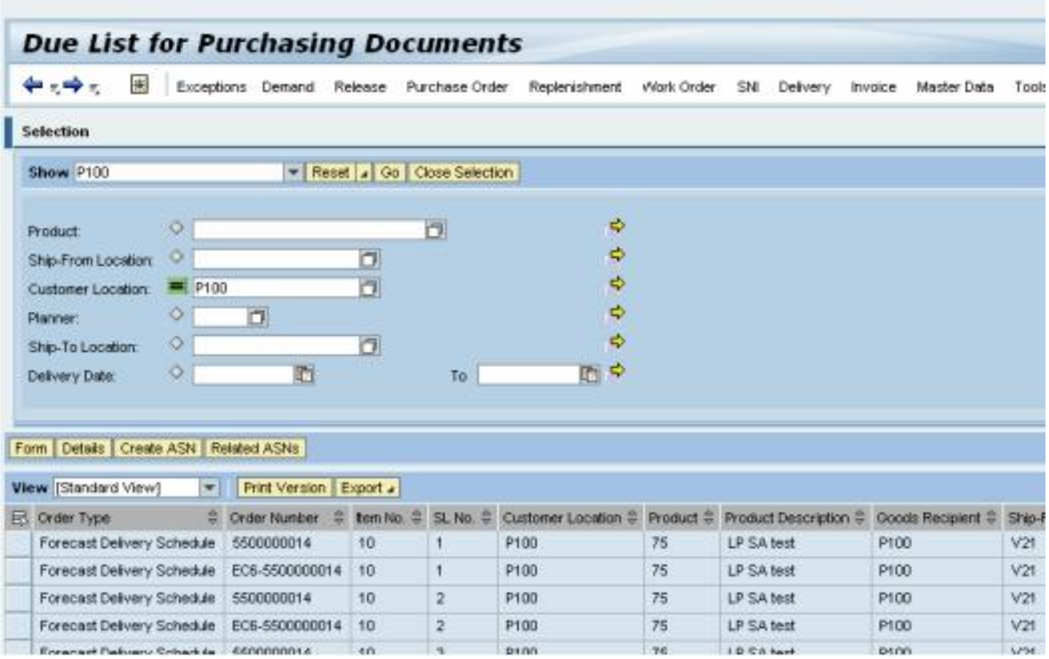
Introduction

Supply Network Collaboration is one of the core capabilities of Adaptive Supply Chain Networks, a fundamental principle for the networked and outsourced enterprise of today and tomorrow. Through SNC, customers and suppliers can simultaneously eliminate inefficiencies in their supply chains by synchronizing the flow of information between them. SNC offers a 360 degree view on supply chain collaboration, offering a company ways to effectively collaborate with its customers, suppliers, 3 rd party logistics providers and outsourced manufacturing partners.[1]

Business Requirement

One uses the Supply Network Collaboration (SAP SNC) Web user interface (Web UI) to perform tasks relevant to one’s role. The SNC Web UI contains different types of screens – overview screens, detail screens, details screens for time series data.

We faced a business requirement wherein we had to modify these screens by adding columns to the tables. For Example: In the ‘Due List for Purchasing documents’ screen, column ‘Commitment Code’ needed to be added.



Due List for Purchasing Documents

Selection

Show P100

Product:
Ship-From Location:
Customer Location: P100
Planner:
Ship-To Location:
Delivery Date: To

View [Standard View]

Order Type	Order Number	Item No.	SL No.	Customer Location	Product	Product Description	Goods Recipient	Ship-F
Forecast Delivery Schedule	5500000014	10	1	P100	75	LP SA test	P100	V21
Forecast Delivery Schedule	EC6-5500000014	10	1	P100	75	LP SA test	P100	V21
Forecast Delivery Schedule	5500000014	10	2	P100	75	LP SA test	P100	V21
Forecast Delivery Schedule	EC6-5500000014	10	2	P100	75	LP SA test	P100	V21
Forecast Delivery Schedule	5500000014	10	3	P100	75	LP SA test	P100	V21

Solution

SNC Report Basics

The SNC Web UI applications have been developed in Web Dynpro ABAP.

Most SNC applications have the following flow:

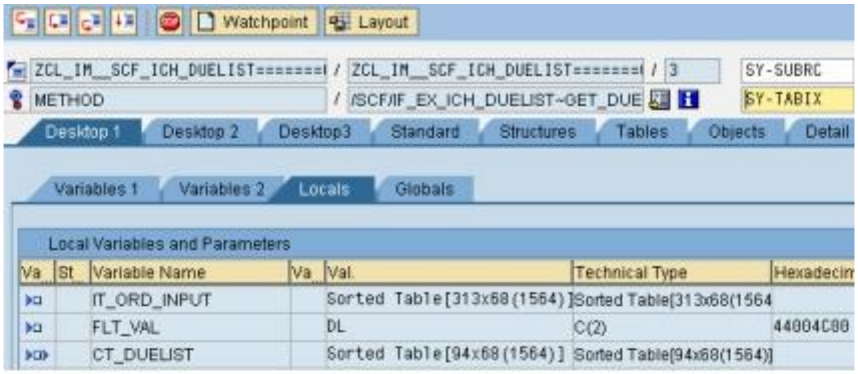
- Data is retrieved from database into an internal table.

For Example, for Due List Processing:

BADI: /SCF/ICH_DUELIST

Method: GET_DUELIST

The internal table CT_DUELIST contains the Duelist data.

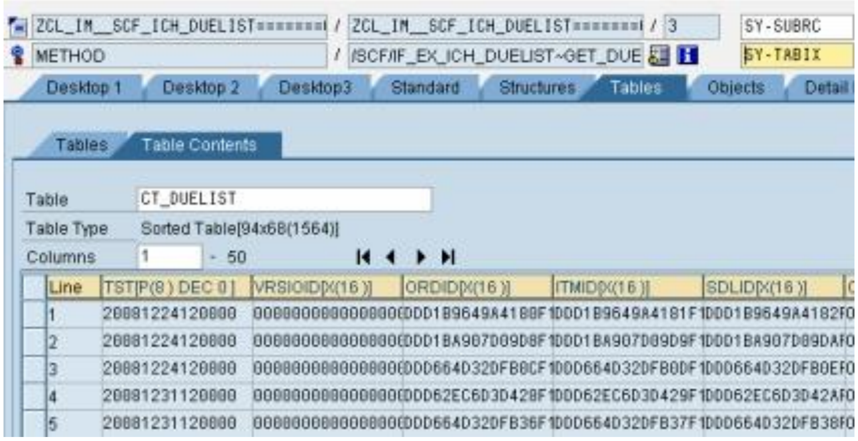


ZCL_IM__SCF_ICH_DUELIST======= / ZCL_IM__SCF_ICH_DUELIST======= / 3 SY-SUBRC

METHOD / /SCF/IF_EX_ICH_DUELIST~GET_DUE SY-TABIX

Local Variables and Parameters

Va	St	Variable Name	Va	Val.	Technical Type	Hexadecim
▸□		IT_ORD_INPUT		Sorted Table[313x68(1564)]	Sorted Table[313x68(1564	
▸□		FLT_VAL		DL	C(2)	44004C00
▸□▸		CT_DUELIST		Sorted Table[94x68(1564)]	Sorted Table[94x68(1564)]	



ZCL_IM__SCF_ICH_DUELIST======= / ZCL_IM__SCF_ICH_DUELIST======= / 3 SY-SUBRC

METHOD / /SCF/IF_EX_ICH_DUELIST~GET_DUE SY-TABIX

Table CT_DUELIST

Table Type Sorted Table[94x68(1564)]

Columns 1 - 50

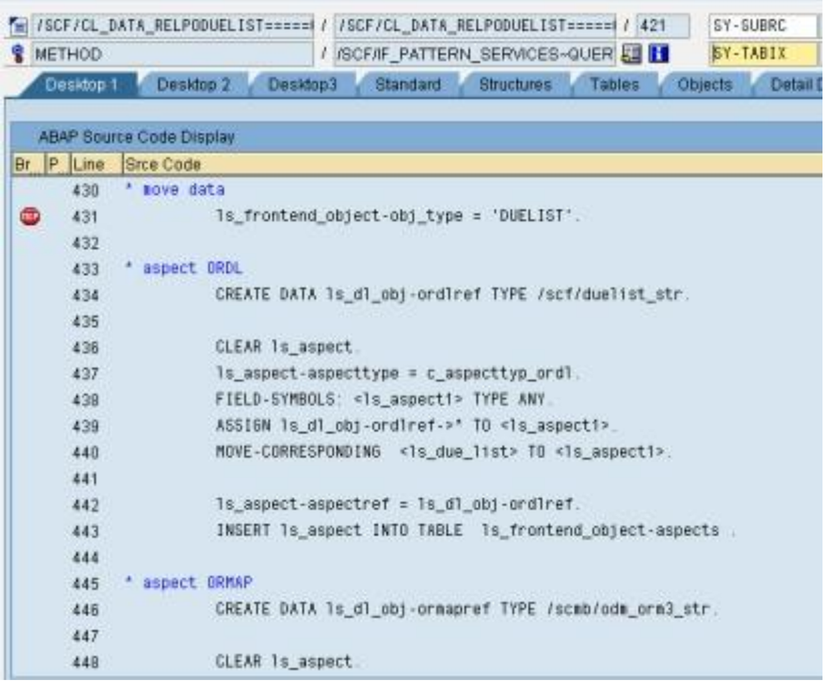
Line	TST[P(8) DEC 0]	VRSIOID[X(16)]	ORDID[X(16)]	ITMID[X(16)]	SDLID[X(16)]
1	20081224120000	000000000000000	DDD1B9649A4180F	1DDD1B9649A4181F	1DDD1B9649A4182F0
2	20081224120000	000000000000000	DDD1BA907D09D8F	1DDD1BA907D09D9F	1DDD1BA907D09DAF0
3	20081224120000	000000000000000	DDD664D32DFB0CF	1DDD664D32DFB0DF	1DDD664D32DFB0EF0
4	20081231120000	000000000000000	DDD62EC6D3D428F	1DDD62EC6D3D429F	1DDD62EC6D3D42AF0
5	20081231120000	000000000000000	DDD664D32DFB36F	1DDD664D32DFB37F	1DDD664D32DFB38F0

- This data is then mapped to a table column containing aspect structures. These aspect structure columns are mapped to the tables on the screen. One screen could be linked to multiple aspect structures

Class: /SCF/CL_DATA_RELPODUELIST

Method: /SCF/IF_PATTERN_SERVICES~QUERY

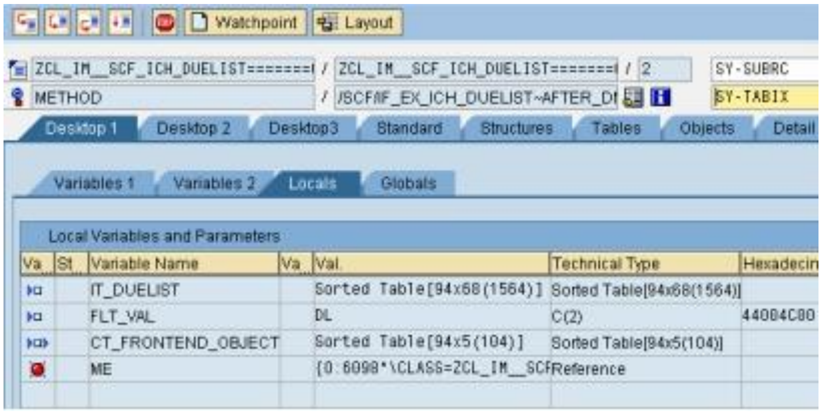
In the screenshot below, we can see that the duelist data is being mapped to several Aspects namely ORDL, ORMAP etc.



```
* move data
        ls_frontend_object-obj_type = 'DUELIST'.

* aspect ORDL
        CREATE DATA ls_dl_obj-ordlref TYPE /scf/duelist_str.

        CLEAR ls_aspect.
        ls_aspect-aspecttype = c_aspecttyp_ordl.
        FIELD-SYMBOLS: <ls_aspect1> TYPE ANY.
        ASSIGN ls_dl_obj-ordlref->* TO <ls_aspect1>.
        MOVE-CORRESPONDING  <ls_due_list> TO <ls_aspect1>.

        ls_aspect-aspectref = ls_dl_obj-ordlref.
        INSERT ls_aspect INTO TABLE  ls_frontend_object-aspects .

* aspect ORMAP
        CREATE DATA ls_dl_obj-ormapref TYPE /scmb/odm_orm3_str.

        CLEAR ls_aspect.
```

BADI: /SCF/ICH_DUELIST
Method: AFTER_DM2FROBJ

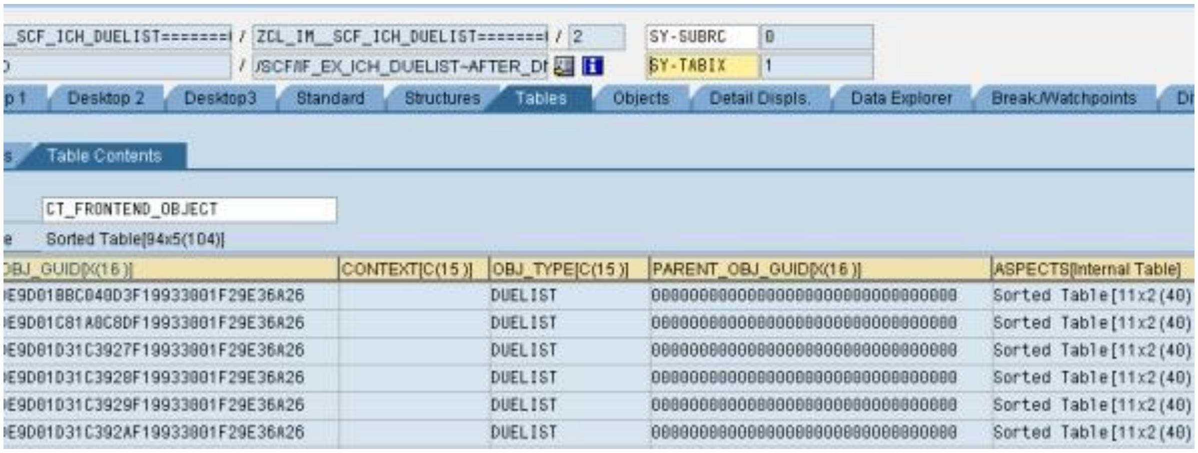
The table CT_FRONTEND_OBJECT contains each row of the Duelist data.



Local Variables and Parameters

Va	St	Variable Name	Va	Val.	Technical Type	Hexadecin
▸□		IT_DUELIST		Sorted Table[94x68(1564)]	Sorted Table[94x68(1564)]	
▸□		FLT_VAL		DL	C(2)	44004C00
▸□▸		CT_FRONTEND_OBJECT		Sorted Table[94x5(104)]	Sorted Table[94x5(104)]	
●		ME		{O:6098*\CLASS=ZCL_IM__SCF	Reference	

The column ‘ASPECTS’ contains the mapped data.



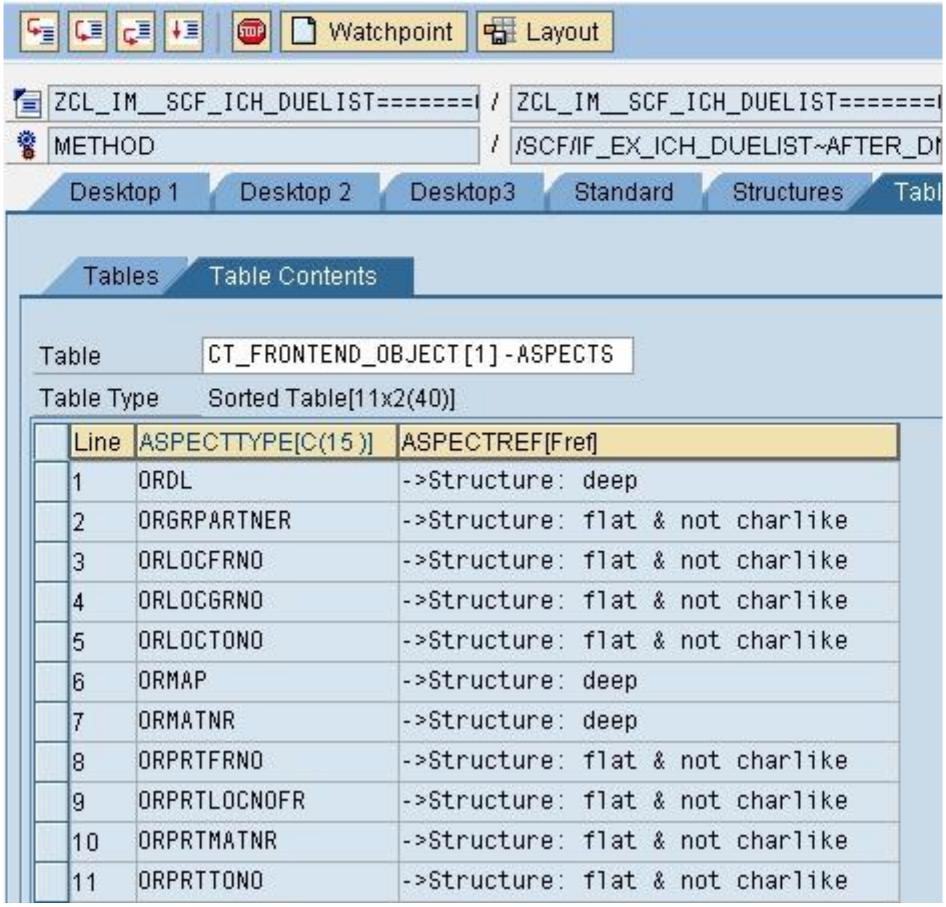
_SCF_ICH_DUELIST======= / ZCL_IM__SCF_ICH_DUELIST======= / 2
/ /SCF/IF_EX_ICH_DUELIST~AFTER_DI

SY-SUBRC 0
SY-TABIX 1

CT_FRONTEND_OBJECT
e Sorted Table[94x5(104)]

OBJ_GUID[X(16)]	CONTEXT[C(15)]	OBJ_TYPE[C(15)]	PARENT_OBJ_GUID[X(16)]	ASPECTS[Internal Table]
E9D01BBC040D3F19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)
E9D01C81A0C8DF19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)
E9D01D31C3927F19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)
E9D01D31C3928F19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)
E9D01D31C3929F19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)
E9D01D31C392AF19933001F29E36A26		DUELIST	00000000000000000000000000000000	Sorted Table[11x2(40)

We can see that data for each row is mapped to several Aspects



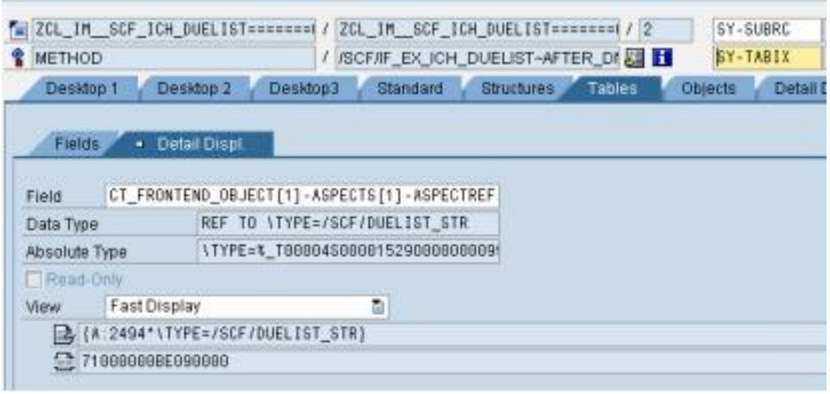
ZCL_IM__SCF_ICH_DUELIST======= / ZCL_IM__SCF_ICH_DUELIST=======
METHOD / /SCF/IF_EX_ICH_DUELIST~AFTER_DI

Table CT_FRONTEND_OBJECT[1]-ASPECTS
Table Type Sorted Table[11x2(40)]

Line	ASPECTTYPE[C(15)]	ASPECTREF[Fref]
1	ORDL	->Structure: deep
2	ORGRPARTNER	->Structure: flat & not charlike
3	ORLOCFRNO	->Structure: flat & not charlike
4	ORLOCGRNO	->Structure: flat & not charlike
5	ORLOCTONO	->Structure: flat & not charlike
6	ORMAP	->Structure: deep
7	ORMATNR	->Structure: deep
8	ORPRTFRNO	->Structure: flat & not charlike
9	ORPRTLOCNOFR	->Structure: flat & not charlike
10	ORPRTMATNR	->Structure: flat & not charlike
11	ORPRTTONO	->Structure: flat & not charlike

Aspect ORDL corresponds to the structure /SCF/DUELIST_STR.

Similarly other aspects correspond to different structures.



ZCL_IM__SCF_ICH_DUELIST=======(/ ZCL_IM__SCF_ICH_DUELIST=======(/ 2 SY-SUBRC

METHOD / /SCF/IF_EX_ICH_DUELIST~AFTER_DI SY-TABIX

Field CT_FRONTEND_OBJECT[1]-ASPECTS[1]-ASPECTREF

Data Type REF TO \TYPE=/SCF/DUELIST_STR

Absolute Type \TYPE=%_T00004S00001529000000009

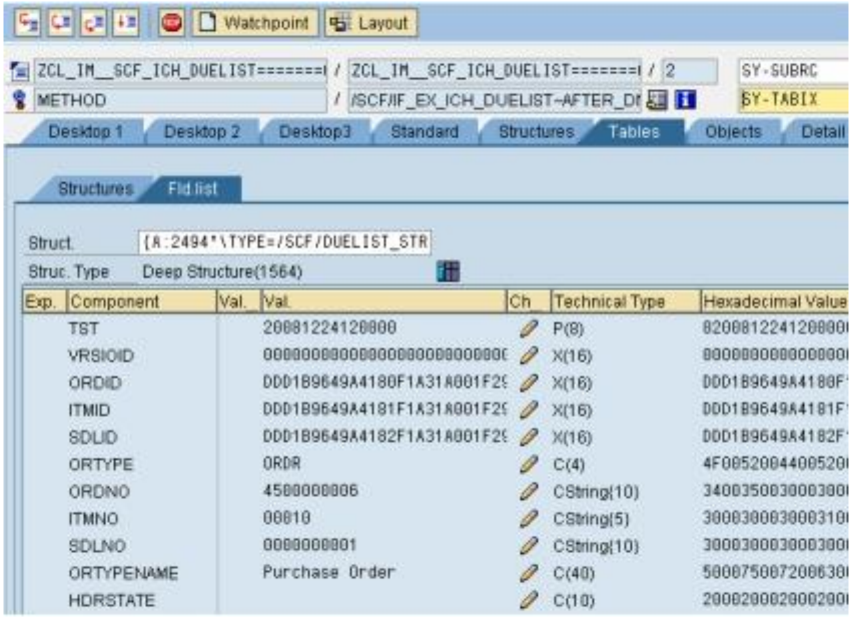
□ Read-Only

View Fast Display

{A:2494*\TYPE=/SCF/DUELIST_STR}

71000000BE090000

We can see the data contained in the aspect structure. This data is mapped to the screen columns.



ZCL_IM__SCF_ICH_DUELIST=======(/ ZCL_IM__SCF_ICH_DUELIST=======(/ 2 SY-SUBRC

METHOD / /SCF/IF_EX_ICH_DUELIST~AFTER_DI SY-TABIX

Struct. {A:2494*\TYPE=/SCF/DUELIST_STR

Struc. Type Deep Structure(1564)

Exp.	Component	Val.	Val.	Ch	Technical Type	Hexadecimal Value
	TST		20081224120000	✎	P(8)	020081224120000
	VRSIOID		0000000000000000000000000000	✎	X(16)	0000000000000000
	ORDID		DDD1B9649A4180F1A31A001F29	✎	X(16)	DDD1B9649A4180F
	ITMID		DDD1B9649A4181F1A31A001F29	✎	X(16)	DDD1B9649A4181F
	SDLID		DDD1B9649A4182F1A31A001F29	✎	X(16)	DDD1B9649A4182F
	ORTYPE		ORDR	✎	C(4)	4F00520044005200
	ORDNO		4500000006	✎	CString(10)	3400350030003000
	ITMNO		00010	✎	CString(5)	3000300030003100
	SDLNO		0000000001	✎	CString(10)	3000300030003000
	ORTYPENAME		Purchase Order	✎	C(40)	5000750072006300
	HDRSTATE			✎	C(10)	2000200020002000

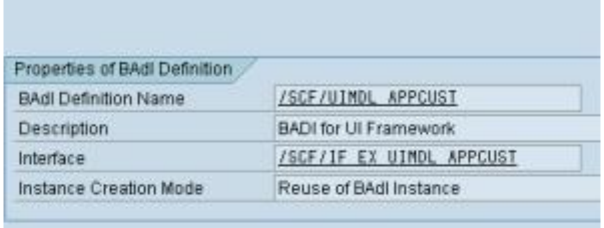
Development Details

Step I: Add the column on the screen.

1. Now that we have an idea of Aspects, we'll see how we can map the aspect structure field to the column on the screen. For this we'll have to first add the column on the screen. This can be done by implementing the following BADI.

BADI: /SCF/UIMDL_APPCUST

Method: /SCF/IF_EX_UIMDL_APPCUST~GET_TABLEVIEW_DATA

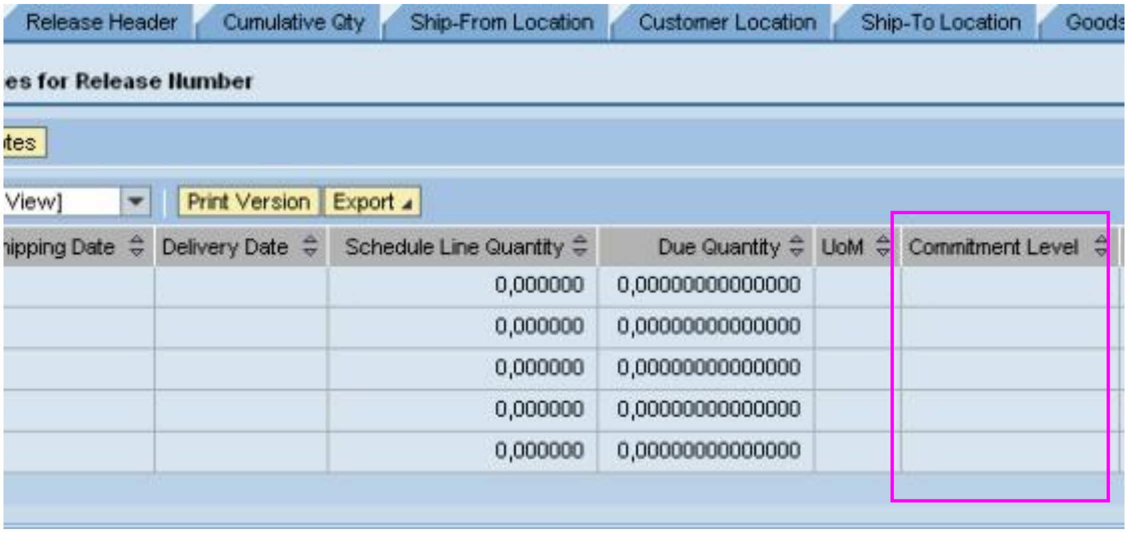


Properties of BAdI Definition

BAdI Definition Name	/SCF/UIMDL_APPCUST
Description	BADI for UI Framework
Interface	/SCF/IF_EX_UIMDL_APPCUST
Instance Creation Mode	Reuse of BAdI Instance

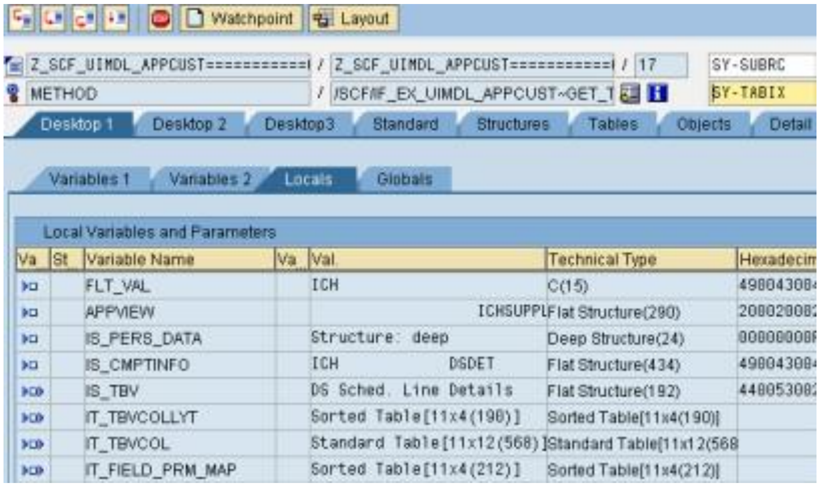
2. Check if the column exists on another screen. If so identify the Column ID, Data Element and Param ID.

   For example, the Release Details screen contains a column called 'Commitment Level', that we need to add to the Due List for Purchasing Document Screen.



hipping Date	Delivery Date	Schedule Line Quantity	Due Quantity	UoM	Commitment Level
		0,000000	0,00000000000000		
		0,000000	0,00000000000000		
		0,000000	0,00000000000000		
		0,000000	0,00000000000000		
		0,000000	0,00000000000000		

Find out the Column ID from the BADI /SCF/UIMDL_APPCUST,
Method: /SCF/IF_EX_UIMDL_APPCUST~GET_TABLEVIEW_DATA

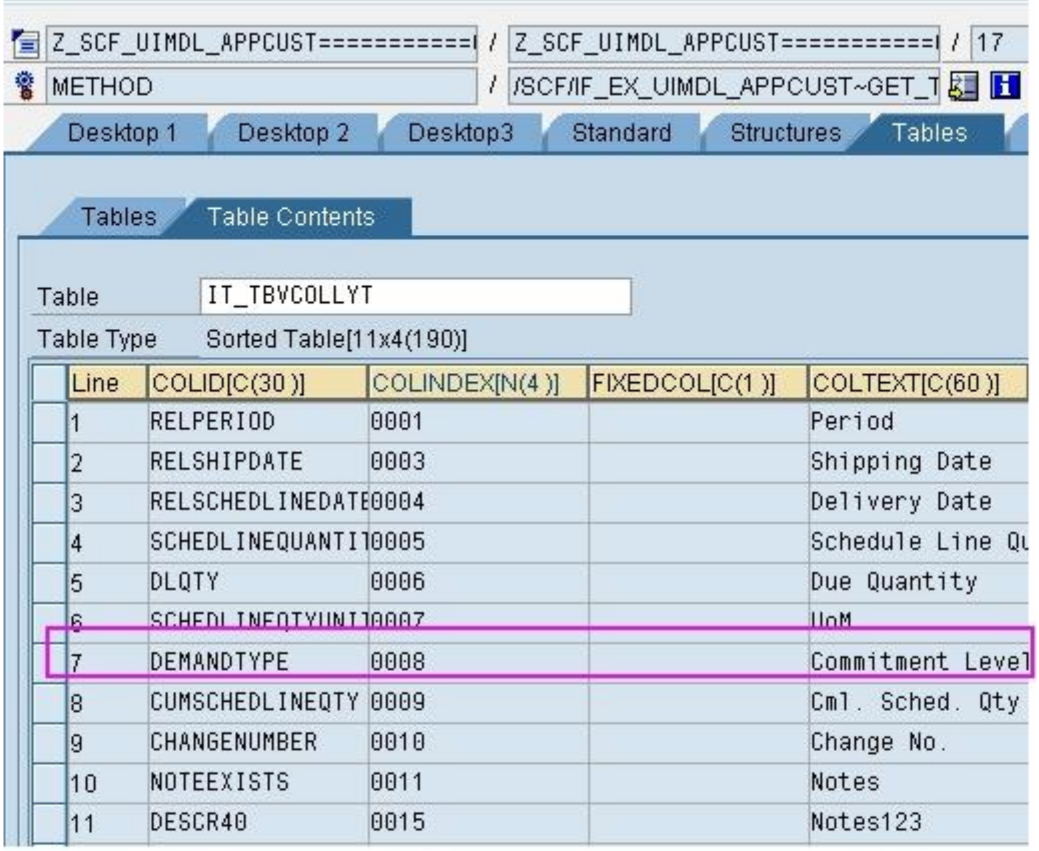


Z_SCF_UIMDL_APPCUST===========| / Z_SCF_UIMDL_APPCUST===========| / 17 SY-SUBRC
METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_T SY-TABIX

Local Variables and Parameters

Va	St	Variable Name	Va	Val.	Technical Type	Hexadecim
▸□		FLT_VAL		ICH	C(15)	49804308
▸□		APPVIEW		ICHSUPPL	Flat Structure(280)	20802080
▸□		IS_PERS_DATA		Structure: deep	Deep Structure(24)	00000000
▸□		IS_CMPTINFO		ICH DSDET	Flat Structure(434)	49804308
▸□		IS_TBV		DS Sched. Line Details	Flat Structure(192)	44805308
▸□		IT_TBVCOLLYT		Sorted Table[11x4(190)]	Sorted Table[11x4(190)]	
▸□		IT_TBVCOL		Standard Table[11x12(568)]	Standard Table[11x12(568	
▸□		IT_FIELD_PRM_MAP		Sorted Table[11x4(212)]	Sorted Table[11x4(212)]	

Table IT_TBVCOLLYT will provide the Column ID. You can identify it from the column text.



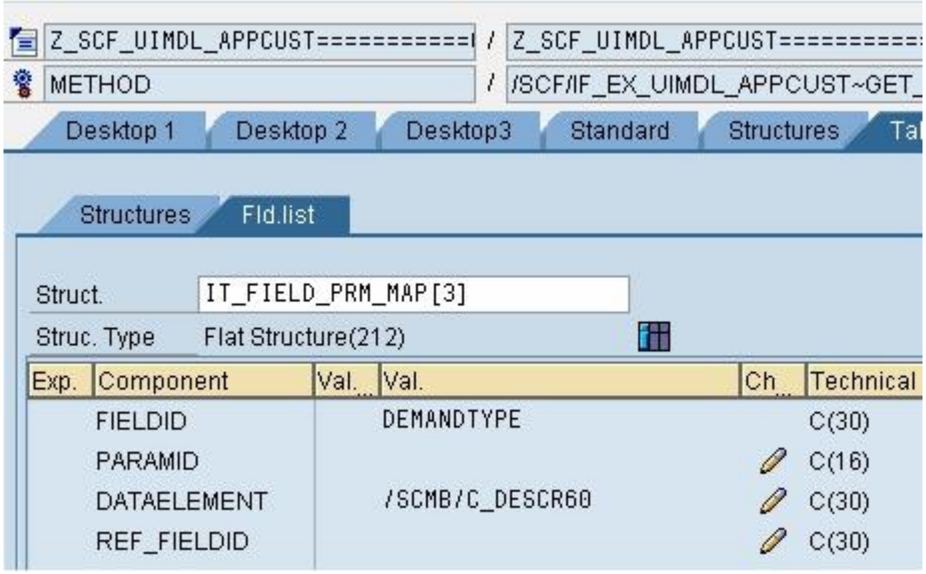
Z_SCF_UIMDL_APPCUST===========| / Z_SCF_UIMDL_APPCUST===========| / 17
METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_T

Table IT_TBVCOLLYT

Table Type Sorted Table[11x4(190)]

Line	COLID[C(30)]	COLINDEX[N(4)]	FIXEDCOL[C(1)]	COLTEXT[C(60)]
1	RELPERIOD	0001		Period
2	RELSHIPDATE	0003		Shipping Date
3	RELSCHEDLINEDATE	0004		Delivery Date
4	SCHEDLINEQUANTI1	0005		Schedule Line Qu
5	DLQTY	0006		Due Quantity
6	SCHEDLINEQTYUNIT	0007		UoM
7	DEMANDTYPE	0008		Commitment Level
8	CUMSCHEDLINEQTY	0009		Cml. Sched. Qty
9	CHANGENUMBER	0010		Change No.
10	NOTEEXISTS	0011		Notes
11	DESCR40	0015		Notes123

Table IT_FIELD_PRM_MAP will provide the Data Element and Param ID.



Z_SCF_UIMDL_APPCUST===========| / Z_SCF_UIMDL_APPCUST==========

METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_

Struct. IT_FIELD_PRM_MAP[3]

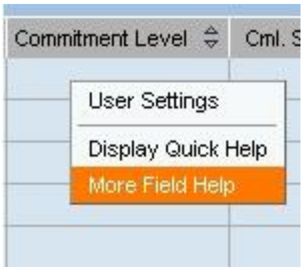
Struc. Type Flat Structure(212)

Exp.	Component	Val...	Val.	Ch...	Technical
	FIELDID		DEMANDTYPE		C(30)
	PARAMID			✎	C(16)
	DATAELEMENT		/SCMB/C_DESCR60	✎	C(30)
	REF_FIELDID			✎	C(30)

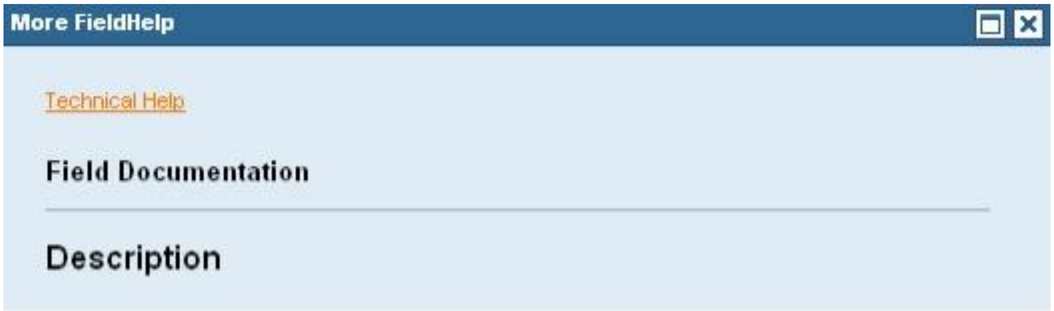
3. If the column to be added does not exist on any other screen, then choose the Data element (Data type) of a similar column/ create a Data element.

Note: If we use a similar column instead of the exact column, we will have to maintain the language translations for the same in different languages. For example if we use ‘Schedule Line Quantity’ column to display a custom column ‘Last ASN Quantity’, we would need to maintain the text ‘Last ASN Quantity’ in different languages .If this is not done, then if the user views the application in German for example, he would see the German translation of ‘Schedule Line Quantity’. Same logic applies for custom Data Element.

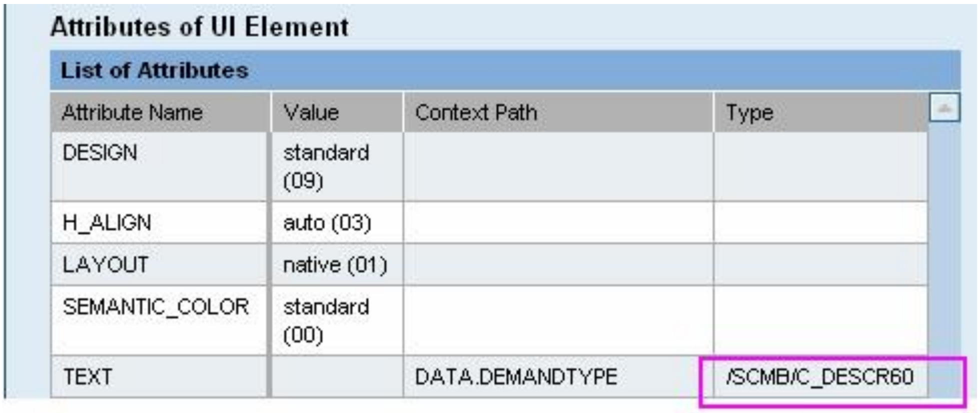
Right click on the column and select ‘More field Help’.



Click ‘Technical Help’.



The Data Element will be available in the Attributes of UI element.

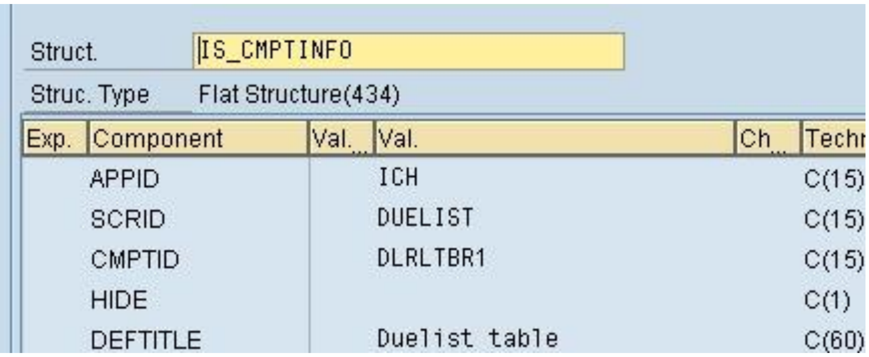


Attributes of UI Element

List of Attributes

Attribute Name	Value	Context Path	Type
DESIGN	standard (09)		
H_ALIGN	auto (03)		
LAYOUT	native (01)		
SEMANTIC_COLOR	standard (00)		
TEXT		DATA.DEMANDTYPE	/SCMB/C_DESCR60

4. Find the application ID, Screen ID and Component ID of the screen that we want to add the column to. This can be obtained from the parameter IS_CMPTINFO in the method GET_TABLEVIEW_DATA. The Default Title field will provide a clue for the correct Screen and Component ID, but to ensure that these parameters are the correct ones, check the table IT_TBVCOLLYT for the columns being displayed on screen.



Struct. IS_CMPTINFO

Struc. Type Flat Structure(434)

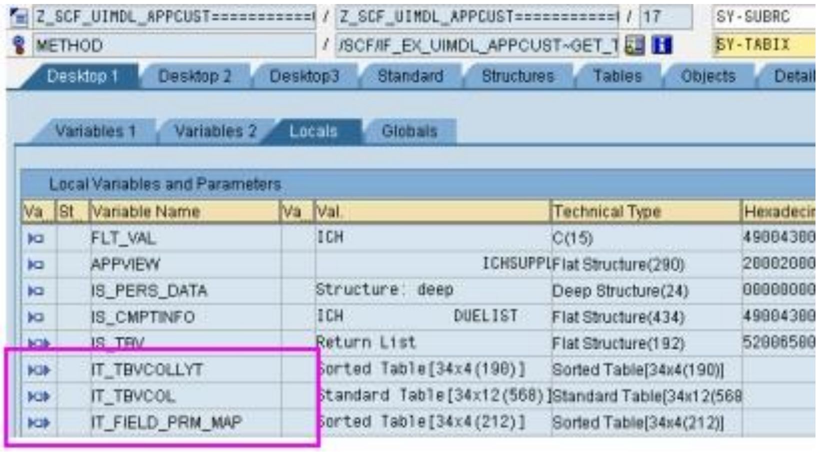
Exp.	Component	Val....	Val.	Ch...	Techr
	APPID		ICH		C(15)
	SCRID		DUELIST		C(15)
	CMPTID		DLRLTBR1		C(15)
	HIDE				C(1)
	DEFTITLE		Duelist table		C(60)

5. Now that we have the screen and column details we can write the code In the BADI,

BADI: /SCF/UIMDL_APPCUST

Method: /SCF/IF_EX_UIMDL_APPCUST~GET_TABLEVIEW_DATA

Add values to the three tables: IT_TBVCOLLYT, IT_TBVCOL, IT_FIELD_PRM_MAP



Z_SCF_UIMDL_APPCUST============ / Z_SCF_UIMDL_APPCUST============ / 17 SY-SUBRC

METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_T SY-TABIX

Local Variables and Parameters

Va...	St...	Variable Name	Va...	Val.	Technical Type	Hexadecir
		FLT_VAL		ICH	C(15)	49004300
		APPVIEW		ICHSUPPL	Flat Structure(290)	20002000
		IS_PERS_DATA		Structure: deep	Deep Structure(24)	00000000
		IS_CMPTINFO		ICH DUELIST	Flat Structure(434)	49004300
		IS_TBV		Return List	Flat Structure(192)	52006500
		IT_TBVCOLLYT		Sorted Table[34x4(190)]	Sorted Table[34x4(190)]	
		IT_TBVCOL		Standard Table[34x12(568)]	Standard Table[34x12(568	
		IT_FIELD_PRM_MAP		Sorted Table[34x4(212)]	Sorted Table[34x4(212)]	

6. Make the following entries for the particular Application, Screen, and Component only. (These were identified in step 5 above)

i.e.

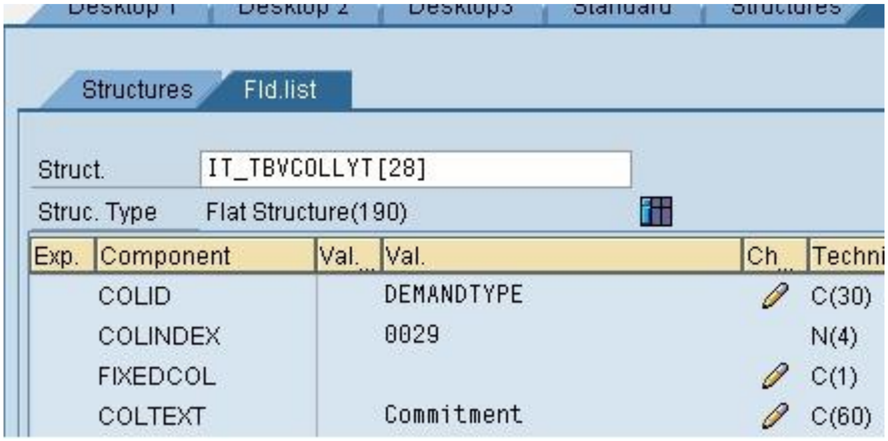
```
 IF is_cmptinfo-appid EQ 'ICH'
 AND is_cmptinfo-scrid EQ 'DUELIST'
AND is_cmptinfo-cmptid EQ 'DLRLTBR1'.
----------------
----------------
ENDIF
```

- **IT_TBVCOLLYT: Table for Tableview column layout**

Fill the Column ID (obtained in step 2), position and text.

If the Column ID was not obtained, then we can provide our own Column ID.

Note: If we provide our own column names, they will not be automatically translated into other languages. We will need to maintain text, tool tip for the same in different languages.



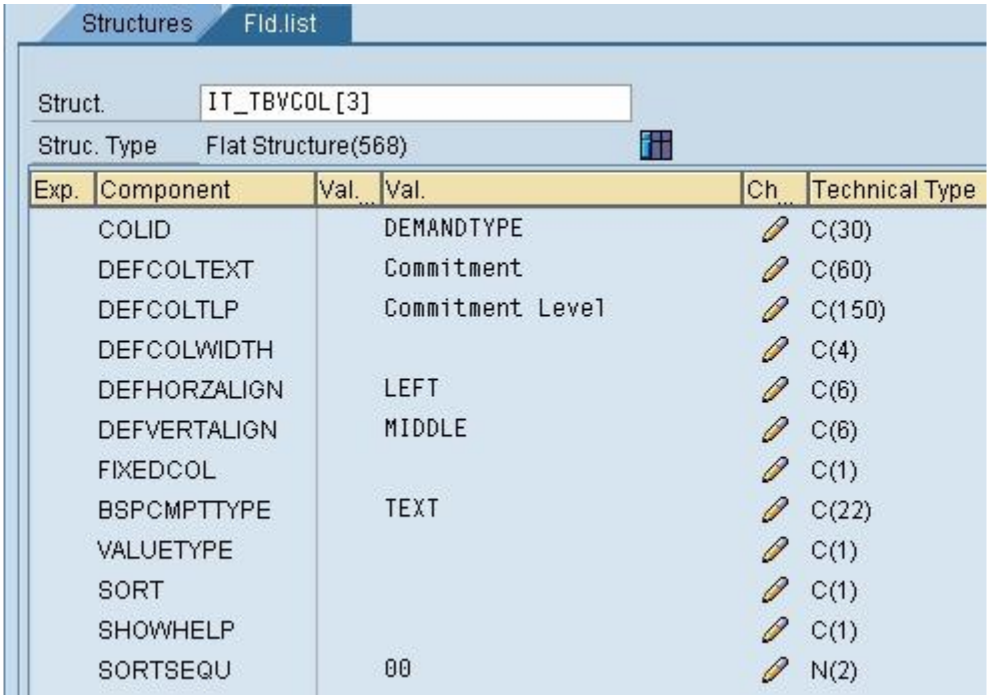
Struct. IT_TBVCOLLYT[28]

Struc. Type Flat Structure(190)

Exp.	Component	Val...	Val.	Ch...	Techni
	COLID		DEMANDTYPE	✎	C(30)
	COLINDEX		0029		N(4)
	FIXEDCOL			✎	C(1)
	COLTEXT		Commitment	✎	C(60)

IT_TBVCOL: Table for Tableview column configuration

Provide the default Column text, tool tip, alignment, column type, sort sequence etc.



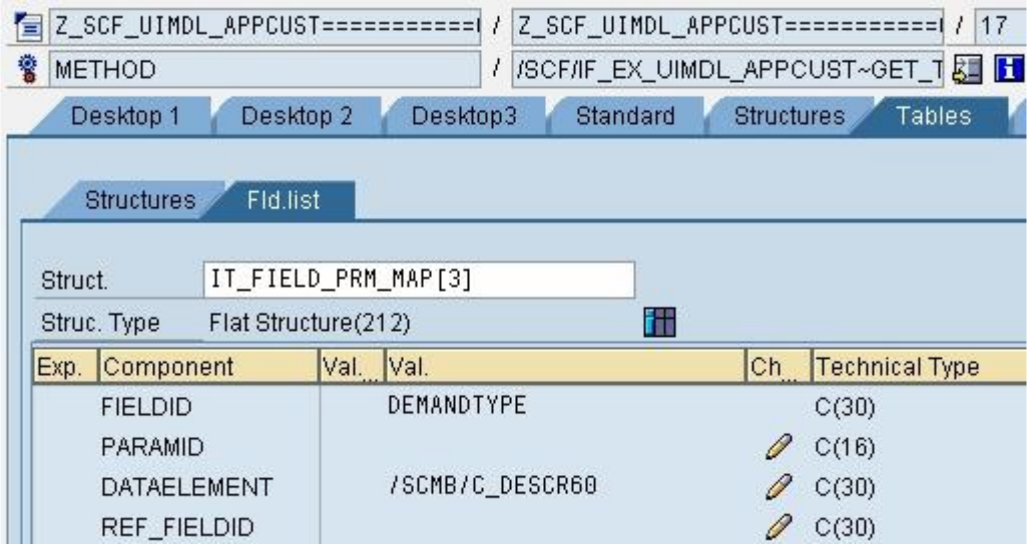
Struct. IT_TBVCOL[3]

Struc. Type Flat Structure(568)

Exp.	Component	Val....	Val.	Ch...	Technical Type
	COLID		DEMANDTYPE	✎	C(30)
	DEFCOLTEXT		Commitment	✎	C(60)
	DEFCOLTLP		Commitment Level	✎	C(150)
	DEFCOLWIDTH			✎	C(4)
	DEFHORZALIGN		LEFT	✎	C(6)
	DEFVERTALIGN		MIDDLE	✎	C(6)
	FIXEDCOL			✎	C(1)
	BSPCMPTTYPE		TEXT	✎	C(22)
	VALUETYPE			✎	C(1)
	SORT			✎	C(1)
	SHOWHELP			✎	C(1)
	SORTSEQU		00	✎	N(2)

- **IT_FIELD_PRM_MAP: Field parameter mapping**

Provide the data element, Param ID for the column.



Z_SCF_UIMDL_APPCUST===========(/ Z_SCF_UIMDL_APPCUST===========(/ 17

METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_T

Struct. IT_FIELD_PRM_MAP[3]

Struc. Type Flat Structure(212)

Exp.	Component	Val....	Val.	Ch...	Technical Type
	FIELDID		DEMANDTYPE		C(30)
	PARAMID			✎	C(16)
	DATAELEMENT		/SCMB/C_DESCR60	✎	C(30)
	REF_FIELDID			✎	C(30)

Note: Providing the data element is mandatory. Without this step the column will not be visible on screen.

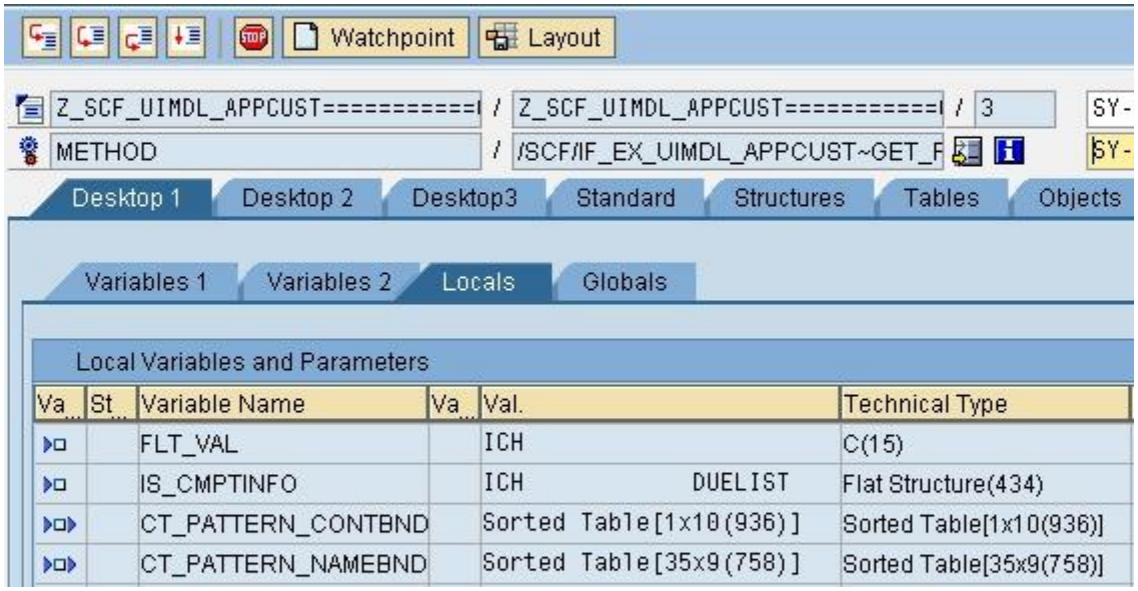
Now we have added the column on the screen. But we still need to bind data to it.

Step II: Binding the Aspect data with Column.

In the same BADI, implement another method.

BADI: /SCF/UIMDL_APPCUST

Method: /SCF/IF_EX_UIMDL_APPCUST~GET_PATTERN_BINDING



Z_SCF_UIMDL_APPCUST===========(/ Z_SCF_UIMDL_APPCUST===========(/ 3

METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_F

Local Variables and Parameters

Va...	St...	Variable Name	Va...	Val.	Technical Type
		FLT_VAL		ICH	C(15)
		IS_CMPTINFO		ICH DUELIST	Flat Structure(434)
		CT_PATTERN_CONTBND		Sorted Table[1x10(936)]	Sorted Table[1x10(936)]
		CT_PATTERN_NAMEBND		Sorted Table[35x9(758)]	Sorted Table[35x9(758)]

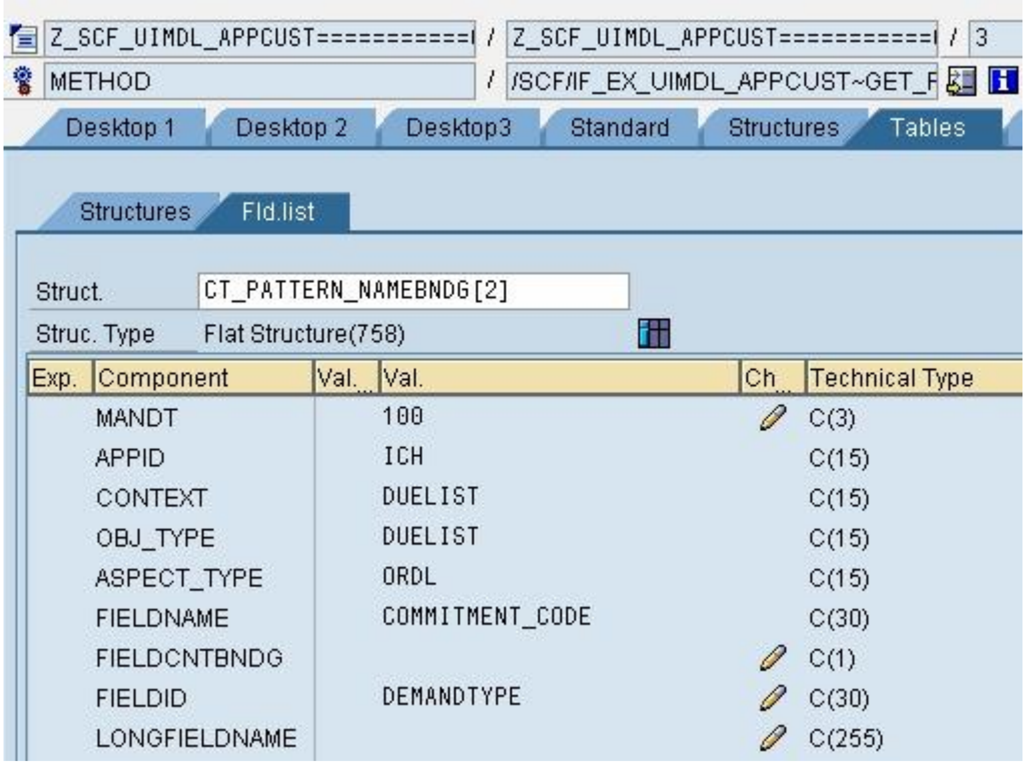
The table CT_PATTERN_NAMEBNDG contains the binding between aspect and screen columns.

Make the following entries for the particular Application, Screen, and Component only. (These were identified in step 5 above)

i.e.

```
 IF is_cmptinfo-appid EQ 'ICH'
 AND is_cmptinfo-scrid EQ 'DUELIST'
AND is_cmptinfo-cmptid EQ 'DLRLTBR1'.
----------------
----------------
ENDIF
```

Provide the following:



Z_SCF_UIMDL_APPCUST===========| / Z_SCF_UIMDL_APPCUST===========| / 3

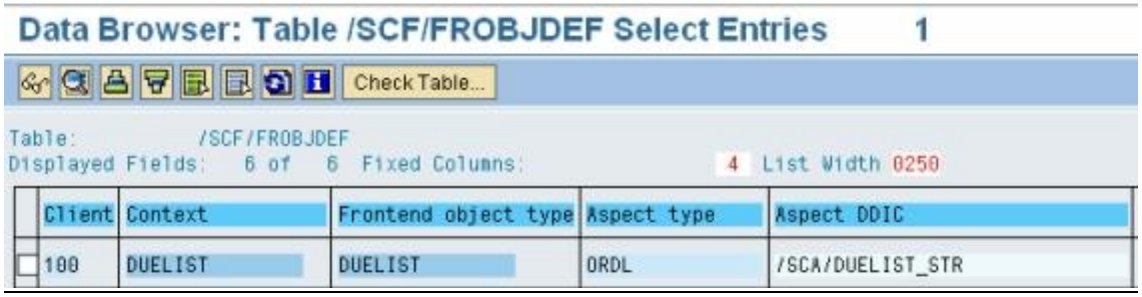
METHOD / /SCF/IF_EX_UIMDL_APPCUST~GET_F

Struct. CT_PATTERN_NAMEBNDG[2]

Struc. Type Flat Structure(758)

Exp.	Component	Val...	Val.	Ch...	Technical Type
	MANDT		100	✎	C(3)
	APPID		ICH		C(15)
	CONTEXT		DUELIST		C(15)
	OBJ_TYPE		DUELIST		C(15)
	ASPECT_TYPE		ORDL		C(15)
	FIELDNAME		COMMITMENT_CODE		C(30)
	FIELDCNTBNDG			✎	C(1)
	FIELDID		DEMANDTYPE	✎	C(30)
	LONGFIELDNAME			✎	C(255)

Application ID: can be obtained in the BADI in debugging mode. Varies with type of SNC application.
ICH = Inventory Collaboration Hub
SPP = Service Parts Planning
CDP = Demand Planning and so on
Aspect type: Can be obtained in the BADI in debugging mode.
Context, Object Type: Can be obtained from the table /SCF/FROBJDEF

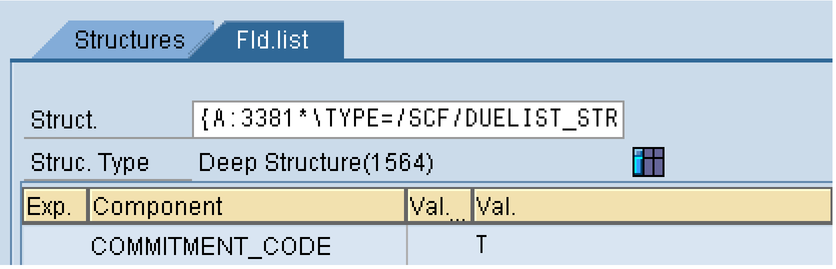


Data Browser: Table /SCF/FROBJDEF Select Entries 1

Table: /SCF/FROBJDEF
Displayed Fields: 6 of 6 Fixed Columns: 4 List Width 0250

	Client	Context	Frontend object type	Aspect type	Aspect DDIC
□	100	DUELIST	DUELIST	ORDL	/SCA/DUELIST_STR

Field name: Name of the field in the aspect structure that will contain the data



Struct. {A:3381*\TYPE=/SCF/DUELIST_STR

Struc. Type Deep Structure(1564)

Exp.	Component	Val...	Val.
	COMMITMENT_CODE		T

FieldID: Column ID that you have added
Now the data has been bound to the screen column.

Step III: Modify the data as required.

This can be done using BADIs/ Enhancement spots. Data can be modified after mapping to aspect structure or before, according to requirement.

For example for the Due List for Purchasing Documents, we can use

BADI: /SCF/ICH_DUELIST

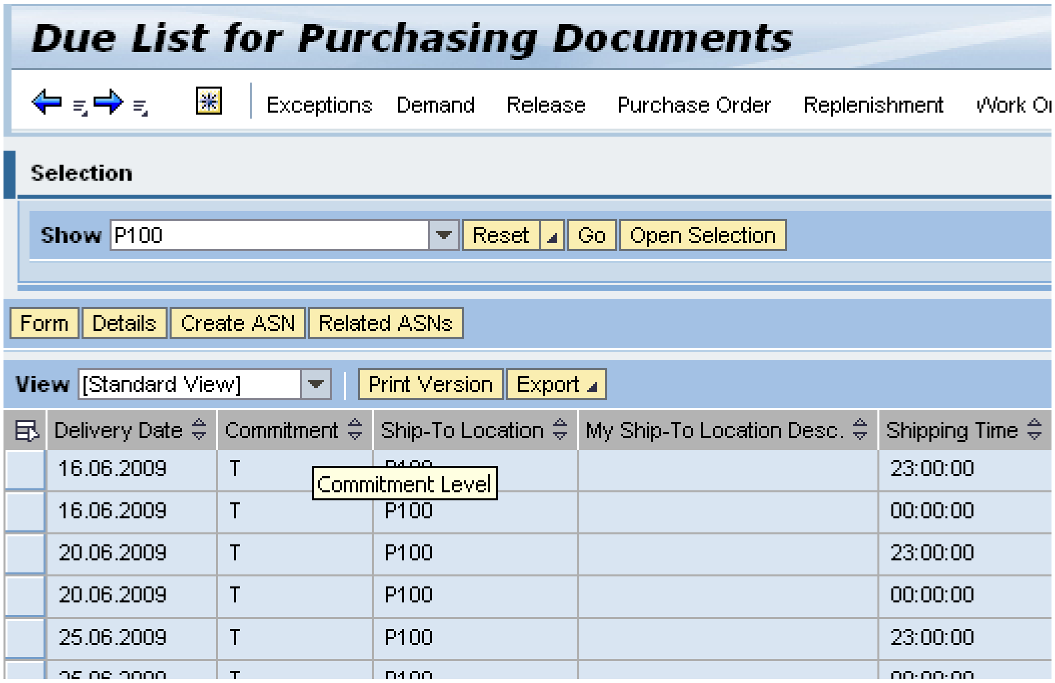
Method: GET_DUELIST (Before mapping to aspect)

AFTER_DM2FROBJ (After mapping to aspect)

Note: 1. In case the column(s) we wish to add do not exist in the aspect structure we can create an append structure for the aspect structure and add our column(s). We can then fill the data in the aspect using BADI/ Enhancements. The rest of the procedure, i.e. adding the column to screen and binding the column to aspect structure remains the same.
2. Not all SNC applications have Column binding to Aspect structures. The output table data is mapped to other tables containing "column ID- Data" mapping. In such cases, we'll need to create enhancements in the respective class – method to add out column ID and Data. For example: ASN Overview screen.

Result

We can see that the Column 'Commitment Level' has been added on the screen with header as 'Commitment' and tool tip as 'Commitment Level'. Note that we've even modified the value being displayed in the column to a code (T = Trade Off Zone, F = Firm Zone, P = Planning Zone) rather than the actual Commitment level description.



Delivery Date	Commitment	Ship-To Location	My Ship-To Location Desc.	Shipping Time
16.06.2009	T	P100		23:00:00
16.06.2009	T	P100		00:00:00
20.06.2009	T	P100		23:00:00
20.06.2009	T	P100		00:00:00
25.06.2009	T	P100		23:00:00
25.06.2009	T	P100		00:00:00

Summary

The procedure for adding a column to a table in SNC is three-fold.

a. Add the column to the screen.

BADI: /SCF/UIMDL_APPCUST (BADI for UI Framework)

Method: /SCF/IF_EX_UIMDL_APPCUST~GET_TABLEVIEW_DATA

b. Bind the column to the aspect structure. As mentioned earlier, the aspect structures contain the data that is to be displayed on the screen. There could be certain screens where this step would be required eg. PO Overview, Due List for purchasing documents. However on certain screens such as ASN Overview aspect structures are not used to store the output data.

c. Modify the internal table data that will be bound to the aspect structure or modify data after mapping to aspect structure.

References

Transactions:

/n/SCF/TBVCFG: Tableview Configuration

/n/SCF/BINDANDASS: Assign Aspects to Components

Related Content

SAP Note 1337787 - Product description not available in Due List screen

SAP Supply Network Collaboration (SNC) - SDN Wiki

SNC Documentation SCM 2007

Consuming Enterprise Services of SNC using XI Content

SAP SNC - Article on Basic scenarios

xsupply

For more information, visit the Supply Chain Management homepage

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

[1] SAP Supply Chain Management - Supply Network Collaboration, SAP AG

http://www50.sap.com/businessmaps/32263564DAF04BA59CFBF2C6307F5EFC.htm