

Automating adding Value Helps in Web Dynpro for JAVA



Applies to:

SAP NetWeaver 7.0 and above.

Summary

Easily implement value helps linked to ECC using helper class and Value Help Builder application.

Author: Alexei Isaev

Company: Pacific Coast Companies Inc.

Created on: 19 February 2008

Author Bio



Alexei Isaev has been developing solutions in SAP environment for 12 years. He is now working as Enterprise Application Architect for Pacific Coast Companies in Sacramento, CA.

Table of Contents

Introduction	3
Implementing ABAP part	4
Implementing Web Dynpro Java part	7
Using Value Help Builder application	8
Initial view.....	8
Selection of simple value help from complex value help	9
Code generator view	10
How to use generated code. Example.....	13
Appendix A. Code for function module Z_BAPI_SEARCHHELP_METADATA	16
Copyright.....	31

Introduction

When I first encountered the requirement to add value help to Web Dynpro application, I did not realize it can be so time-consuming. When you have to do it over and over again, it becomes a definite pain. First, you have to consider whether you're going to implement simple value selector (SVS) as dropdown, or slightly more complex Extended Value Selector (EVS), that allows to do some minimal filtering of values, or full-fledged Object Value Selector, that gives you almost all the same options you have in SAP R/3 systems.

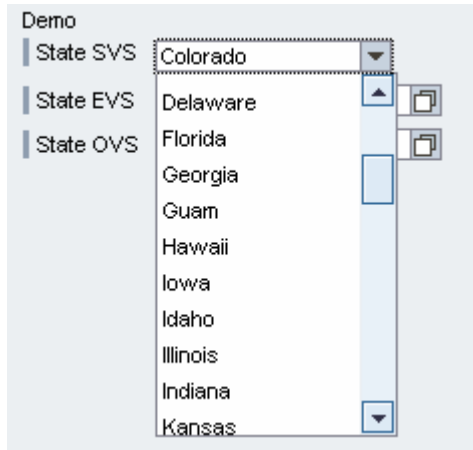


Fig.1. Simple Value Selector (SVS)

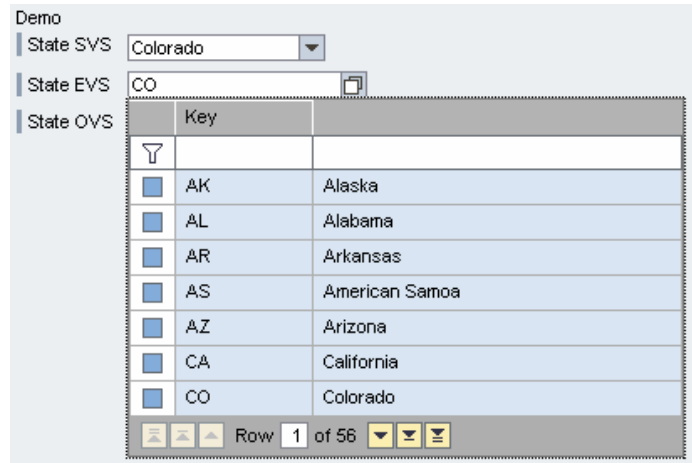


Fig.2. Extended Value Selector (EVS)

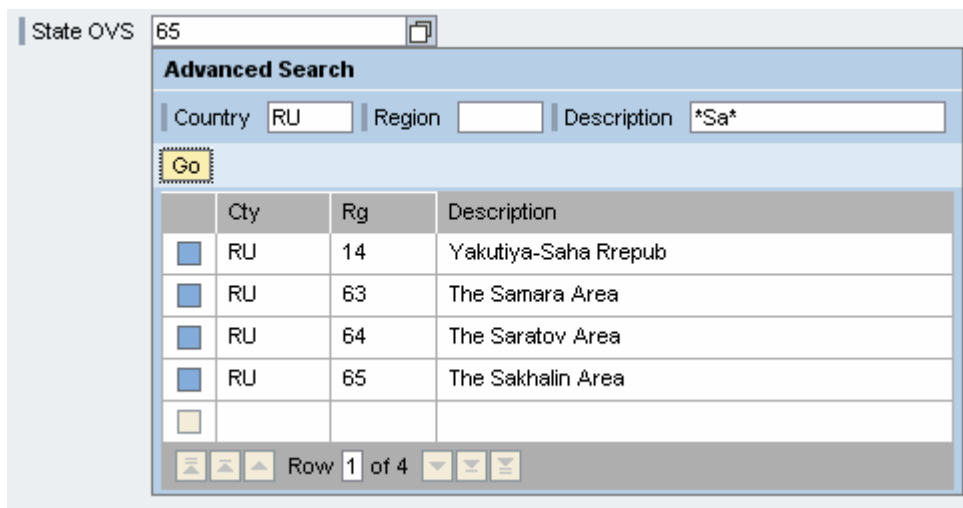


Fig.3. Object Value Selector (OVS)

Then comes the task of determining where to get the values. Everybody who have dealt with BAPI_HELPVALUES_GET know that you have to find appropriate Business Object and one of its methods, and then the parameter and field of that method in order to call this function module. This task, though not too difficult for anyone who worked on R/3 side, is not very obvious for everyone else. Additionally, users require fields to appear in sequence that is familiar to them, as well as making sure certain values are defaulted from other application values (e.g., when looking up materials for sales order, sales organization and distribution channel should be defaulted from sales order header information).

I wrote application to simplify task of defining value helps in Web Dynpro for Java. This application consists of 2 parts:

- Part that retrieves the data from R/3 system , formats them and returns them to appropriate places, and

- Value Help Builder, allowing programmer to create code snippets necessary to perform the first part.

Prerequisite for this application to work is presence in your R/3 system of the RFC-enabled function module that I wrote, Z_BAPI_SEARCHHELP_METADATA and buffering table ZSHELPBOR. Code for the function module and relevant includes is attached.

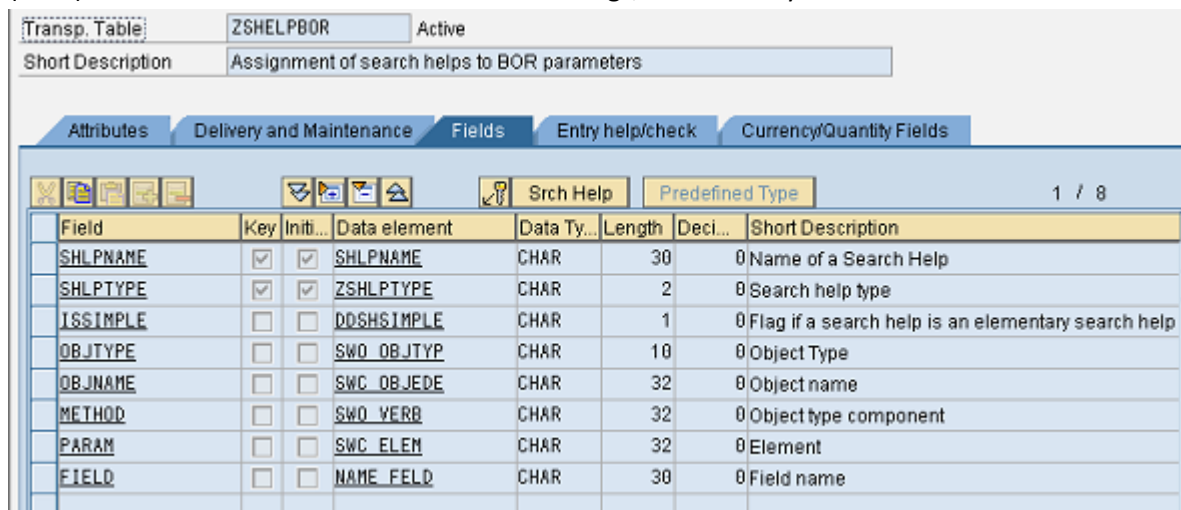
Implementing ABAP part

- (SE11) Define domain ZSHLPTYPE (Search help type), type CHAR(2)

Single Vals	
Fix.Val.	Short Descript.
FW	Fixed values
SH	Search help
CH	Value table

Fig. 4. Domain fixed values for ZSHLPTYPE

- (SE11) Define data element ZSHLPTYPE based on domain ZSHLPTYPE.
- (SE11) Create table ZSHELPBOR. In technical settings, I have it fully buffered.



Field	Key	Initi...	Data element	Data Ty...	Length	Deci...	Short Description
SHLPNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SHLPNAME	CHAR	30		0 Name of a Search Help
SHLPTYPE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZSHLPTYPE	CHAR	2		0 Search help type
ISSIMPLE	<input type="checkbox"/>	<input type="checkbox"/>	DDSHSIMPLE	CHAR	1		0 Flag if a search help is an elementary search help
OBJTYPE	<input type="checkbox"/>	<input type="checkbox"/>	SWO OBJTYP	CHAR	10		0 Object Type
OBJNAME	<input type="checkbox"/>	<input type="checkbox"/>	SWC OBJEDE	CHAR	32		0 Object name
METHOD	<input type="checkbox"/>	<input type="checkbox"/>	SWO VERB	CHAR	32		0 Object type component
PARAM	<input type="checkbox"/>	<input type="checkbox"/>	SWC ELEM	CHAR	32		0 Element
FIELD	<input type="checkbox"/>	<input type="checkbox"/>	NAME_FELD	CHAR	30		0 Field name

Fig 5. Table ZSHELPBOR

- (optional) Create maintenance dialog for table ZSHELPBOR, just in case you want to maintain assignments of helps and BOR manually. Good example of why you want to do that is if you want to use search help VMVA (contracts), you will need to explicitly specify business object, method and its parameters, because there is no explicit assignment of VMVA to any existing BOR parameters or fields. However, you don't *have to* maintain table explicitly – first time you call function module Z_BAPI_SEARCHHELP_METADATA with explicitly specified BOR data, the entry will be created in ZSHELPBOR automatically.

5. (SE11) Define structure ZBAPISHMETA:

Component	RTy...	Component type	Data Type	Length	Deci...	Short Description
TABNAME	<input type="checkbox"/>	TABNAME	CHAR	30	0	Table Name
FIELDNAME	<input type="checkbox"/>	FIELDNAME	CHAR	30	0	Field Name
SHLPNAME	<input type="checkbox"/>	SHLPNAME	CHAR	30	0	Name of a Search Help
SHLPTYPE	<input type="checkbox"/>	CHAR2	CHAR	2	0	Version Number Component
ISSIMPLE	<input type="checkbox"/>	DDSHSIMPLE	CHAR	1	0	Flag if a search help is an elementary search help
_INCLUDE	<input type="checkbox"/>	BAPIF4A		0	0	Transfer structure for BAPI_HELPVALUES_GET
OBJTYPE	<input type="checkbox"/>	SVC_OBJTYP	CHAR	10	0	Object Type
OBJNAME	<input type="checkbox"/>	SVC_OBJEDE	CHAR	32	0	Object name
METHOD	<input type="checkbox"/>	SVC_VERB	CHAR	32	0	Object type component
PARAM	<input type="checkbox"/>	SVC_ELEM	CHAR	32	0	Element
FIELD	<input type="checkbox"/>	NAME_FELD	CHAR	30	0	Field name
MAX_ROWS	<input type="checkbox"/>	BAPIMAXROW	INT4	10	0	Maximum number of lines of hits
DESCRIP	<input type="checkbox"/>	DESCRIP	CHAR	1	0	Only determine descriptions of help values with value 'X'
TYPE	<input type="checkbox"/>	BAPI_NTTYPE	CHAR	1	0	Message type: S Success, E Error, W Warning, I Info, A Abort
ID	<input type="checkbox"/>	SYMSGID	CHAR	20	0	Message Class
NUMBER	<input type="checkbox"/>	SYMSGNO	NUMC	3	0	Message Number
MESSAGE	<input type="checkbox"/>	BAPI_MSG	CHAR	220	0	Message Text
MESSAGE_Y1	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MESSAGE_Y2	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MESSAGE_Y3	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MESSAGE_Y4	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
ROW	<input type="checkbox"/>	BAPI_LINE	INT4	10	0	Lines in parameter
DDTEXT	<input type="checkbox"/>	DDTEXT	CHAR	60	0	Explanatory short text

Fig 6. Structure ZBAPISHMETA

6. (SE11) Define structure ZBAPISHDESC:

Component	RTy...	Component type	Data Type	Length	Deci...	Short Description
TABIX	<input type="checkbox"/>	SYTABIX	INT4	10	0	Index of Internal Tables
FIELDNAME	<input type="checkbox"/>	SHLPFIELD	CHAR	30	0	Name of a search help parameter
DATATYPE	<input type="checkbox"/>	DYNPTYPE	CHAR	4	0	ABAP/4 Dictionary: Screen data type for Screen Paint
FIELDTEXT	<input type="checkbox"/>	AS4TEXT	CHAR	60	0	Short Description of Repository Objects
REPTXT	<input type="checkbox"/>	REPTXT	CHAR	55	0	Heading
SCRTEXT_S	<input type="checkbox"/>	SCRTEXT_S	CHAR	10	0	Short Field Label
SCRTEXT_M	<input type="checkbox"/>	SCRTEXT_M	CHAR	20	0	Medium Field Label
SCRTEXT_L	<input type="checkbox"/>	SCRTEXT_L	CHAR	40	0	Long Field Label
SHLPOUTPUT	<input type="checkbox"/>	SHLPOUTPUT	CHAR	1	0	Flag for EXPORT Parameters for Search Help
SHLPSELPOS	<input type="checkbox"/>	SHLPSELPOS	NUMC	2	0	Position in dialog box of an elementary search help
SHLPLISPOS	<input type="checkbox"/>	SHLPLISPOS	NUMC	2	0	Position in the hit list of an elementary search help
OFFSET	<input type="checkbox"/>	DOFFSET	NUMC	6	0	Offset of a field
LENG	<input type="checkbox"/>	DDLENG	NUMC	6	0	Length (No. of Characters)

Fig 7. Structure ZBAPISHDESC

7. (SE11) Create index for table SWOTLQ, with index fields PARAMTYPE, REFSTRUCT and REFFIELD (I called mine Z01).
8. (SE80) Define function module Z_BAPI_SEARCHHELP_METADATA in a new function group of your choice (in my case its name is ZZDDIC). Code to paste is in Appendix A.
9. (SE80) Change top include of new function group using code in Appendix B.
10. (SE91) Define message class ZZDDIC, and add following messages:

Message	Message short text	Self-explanatory
000	& & & &	<input checked="" type="checkbox"/>
001	Either SHLPTYPE / SHLPNAME or TABNAME / FIELDNAME must be supplied	<input checked="" type="checkbox"/>
002	Non-empty SEARCHHELPS must be supplied	<input checked="" type="checkbox"/>
003	Must use SH / &1 with table &2, not CH / &2	<input checked="" type="checkbox"/>
004	Search help type & is not recognized. Use SH or CH instead.	<input checked="" type="checkbox"/>
005	No input help exist for field &1-&2	<input checked="" type="checkbox"/>
006	No handling of search helps of origin &	<input checked="" type="checkbox"/>
007	Field &1-&2 does not exist, check your input	<input checked="" type="checkbox"/>
008	Invalid parameter &1 of objtype/method &1 / &2	<input checked="" type="checkbox"/>
009	With BOR input, either PARAM must be a field, or a FIELD must be supplied	<input checked="" type="checkbox"/>

Fig 8. Messages of message class ZZDDIC.

Implementing Web Dynpro Java part

Next we create model and component ValueHelp. You can do it in a single DC, or in two separate DCs – one for model, one for WebDynpro component, it all depends on your company's coding standards and personal preferences. I'll describe using 2 DCs, doing it in 1 is even simpler.

1. Create local DC of type WebDynpro for model named *helpmodl* in package *sap.com*.
2. Create Adaptive RFC model named *ValueHelp* in package *com.sap.model* with 3 function modules from your R/3 system: *Z_BAPI_SEARCHHELP_METADATA*, *BAPI_HELPVALUES_GET*, and *BAPI_HELPVALUES_GET_SEARCHHELP*.
3. Add newly created model to public parts of your model DC with name *ValueHelpModel*.
4. Build *helpmodl* project.
5. Create local DC of type Web Dynpro called *helpcomp* in package *sap.com*.
6. In Web Dynpro Explorer perspective select Navigator tab, right-click on newly created *LocalDevelopment~helpcomp~sap.com* project, and select *Import...* from the context menu. In the next menu choose *Zip* file.
7. Choose the downloaded *helpcomp.zip* file that is [available here](#), and press *Finish* button.
8. Reload and rebuild *helpcomp* project.
9. Deploy model *helpmodl* and component *helpcomp* – that's it!

Using Value Help Builder application

Value Help Builder application is included with *helpcomp* Web Dynpro project. There are 3 distinct views in this application:

- Initial view
- Selection of simple value help from complex value help
- Code generator view

Let's examine each of the screens in more detail.

Initial view

Build code for value help

Enter value help info	and/or SAP table/structure & field
Value help type: Elementary help (SH) ▼	Tabname: <input type="text"/>
Value help name: MAT1	Fieldname: <input type="text"/>
(Optional) BOR reference	
Objtype: <input type="text"/>	
Method: <input type="text"/>	
Parameter: <input type="text"/>	
Field: <input type="text"/>	
<input type="button" value="Find"/>	

Fig 9. Start screen of Value Help Builder

Programmer enters known parameters required to identify SAP search help on initial screen. Usually some, or all parameters are known, for example:

- Value help type: SH; Value help name: MAT1N
- Tabname: VBAK; Fieldname: VKORG
- Value help type: SH; Value help name: VMVAE; Objtype: BUS2034, Method: CREATEFROMDATA, Parameter: CONTRACTHEADERIN

Selection of simple value help from complex value help

If chosen value help is complex (e.g., MAT1), the following screen shows up:

Value help is complex. Please select one of its elementary components

Value help:

Available fields			
	Field	Description	Length
<input checked="" type="checkbox"/>	AVAIL_QTY	Quantity	13
<input type="checkbox"/>	FUZZY	Description	40
<input type="checkbox"/>	MAKTG	Description	40
<input type="checkbox"/>	MATGROUPTXT	Matl Grp Desc.	20
<input type="checkbox"/>	MATNR	Material	18
<input type="checkbox"/>	MEINS	Base Unit	3
<input type="checkbox"/>	NUM_DOCS		10
<input type="checkbox"/>	PRODHIER1	HierNode Descr.	40
<input type="checkbox"/>	PRODHIER2	HierNode Descr.	40
<input type="checkbox"/>	PRODHIER3	HierNode Descr.	40

Row 1 of 12

Fig 10. Selection of simple value help from complex value help

It allows selecting elementary value help, with preview of which fields are available in it.

Code generator view

Select fields: OVS EVS/SVS first?

1 3 2 4

Input fields						Available fields				Output fields								
Field	Description	Length	Manual	applyInput	Auto-fill	Sel	Field	Description	Length	List	Field	Desc	Length	Return?	No 0			
<input type="checkbox"/>	MCCD1	Name	25	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	KUNNR	Customer	10	<input type="checkbox"/>	<input type="checkbox"/>	NAME1	Name 1	35	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	MCCD3	City	25	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VWORG	Sales Org.	4	<input type="checkbox"/>	<input type="checkbox"/>	ORT01	City	35	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	REGIO	Region	3	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VTWEG	Distr. Channel	2	<input type="checkbox"/>	<input type="checkbox"/>	REGIO	Rg	3	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	SORTL	Search term	10	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPART	Division	2	<input type="checkbox"/>	<input type="checkbox"/>	PSTLZ	PostalCode	10	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	SORT2	Search Term 2	20	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	KTOKD	Account group	4	<input type="checkbox"/>	<input type="checkbox"/>	SORTL	SearchTerm	10	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	KTOKD	Account group	4	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NAME1	Name	35	<input type="checkbox"/>	<input type="checkbox"/>	SORT2	Search Term 2	20	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	VKORG	Sales Org.	4	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ORT01	City	35	<input type="checkbox"/>	<input type="checkbox"/>	KTOKD	Group	4	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	VTWEG	Distr. Channel	2	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PSTLZ	Postal Code	10	<input type="checkbox"/>	<input type="checkbox"/>	VKORG	SOrg.	4	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	SPART	Division	2	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	REGIO	Region	3	<input type="checkbox"/>	<input type="checkbox"/>	VTWEG	DCh	2	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>								MCCD1	Name	25	<input type="checkbox"/>	<input type="checkbox"/>	SPART	Dv	2	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>								MCCD3	City	25	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	KUNNR	Customer	10	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>								SORTL	Search term	10	<input type="checkbox"/>	<input type="checkbox"/>						

Generate code

Insert this code at the end of wdDoInit() of relevant controller

```

SearchHelpsElement she;
she = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(she);
she.setABInfo(
//TODO: insert your own MDAtributeInfo reference here
);
she.setBORObjtype("BUS2105");
she.setBORMethod("CREATEFROMDATA");
she.setBORParameter("REGISTRATIONACCOUNTAS");
she.setBORField("PART_ACCT");
she.setShipname("ZDEB1");
she.setShiptype("SH");
she.setOVS(true);
she.setOVSReturnField("KUNNR");
AddInField(new String[] {

```

This code goes into @@others@@ section

```

class OVSCallback extends OVSCallbackAbstract {
public Vector[] addSelections(String node, String fld) {
Vector[] sel;
//TODO: replace selection values!!
sel = new Vector[] {
addToSel("KTOKD", "T", "EQ", "low", "high");
return sel;
}
private Vector addToSel(
Object low, Object high) {
Vector v = new Vector();
v.add(fld);
v.add(sign);
v.add(option);
v.add(low);

```

5

Back

Fig 11. Help designer and code generator screen

Upon selecting elementary value help, code generator screen appear. It consists of the following sections:

1. Value help type (OVS or SVS/EVS): selecting OVS produces code required to initialize OVS value help, EVS/SVS – the one for EVS/SVS value help. *Is first?* checkbox produces extra lines of code for data declarations and interface method *AssignValueHelps()* call. If defining multiple value helps, check *Is first?* for the first one, then uncheck it for the rest, and copy code in *wdDoInit()* section just before *AssignValueHelps()* call.
2. Available fields : fields that can be selected as input/output fields of value help. Columns are:
 1. *Sel* - field can be selected as input field if it shows black triangle,
 2. *Field* - SAP technical name of the help field,
 3. *Description* - description of the field,
 4. *Length* - output length of the field
 5. *List* – field can be selected as an output field if it shows black triangle

Also, the toolbar has buttons that select fields from this list into Input (for OVS/EVS/SVS) and/or Output field list (for OVS only).

3. Input fields: fields that can be populated during selection, by user or program. In addition to already mentioned columns, it has radiobuttons *Manual* and *Auto-fill*. If *Manual* is selected, field is available for user input, and if *Auto-fill* is selected, field is not visible and value is transferred to query by method *addSelections()* of generated class *OVSCallback*. Also checkbox *applyInput*, in combination with *Manual*, results in code that prepopulates value of user input on selection (using method *applyInputValues()* of generated class *OVSCallback*, which is in turn called within internal implementation of method *applyInputValues()* of interface *IWDOVSContextNotificationListener*). For example, when looking up sold-to customer during creation of sales document, customer account group is already known from the fact that it is a sold-to customer, and sales org., distribution channel and division may be available from some other place, so it makes sense to default those values in search help as shown in the example screenshot above. Note, that *Auto-fill* is only visible for OVS code generation, in EVS/SVS it is considered that whatever fields are selected in Input fields are auto-filled.

Toolbar for this table allows moving fields up and down in the list (for OVS help only), which changes sequence in which those fields are shown in OVS selection.

4. Output fields: fields that are shown as a result of user selection. This section comes in two flavors, one for OVS (show in screenshot above), and another – for EVS/SVS:

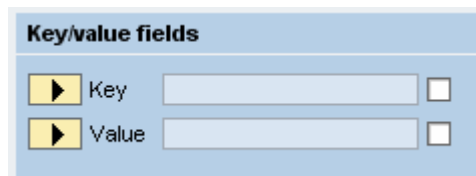



Fig 12. EVS/SVS design – output fields

In case of OVS, the following additional columns are available:

1. *Return?*: Places a snippet in the code that allows transfer of additional values from selected record (e.g., material description can also be populated in addition to material number). Used by function *applyResult* of *IWDOVSContextNotificationListener* interface.
2. *No 0*: If checked, removes leading 0 in field value. I found this handy with material and customer numbers. Just remember that if those values are used later in most standard BAPIs, they will have to be formatted back to SAP internal representation (i.e., 0-padded).

Emphasized row means that this field is main return field. To make some other field main return field, use  button.

EVS/SVS case is much simpler, since it's based on key/value pair. Checkbox behind field names has the same function as one in *No 0* column of OVS help.

5. Generated code for *wdInit()*: After all fields are selected as desired in the top portion of the view, click *Generate* button. This will produce 1 or 2 snippets of code. First one is a piece of code that has to be copied to *wdInit()* method of either component or view controller (depending upon where attribute to which value help is assigned is declared). Second one, if generated, goes into *@@other@@* section at the end of class definition for the same controller. Some changes are required to the code after it is copied into your application (marked by TODO: tags):
 1. Replace comment after **shel.setAttInfo(** with real context attribute;
 2. Change name of used ValueHelp interface with the name that you defined (unless, when adding used component, you used ValueHelp as a name of used Web Dynpro Component (see usage examples below).
 3. Replace template values in auto-fill section for selection in method *addSelections* of generated class *OVSCallback*.
 4. Replace *<IWDNodeElement>*,*<AttName>* in method *applyResult* of generated class *OVSCallBack* with values for node element and its attribute.
 5. Replace “<replace with value>” in method *applyInputValues* with actual values that should show up as default user inputs in OVS selections.

How to use generated code. Example.

In this example we'll build a view with 2 input and 1 output fields. Input fields will be country (with SVS lookup) and regions of this country (with OVS lookup); output field will show selected region's descriptions.

To follow this example:

1. Create another local DC of type Web Dynpro, e.g. *helptest*.
2. Add *helpcomp* to Used DCs.
3. Create new Web Dynpro Component *Test*
4. Add *ValueHelp* as used Web Dynpro Component under name *ValueHelp*.
5. Add *ValueHelp* as a required controller of the component controller.
6. Expand *Component Controller* subtree and double-click link to *ValueHelp Interface Controller*.
7. Drag and drop *SearchHelps* node from interface controller context to component controller context. Select all field and subnodes.
8. Add *Test* component controller to required controllers of view *TestView*.
9. Create application Test.
10. Create 2 context attributes, *Country* and *Region* of type String.
11. Create and map above attributes in view *TestView*.
12. Place input fields on layout; use *DropDownByKey* for *Country* and *InputField* for *Region*

We'll start with country field.

Initial screen selection in ValueHelp Builder: tabname: KNA1, fieldname: LAND1.

Resulting screen:

The screenshot shows the ValueHelp Builder interface with three panels: 'Input fields', 'Available fields', and 'Key/value fields'. The 'Available fields' panel contains a table with the following data:

Sel	Field	Description	Length	List
<input type="checkbox"/>	LAND1	Country	3	<input type="checkbox"/>
<input checked="" type="checkbox"/>	LANDX	Name	15	<input type="checkbox"/>

The 'Key/value fields' panel shows 'Key' set to 'LAND1' and 'Value' set to 'LANDX'. Below the panels, there is a 'Generate code' button and two text areas for code. The left text area contains the following code:

```

Insert this code at the end of wdDoInit() of relevant controller

ISearchHelpsElement shel;
shel = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(shel);
shel.setAttInfo(
//TODO: Insert your own MDAttributeInfo reference here
);
shel.setBORObjtype("EQUI_EXT");
shel.setBORMethod("GETDETAIL");
shel.setBORParameter("EQUIMASTER");
shel.setBORField("MANCOUNTRY");
shel.setShlname("TOOS");
shel.setShotype("CH");
shel.setTabname("KNA1");
shel.setFieldname("LAND1");
shel.setIsOVS(false);
  
```

The right text area contains the following code:

```

This code goes into @others@@ section

private void AddOutField(String[] s, ISearchHelpsElement shel) {
    IOutFieldsElement out;
    for (int i = 0; i < s.length; i++) {
        out = shel.nodeOutFields().createOutFieldsElement();
        shel.nodeOutFields().addElement(out);
        out.setFieldName(s[i]);
    }
}
  
```

Fig 13. Example: Designing SVS for country lookup

Note that *Is first?* is checked, because we are generating code for the 1st out of 2 value helps. This generates extra declaration and method call in *wdDoInit()* section.

We'll copy generated code to *wdDoInit()* of component controller, and change it so it looks like this:

```

ISearchHelpsElement shel;
shel = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(shel);
shel.setAttInfo(wdContext.getNodeInfo().getAttribute("Country"));
shel.setBORObjtype("EQUI_EXT");
  
```

```

shel.setBORMethod("GETDETAIL");
shel.setBORParameter("EQUIMASTER");
shel.setBORField("MANCOUNTRY");
shel.setShlpname("T005");
shel.setShlptype("CH");
shel.setTabname("KNA1");
shel.setFieldName("LAND1");
shel.setIsOVS(false);
AddOutField(new String[] { "OLAND1", "LANDX" }, shel);
//TODO: Replace ValueHelpInterface with name of your interface
wdThis.wdGetValueHelpInterface().AssignValueHelps();

```

And code that we paste to the `@@other@@` section will look like this after changes:

```

private void AddOutField(String[] s, ISearchHelpsElement shel) {
    IOutFieldsElement outf;
    for (int i = 0; i < s.length; i++) {
        outf = shel.nodeOutFields().createOutFieldsElement();
        shel.nodeOutFields().addElement(outf);
        outf.setFieldName(s[i]);
    }
}

```

Keep in mind, you will probably need to press Ctrl-Shift-O to resolve imports declarations.

Now that country field is taken care of, let's add region. On initial screen we'll use tabname KNA1, and fieldname REGIO. Since we already selected the country, we add it to input fields and auto-fill it:

The screenshot shows the 'Select fields' dialog with the following data:

Input fields							Available fields					Output fields						
Field	Description	Length	Manual	applyInput	Auto-fill		Sel	Field	Description	Length	List		Field	Desc	Length	Return?	No 0	
LAND1	Country	3	<input type="radio"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	BLAND	Region	3	<input type="checkbox"/>		<input checked="" type="checkbox"/>	BLAND	Rg	3	<input type="checkbox"/>	<input type="checkbox"/>
							<input checked="" type="checkbox"/>	LAND1	Country	3	<input type="checkbox"/>		<input type="checkbox"/>	BEZEI	Description	20	<input type="checkbox"/>	<input type="checkbox"/>
							<input type="checkbox"/>	BEZEI	Description	20	<input type="checkbox"/>							

Below the tables, the 'Generate code' button is highlighted. The code editor shows the following code:

```

Insert this code at the end of wdDoInit() of relevant controller
shel = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(shel);
shel.setAttInfo(
//TODO: insert your own MDAttributeInfo ref.
);
shel.setBORObjtype("BUS1171");
shel.setBORMethod("GETDETAIL");
shel.setBORParameter("PRODUCTLINE");
shel.setBORField("REGIONORIG");
shel.setShlpname("T005");
shel.setShlptype("CH");
shel.setTabname("KNA1");
shel.setFieldName("REGIO");
shel.setIsOVS(true);
shel.setOVSReturnField("BLAND");

Class OVSCallback extends OVSCallbackAbstract {
public Vector[] addSelections(String node, String fld) {
//face selection values!!
sel = new Vector[] {
addToSel("LAND1", "T", "EQ", "low", "high");
return sel;
}
private Vector addToSel(Object fld, Object sign, Object option,
Object low, Object high) {
Vector v = new Vector();
v.add(fld);
v.add(sign);
v.add(option);
v.add(low);
}
}

```

Fig 14. Example: Designing OVS for state/region lookup

Is first? Checkbox was unchecked, so `wdDoInit()` part can be copied as is and only attribute has to be specified. `@@others@@` section should be copied as is, but duplicates from previous insert should be deleted (in this case it will be method `AddOutField()`). Our final code should look like this:

`wdDoInit()`:

```

ISearchHelpsElement shel;
shel = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(shel);
shel.setAttInfo(wdContext.getNodeInfo().getAttribute("Country"));
shel.setBORObjtype("EQUI_EXT");
shel.setBORMethod("GETDETAIL");
shel.setBORParameter("EQUIMASTER");
shel.setBORField("MANCOUNTRY");

```

```

shel.setShlpname("T005");
shel.setShlptype("CH");
shel.setTabname("KNA1");
shel.setFieldname("LAND1");
shel.setIsOVS(false);
AddOutField(new String[] { "OLAND1", "LANDX" }, shel);

shel = wdContext.nodeSearchHelps().createSearchHelpsElement();
wdContext.nodeSearchHelps().addElement(shel);
shel.setAttInfo(wdContext.getNodeInfo().getAttribute("Region"));
shel.setBORObjtype("BUS1503");
shel.setBORMethod("GETLIST");
shel.setBORParameter("OBJECTADDRESS");
shel.setBORField("REGION");
shel.setShlpname("T005S");
shel.setShlptype("CH");
shel.setTabname("KNA1");
shel.setFieldname("REGIO");
shel.setIsOVS(true);
shel.setOVSTReturnField("BLAND");
AddInField(new String[] { "BLAND", "BEZEI" }, shel);
AddOutField(new String[] { "BLAND", "BEZEI" }, shel);
shel.setICallBack(ovsCB);

wdThis.wdGetValueHelpInterface().AssignValueHelps();

```

and @@others@@:

```

//@@begin others
private void AddOutField(String[] s, ISearchHelpsElement shel) {
    IOutFieldsElement outf;
    for (int i = 0; i < s.length; i++) {
        outf = shel.nodeOutFields().createOutFieldsElement();
        shel.nodeOutFields().addElement(outf);
        outf.setFieldName(s[i]);
    }
}

private void AddInField(String[] s, ISearchHelpsElement shel) {
    IInFieldsElement inf;
    for (int i = 0; i < s.length; i++) {
        inf = shel.nodeInFields().createInFieldsElement();
        shel.nodeInFields().addElement(inf);
        inf.setFieldName(s[i]);
    }
}

class OVSCallback extends OVSCallbackAbstract {
    public Vector[] addSelections(String node, String fld) {
        Vector[] sel;
        sel = new Vector[] {
            addToSel("LAND1", "I", "EQ",
                wdContext.currentContextElement().getCountry(), null);
        };
        return sel;
    }

    private Vector addToSel(Object fld, Object sign, Object option, Object low, Object high) {
        Vector v = new Vector();
        v.add(fld);
        v.add(sign);
        v.add(option);
        v.add(low);
        v.add(high);
        return v;
    }
}

```

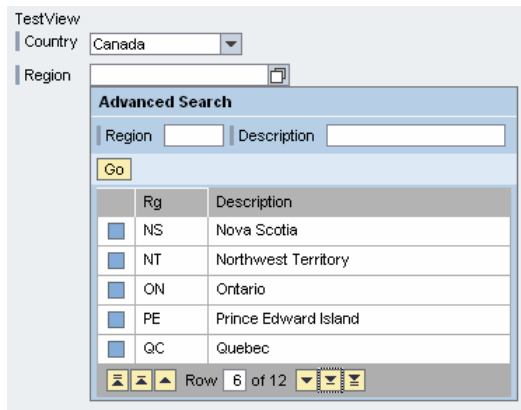
```

public Vector[] applyResults(String node, String fld) {
    return new Vector[] {
        addToSel(wdContext.currentContextElement(), "RegionName", "BEZEI", null, null);
    }
}

IOVSCallback ovsCB = new OVSCallback();
//@@@end

```

And that's the end result you should get after selecting Canada and pressing Go button on OVS:



After selecting BC from the list:



Appendix A. Code for function module Z_BAPI_SEARCHHELP_METADATA

FUNCTION z_bapi_searchhelp_metadata.

```

* "-----
* " "Local Interface:
* " TABLES
* " SEARCHHELPS STRUCTURE ZBAPISHMETA
* " SEARCHHELPDESC STRUCTURE ZBAPISHDESC
* "-----
* SEARCHHELPS must supply either TABNAME/FIELDNAME or
* SHLPTYPE / SHLPNAME. ISSIMPLE will be returned by BAPI; if it is
* not blank, then either call to BAPI_HELPVALUES_GET_SEARCHHELP is
* required to chose existing elementary searchhelp, or programmer
* should already know which simple help to use
* Priority of parameters:
* SHLPTYPE / SHLPNAME
* TABNAME / FIELDNAME
* "-----
*----- data declarations -----
data: v_tabix like sy-tabix, l_outfields type i, l_from type sy-tabix.
data: lt_fielddescr like dfies occurs 1 with header line.
data: lt_descr_for_hv like bapif4e occurs 0,
      lt_helps type table of bapishlp.
data: lw_help like line of lt_helps.
data: lw_shm type shlp_descr, lw_ret type bapireturn.
field-symbols: <fd> like line of lw_shm-fielddescr,
               <fp> like line of lw_shm-fieldprop,
               <dhv> like line of lt_descr_for_hv,
               <hlp> like line of lt_helps.

clear v_index_changed.
* pre-read most commonly used fields

```



```

data: lt_dtel like table of w_dtel.
select distinct rollname tabname into table lt_dtel
  from dd031
  where tabname in ('BAPISDITM','BAPIPARNR','BAPISCHDL',
    'BAPISDHD1',
    'BAPIMEPOHEADER','BAPIMEPOITEM','BAPIMEPOACCOUNT',
    'BAPI_ALM_ORDER_HEADER_E','BAPI_ALM_ORDER_OPERATION',
    'BAPI_ALM_ORDER_COMPONENT') and
    as4local = 'A'.
sort lt_dtel by rollname.
delete adjacent duplicates from lt_dtel comparing rollname.
t_dtel = lt_dtel.
*----- processing -----
loop at searchhelps.
  v_tabix = sy-tabix.
  clear: dd031, zshelpbor.
  clear: lw_shm, w_shlp, searchhelpdesc.
  refresh: lt_fielddescr, lt_descr_for_hv, t_shlp.

  if not searchhelps-shlptype is initial and
    not searchhelps-shlpname is initial.
    "by explicit search help
    case searchhelps-shlptype.
      when 'CH' or 'CT'. "value table with text
        perform get_meta_ch changing searchhelps.
      when 'FV'.
        perform get_meta_fv changing searchhelps.
      when 'SH'. "explicitly defined search help
        perform get_meta_sh changing searchhelps.
      when others. "error
        error_fill 'ZZDDIC' '004' searchhelps-shlptype ' ' '.
    endcase.
  elseif not searchhelps-tabname is initial and
    not searchhelps-fieldname is initial.
    "by table/field
    perform get_meta_by_field changing searchhelps.
  elseif not searchhelps-objtype is initial and
    not searchhelps-method is initial and
    not searchhelps-param is initial.
    select single *
      from swotlq
      where lobjtype = searchhelps-objtype and
        verb = searchhelps-method and
        paramtype = 'M' and element = searchhelps-param.
    if sy-subrc <> 0.
      message e008.
      error_fill 'ZZDDIC' '004' searchhelps-param
        searchhelps-objtype searchhelps-method '.
    else.
      if swotlq-reffield is initial.
        if searchhelps-field is initial.
          message e009.
          error_fill 'ZZDDIC' '009' ' ' ' ' ' '.
          modify searchhelps.
          continue.
        else.
          swotlq-reffield = searchhelps-field.
        endif.
      endif.
    endif.
  endif.

```

```

call function 'F4IF_DETERMINE_SEARCHHELP'
  exporting
    tabname          = swotlq-refstruct
    fieldname        = swotlq-reffield
  importing
    shlp              = lw_shm
  exceptions
    field_not_found  = 1
    no_help_for_field = 2
    inconsistent_help = 3
    others            = 4.
if sy-subrc <> 0.
  error_fill sy-msgid sy-msgno
    sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
else.
  searchhelps-shlpname = lw_shm-shlpname.
  searchhelps-shlptype = lw_shm-shlptype.
  searchhelps-issimple = lw_shm-intdescr-issimple.
  perform add_to_buffer using searchhelps.
endif.
endif.
else.
  "Either SHLPTYPE / SHLPNAME or TABNAME / FIELDNAME must
  "be supplied
  error_fill 'ZZDDIC' '001' ' ' ' ' ' '.
endif.
if searchhelps-shlpname is initial and searchhelps-type is initial.
  error_fill 'ZZDDIC' '000' 'No search help was found' ' ' ' ' '.
endif.
modify searchhelps.
check searchhelps-type is initial and "no errors
  not searchhelps-shlpname is initial and "help found
  searchhelps-issimple = 'X'.
call function 'F4IF_GET_SHLP_DESCR'
  exporting
    shlpname = searchhelps-shlpname
    shlptype = searchhelps-shlptype
  importing
    shlp = lw_shm.
searchhelps-ddtext = lw_shm-intdescr-ddtext.
modify searchhelps transporting ddtext.
* Transfer DDIC-field's description of helpvalues to intefacetable
searchhelpdesc-tabix = v_tabix.
lt_fielddescr[] = lw_shm-fielddescr[].
call function 'BALW_SHLP_BAPI_FILL'
  exporting
    shlp_method          = lw_shm
  tables
    lt_descrip           = lt_fielddescr
    description_for_helpvalues = lt_descr_for_hv.
clear l_outfields.
describe table searchhelpdesc lines l_from.
loop at lt_descr_for_hv assigning <dhv>.
  move-corresponding <dhv> to searchhelpdesc.
  read table lw_shm-fielddescr assigning <fd>
    with key fieldname = <dhv>-fieldname.
  if sy-subrc = 0.
    searchhelpdesc-datatype = <fd>-datatype.
  endif.

```

```

read table lw_shm-fieldprop assigning <fp>
  with key fieldname = <dhv>-fieldname.
if sy-subrc = 0.
  searchhelpdesc-shlpoutput = <fp>-shlpoutput.
  if not <fp>-shlpoutput is initial.
    "count output fields for later
    add 1 to l_outfields.
  endif.
  searchhelpdesc-shlpselpos = <fp>-shlpselpos.
  searchhelpdesc-shlplispos = <fp>-shlplispos.
endif.
append searchhelpdesc.
endloop.
* if more than 1 output fields, and reference field is specified, try and
* return only that field
if l_outfields > 1 and not searchhelps-fieldname is initial.
  loop at searchhelpdesc from l_from where shlpoutput <> ' '.
    read table lw_shm-fielddescr assigning <fd>
      with key fieldname = searchhelpdesc-fieldname.
    if sy-subrc = 0 and <fd>-rollname <> dd031-rollname and
      <fd>-domname <> dd031-domname.
      clear searchhelpdesc-shlpoutput.
      modify searchhelpdesc transporting shlpoutput.
    endif.
  endloop.
endif.
endloop.
check v_index_changed = 'X'.
commit work.
ENDFUNCTION.

*-----*
***INCLUDE LZZDDICF01 .
*-----*
*&-----*
*&      Form find_usage_in_bo
*&-----*
form find_usage_in_bo using      p_tabname
                             p_fieldname.
statics: lt_swotlq type sorted table of swotlq
         with non-unique key refstruct reffield.
clear swotlq.
read table lt_swotlq into swotlq
  with table key refstruct = p_tabname reffield = ' '.
if sy-subrc <> 0.
  if p_tabname(4) = 'BAPI'.
    read table lt_swotlq into swotlq
      with table key refstruct = p_tabname reffield = p_fieldname.
  endif.
endif.
check sy-subrc <> 0.
select * "try whole structure as parameter
       from swotlq
       where refstruct = p_tabname and reffield = ' ' and
             paramtype = 'M'.
check swotlq-lobjtype(1) <> '/'.
exit.
endselect.
if ( sy-subrc <> 0 or swotlq-lobjtype(1) = '/' ) and

```

```

    p_tabname(4) = 'BAPI'.
    select * "try field as parameter
      from swotlq
      where refstruct = p_tabname and reffield = p_fieldname and
        paramtype = 'M'.
      check swotlq-lobjtype(1) <> '/'.
      exit.
    endselect.
  endif.
  if sy-subrc <> 0 or swotlq-lobjtype(1) = '/'.
    clear swotlq.
    swotlq-refstruct = p_tabname.
    insert swotlq into table lt_swotlq.
    if p_tabname(4) = 'BAPI'.
      swotlq-reffield = p_fieldname.
      insert swotlq into table lt_swotlq.
    endif.
  else.
    insert swotlq into table lt_swotlq.
  endif.
endform.
" find_usage_in_bo
*&-----*
*&      Form  get_meta_ch
*&-----*
form get_meta_ch changing searchhelps type zbapishmeta.
  types: begin of ty_031,
         tabname type dd031-tabname,
         fieldname type dd031-fieldname,
         shlporigin type dd031-shlporigin,
         end of ty_031.
  data: lt_031 type table of ty_031.
  data: lw_031 like line of lt_031.

  select single * from dd021
    where tabname = searchhelps-shlpname and as4local = 'A'.
  if sy-subrc <> 0.
    error_fill 'E2' '007' searchhelps-shlpname ' ' '.
    return.
  endif.
  "is it proper value help?
  select single *
    from dd351
    where tabname = dd021-tabname and fieldname = ' ' and
      as4local = 'A'.
  if sy-subrc = 0.
    error_fill 'ZZDDIC' '003' dd351-shlpname dd021-tabname ' ' '.
    return.
  endif.
  "try buffer
  select single * from zshelpbor
    where shlpname = searchhelps-shlpname and shlptype = 'CH'.
  if sy-subrc = 0.
    searchhelps-issimple = zshelpbor-issimple.
    searchhelps-objtype = zshelpbor-objtype.
    searchhelps-method = zshelpbor-method.
    searchhelps-param = zshelpbor-param.
    searchhelps-field = zshelpbor-field.
    return.
  endif.

```

```

"Is it a checktable for any BAPI field?
select tabname fieldname shlporigin into table lt_031
  from dd031
  where checktable = dd021-tabname and as4local = 'A'.
delete lt_031 where tabname(4) <> 'BAPI' and shlporigin <> 'P'.
loop at lt_031 into lw_031.
  perform find_usage_in_bo
    using lw_031-tabname lw_031-fieldname.
  if not swotlq-lobjtype is initial.
    searchhelps-issimple = 'X'.
    searchhelps-objtype = swotlq-lobjtype.
    searchhelps-method = swotlq-verb.
    searchhelps-param = swotlq-element.
    if swotlq-reffield is initial.
      searchhelps-field = lw_031-fieldname.
    endif.
    perform add_to_buffer using searchhelps.
    exit. "LOOP
  endif.
endloop.
endform.
                                " get_meta_ch
*&-----*
*&      Form get_meta_by_field
*&-----*
form get_meta_by_field changing searchhelps type zbapishmeta.
  data: lt_031 type table of dd031.
  data: lw_031 like line of lt_031.
  clear swotlq.
* table OK?
  select single *
    from dd021
    where tabname = searchhelps-tabname and as4local = 'A'.
  if sy-subrc <> 0.
    error_fill 'E2' '007' searchhelps-tabname ' ' '.
    return.
  endif.
* field exists in table?
  select single *
    from dd031
    where tabname = dd021-tabname and
      fieldname = searchhelps-fieldname and
      as4local = 'A'.
  if sy-subrc <> 0.
    error_fill 'ZZDDIC' '007'
      searchhelps-tabname searchhelps-fieldname ' ' '.
    return.
  endif.
* what help is available?
  case dd031-shlporigin.
    when ' '. "No input help exists?
      perform find_by_data_element changing searchhelps.
      if searchhelps-objtype is initial.
        error_fill 'ZZDDIC' '005' dd031-tabname dd031-fieldname ' ' '.
      endif.
    when 'D'. "Explicit for data element
      select single shlpname into searchhelps-shlpname
        from dd041
        where rollname = dd031-rollname and as4local = 'A'.
      searchhelps-shlptype = 'SH'.

```

```

perform get_meta_sh changing searchhelps.
when 'F'. "Domain fixed values or FM
searchhelps-shlpname = dd031-domname.
searchhelps-shlptype = 'FV'.
perform get_meta_fv using searchhelps.
when 'P'. "Input help implemented with check table
clear: searchhelps-shlpname, searchhelps-shlptype.
if not dd031-checktable is initial and
    dd031-checktable <> '*' and
    dd031-checktable <> '0'.
    searchhelps-shlpname = dd031-checktable.
    searchhelps-shlptype = 'CH'.
else. "get from domain
    select single entitytab into searchhelps-shlpname
        from dd011
        where domname = dd031-domname and as4local = 'A'.
endif.
perform get_meta_ch changing searchhelps.
if searchhelps-id = 'ZZDDIC' and searchhelps-number = '003'.
    clear: searchhelps-type, searchhelps-id,
        searchhelps-number, searchhelps-message.
    searchhelps-shlptype = 'SH'.
    searchhelps-shlpname = dd351-shlpname.
    perform get_meta_sh changing searchhelps.
endif.
when 'X'. "Explicit attachment to field
select single * from dd351
    where tabname = dd031-tabname and
        fieldname = dd031-fieldname and as4local = 'A'.
searchhelps-shlptype = 'SH'.
searchhelps-shlpname = dd351-shlpname.
perform get_meta_sh changing searchhelps.
when others.
    error_fill 'ZZDDIC' '006' dd031-shlporigin ' ' ' '.
endcase.
if searchhelps-shlpname is initial.
select single * from swotlq
    where lobjtype = searchhelps-objtype and
        verb = searchhelps-method and
        paramtype = 'M' and element = searchhelps-param.
check sy-subrc = 0.
if swotlq-reffield is initial.
    swotlq-reffield = searchhelps-field.
endif.
data: lw_shm type shlp_descr.
call function 'F4IF_DETERMINE_SEARCHHELP'
    exporting
        tabname          = swotlq-refstruct
        fieldname        = swotlq-reffield
    importing
        shlp             = lw_shm
    exceptions
        field_not_found = 1
        no_help_for_field = 2
        inconsistent_help = 3
        others           = 4.
if sy-subrc <> 0.
    call function 'F4IF_DETERMINE_SEARCHHELP'
        exporting

```

```

        tabname          = searchhelps-tabname
        fieldname        = searchhelps-fieldname
importing
        shlp             = lw_shm
exceptions
        field_not_found = 1
        no_help_for_field = 2
        inconsistent_help = 3
        others           = 4.
endif.
if sy-subrc = 0.
    searchhelps-shlpname = lw_shm-shlpname.
    searchhelps-shlptype = lw_shm-shlptype.
endif.
endif.
endform.          " get_meta_by_field
*&-----*
*&      Form  get_meta_sh
*&-----*
form get_meta_sh  changing searchhelps type zbapishmeta.
    data: lt_auth type sorted table of bapif4t with non-unique key dtel.
    data: ret type bapireturn.
    data: lt_35l type sorted table of dd35l
           with non-unique key tabname fieldname.
    data: begin of lw_03l,
           tabname type dd03l-tabname,
           fieldname type dd03l-fieldname,
           rollname type dd03l-rollname,
           shlporigin type dd03l-shlporigin,
    end of lw_03l,
    lt_03l like table of lw_03l.

select single * from dd30l
    where shlpname = searchhelps-shlpname and as4local = 'A'.
* does search help exist at all?
if sy-subrc <> 0.
    error_fill 'E2' '012' searchhelps-shlpname ' ' ' '.
    return.
endif.
searchhelps-issimple = dd30l-issimple.
* if BOR data is forced, at least make sure it's valid
if not searchhelps-objtype is initial and
    not searchhelps-method is initial and
    not searchhelps-param is initial.
    data: l_return.
    select single * from swotlq
        where lobjtype = searchhelps-objtype and
              verb = searchhelps-method and
              paramtype = 'M' and
              element = searchhelps-param.
if sy-subrc = 0.
    if swotlq-reffield is initial.
        if not searchhelps-field is initial.
            select single *
                from dd03l
                where tabname = swotlq-refstruct and
                      fieldname = searchhelps-field and
                      as4local = 'A'.
            if sy-subrc = 0.

```

```

        searchhelps-field = dd031-fieldname.
        swotlq-reffield = searchhelps-field.
        l_return = 'X'.
    endif.
else.
    l_return = 'X'.
endif.
else.
    clear searchhelps-field.
    select single *
        from dd031
        where tabname = swotlq-refstruct and
            fieldname = swotlq-reffield and
            as4local = 'A'.
    l_return = 'X'.
endif.
endif.
if l_return is initial.
    clear: searchhelps-objtype, searchhelps-method,
        searchhelps-param, searchhelps-field.
else.
    dd351-tabname = swotlq-refstruct.
    perform check_auth changing searchhelps ret.
    if ret-type is initial.
        select single * from zshelpbor
            where shlpname = searchhelps-shlpname and shlptype = 'SH'.
        if sy-subrc <> 0.
            perform add_to_buffer using searchhelps.
        endif.
    else.
        searchhelps-type = 'E'.
        searchhelps-message = 'No authorization in BAPIF4T'.
    endif.
    return.
endif.
endif.
endif.
"try buffer
select single * from zshelpbor
    where shlpname = searchhelps-shlpname and shlptype = 'SH'.
if sy-subrc = 0.
    searchhelps-issimple = zshelpbor-issimple.
    searchhelps-objtype = zshelpbor-objtype.
    searchhelps-method = zshelpbor-method.
    searchhelps-param = zshelpbor-param.
    searchhelps-field = zshelpbor-field.
    return.
endif.
* if part of complex search help, find all levels on which can be used
select dd31s~shlpname dd301~issimple into table t_shlp
    from dd31s join dd301 on dd301~shlpname = dd31s~shlpname and
        dd301~as4local = dd31s~as4local
    where subshlp = dd301-shlpname and dd31s~as4local = 'A'.
w_shlp-shlpname = dd301-shlpname.
collect w_shlp into t_shlp.
clear shlp_method.
call function 'F4IF_GET_SHLP_DESCR'
    exporting
        shlpname = dd301-shlpname
        shlptype = 'SH'

```



```

importing
  shlp      = shlp_method.
* maybe one of main helps is already in buffer?
loop at t_shlp into w_shlp.
  select single * from zshelpbor
    where shlpname = w_shlp-shlpname and shlptype = 'SH'.
  if sy-subrc = 0.
    searchhelps-issimple = zshelpbor-issimple.
    searchhelps-objtype = zshelpbor-objtype.
    searchhelps-method = zshelpbor-method.
    searchhelps-param = zshelpbor-param.
    searchhelps-field = zshelpbor-field.
    return.
  endif.
endloop.
select * from bapif4t into table lt_auth.
*=====
* attempt 1: find explicit shlp assignment to BAPI structures
* find which check table is associated with this search help
select * into table lt_35l
  from dd35l for all entries in t_shlp
  where tabname like 'BAPI%' and
    shlpname = t_shlp-shlpname and as4local = 'A'.
delete lt_35l where fieldname = ' '.
loop at lt_35l into dd35l.
  perform find_explicit_help changing searchhelps.
  check not searchhelps-objtype is initial.
  return.
endloop.
check searchhelps-objtype is initial. "nothing found yet
*=====
* attempt 2: find assignment via checktable
select * into table lt_35l
  from dd35l for all entries in t_shlp
  where fieldname = ' ' and shlpname = t_shlp-shlpname.
loop at lt_35l into dd35l.
  if shlp_method-intdescr-selmtype = 'H' or
    shlp_method-intdescr-selmtype = 'T' or
    shlp_method-intdescr-selmtype = 'X'.
    clear ret.
    perform authoritychecks(saplbfbv)
      using 'DUMMY' ' ' ' ' dd35l-tabname ' '
      changing shlp_method ret.
    check ret-type <> 'E'.
  endif.

select tabname fieldname rollname shlporigin into table lt_03l
  from dd03l
  where checktable = dd35l-tabname and
    as4local = 'A'.
delete lt_03l where tabname(4) <> 'BAPI'.
sort lt_03l by rollname.
data l_auth.
loop at lt_03l into lw_03l where shlporigin <> 'P'.
  if shlp_method-intdescr-selmtype <> 'H' and
    shlp_method-intdescr-selmtype <> 'T' and
    shlp_method-intdescr-selmtype <> 'X'.
    on change of lw_03l-rollname.
      read table lt_auth

```

```

        with table key dtel = lw_03l-rollname
        transporting no fields.
    if sy-subrc = 0.
        l_auth = 'X'.
    else.
        clear l_auth.
    endif.
endon.
check l_auth = 'X'.
endif.

perform find_usage_in_bo using lw_03l-tabname lw_03l-fieldname.
check not swotlq-lobjtype is initial.
searchhelps-objtype = swotlq-lobjtype.
searchhelps-method = swotlq-verb.
searchhelps-param = swotlq-element.
if swotlq-reffield is initial.
    searchhelps-field = lw_03l-fieldname.
    perform add_to_buffer using searchhelps.
endif.
return.
endloop.
endloop.
check searchhelps-objtype is initial. "nothing found yet
* attempt 3: find assignment via data element
if dd03l-rollname is initial.
    select single * from dd35l
        where shlpname = searchhelps-shlpname and as4local = 'A'.
    if sy-subrc = 0.
        select single *
            from dd03l
            where tabname = dd35l-tabname and
                fieldname = dd35l-fieldname and as4local = 'A'.
    endif.
endif.
if not searchhelps-tabname is initial and
    not searchhelps-fieldname is initial.
    select single * from dd03l
        where tabname = searchhelps-tabname and
            fieldname = searchhelps-fieldname and as4local = 'A'.
    perform find_by_data_element changing searchhelps.
endif.
check searchhelps-objtype is initial. "nothing found yet
* final attempt: let's try to find with non BAPI*
select distinct tabname fieldname into table lt_35l
    from dd35l for all entries in t_shlp
    where shlpname = t_shlp-shlpname and as4local = 'A'.
loop at lt_35l into dd35l.
    perform find_explicit_help changing searchhelps.
    check not searchhelps-objtype is initial.
    return.
endloop.
endform.
                                " get_meta_sh
*&-----*
*&      Form find_explicit_help
*&-----*
form find_explicit_help changing searchhelps type zbapishmeta.
perform find_usage_in_bo using dd35l-tabname dd35l-fieldname.
if not swotlq-lobjtype is initial.

```

```

select single rollname into dd03l-rollname
  from dd03l
  where tabname = dd35l-tabname and
        fieldname = dd35l-fieldname and
        as4local = 'A' and shlporigin = 'X'.
check sy-subrc = 0.
data ret type bapireturn.
perform check_auth changing searchhelps ret.
check ret-type is initial.
searchhelps-objtype = swotlq-lobjtype.
searchhelps-method = swotlq-verb.
searchhelps-param = swotlq-element.
if swotlq-reffield is initial.
  searchhelps-field = dd35l-fieldname.
endif.
perform add_to_buffer using searchhelps.
return.
endif.
endform.                                " find_explicit_help
*&-----*
*&      Form  find_by_data_element
*&-----*
form find_by_data_element  changing searchhelps type zbapishmeta.
  data: lt_03l type table of dd03l.
  data: lw_03l like line of lt_03l.
  * for performance reason, get some most common used data elements
  read table t_dtel into w_dtel
    with table key rollname = dd03l-rollname.
  if sy-subrc <> 0.
    select * into table lt_03l
      from dd03l
      where rollname = dd03l-rollname and as4local = 'A'.
    delete lt_03l
      where tabname(4) <> 'BAPI' or shlporigin = ' ' or
        shlporigin = 'T'.
  else.
    select * into table lt_03l
      from dd03l
      where tabname = w_dtel-tabname and
        rollname = dd03l-rollname and
        as4local = 'A'.
  endif.
  *-----
  loop at lt_03l into lw_03l.
    perform find_usage_in_bo
      using lw_03l-tabname lw_03l-fieldname.
    if not swotlq-lobjtype is initial.
      searchhelps-issimple = 'X'.
      searchhelps-objtype = swotlq-lobjtype.
      searchhelps-method = swotlq-verb.
      searchhelps-param = swotlq-element.
      if swotlq-reffield is initial.
        searchhelps-field = lw_03l-fieldname.
      endif.
      perform add_to_buffer using searchhelps.
      return.
    endif.
  endloop.

```

```

endform.                                " find_by_data_element
*&-----*
*&      Form  add_to_buffer
*&-----*
form add_to_buffer using searchhelps type zbapishmeta.
  zshelpbor-shlptype = searchhelps-shlptype.
  zshelpbor-objtype = searchhelps-objtype.
  zshelpbor-method = searchhelps-method.
  zshelpbor-param = searchhelps-param.
  zshelpbor-field = searchhelps-field.

describe table t_shlp lines sy-tfill.
if sy-tfill > 1.
  loop at t_shlp into w_shlp.
    zshelpbor-shlpname = w_shlp-shlpname.
    zshelpbor-issimple = w_shlp-issimple.
    insert zshelpbor.
  endloop.
else.
  zshelpbor-shlpname = searchhelps-shlpname.
  zshelpbor-issimple = searchhelps-issimple.
  insert zshelpbor.
endif.
v_index_changed = 'X'.
endform.                                " add_to_buffer
*&-----*
*&      Form  get_meta_fv
*&-----*
form get_meta_fv changing searchhelps type zbapishmeta.
  "try buffer
  select single * from zshelpbor
    where shlpname = searchhelps-shlpname and shlptype = 'FV'.
  if sy-subrc = 0.
    searchhelps-issimple = zshelpbor-issimple.
    searchhelps-objtype = zshelpbor-objtype.
    searchhelps-method = zshelpbor-method.
    searchhelps-param = zshelpbor-param.
    searchhelps-field = zshelpbor-field.
    return.
  endif.
data: lw_031 type ty_dd031, lt_031 type table of ty_dd031.
select single domname into dd031-domname
  from dd011
  where domname = searchhelps-shlpname and as4local = 'A'.
if sy-subrc <> 0.
  error_fill 'ZZDDIC' '000' 'Invalid domain name:'
    searchhelps-shlpname ' ' '.
  return.
endif.
select tabname fieldname
  into corresponding fields of table lt_031
  from dd031
  where domname = dd031-domname and as4local = 'A' and
    shlporigin = 'F'.
delete lt_031 where tabname(4) <> 'BAPI'.
loop at lt_031 into lw_031.
  perform find_usage_in_bo
    using lw_031-tabname lw_031-fieldname.
  if not swotlq-lobjtype is initial.

```

```

searchhelps-issimple = 'X'.
searchhelps-shlptype = 'FV'.
searchhelps-shlpname = dd031-domname.
searchhelps-objtype = swotlq-lobjtype.
searchhelps-method = swotlq-verb.
searchhelps-param = swotlq-element.
if swotlq-reffield is initial.
    searchhelps-field = lw_031-fieldname.
endif.
perform add_to_buffer using searchhelps.
exit. "LOOP
endif.
endloop.
endform.                                " get_meta_fv
*&-----*
*&      Form  CHECK_AUTH
*&-----*
*      text
*-----*
*      -->P_SEARCHHELPS  text
*-----*
form check_auth changing searchhelps type zbapishmeta ret type bapireturn.
* also using SWOTLQ, DD35L-TABNAME and DD03L-ROLLNAME
clear ret.
if searchhelps-issimple = 'X'.
    perform authoritychecks(saplbfhv)
        using swotlq-lobjtype ' ' swotlq-verb dd35l-tabname
            dd03l-rollname
        changing shlp_method ret.
else.
    data: l_shlp_method type shlp_descr_t,
          lt_shlp like table of w_shlp.
    select dd31s~subshlp dd30l~issimple into table lt_shlp
        from dd31s join dd30l on dd30l~shlpname = dd31s~subshlp and
            dd30l~as4local = dd31s~as4local
        where dd31s~shlpname = searchhelps-shlpname and
            dd31s~as4local = 'A' and issimple = 'X'.
    loop at lt_shlp into w_shlp.
        call function 'F4IF_GET_SHLP_DESCR'
            exporting
                shlpname = w_shlp-shlpname
                shlptype = 'SH'
            importing
                shlp      = l_shlp_method.
        perform authoritychecks(saplbfhv)
            using swotlq-lobjtype ' ' swotlq-verb dd35l-tabname
                dd03l-rollname
            changing l_shlp_method ret.
        exit.
    endloop.
    if sy-subrc <> 0.
        ret-type = 'E'.
    endif.
endif.
endform.                                " CHECK_AUTH
*INCLUDE LZZDDICTOP

function-pool zzddic message-id zzddic.
type-pools shlp.

```

```

types: begin of ty_shlps,
      shlptype(2),
      shlpname type dd301-shlpname,
      objtype type swotlq-lobjtype,
      method type swotlq-verb,
      parm type swotlq-element,
      field type dd031-fieldname,
end of ty_shlps.
types: begin of ty_dd031,
      tabname type dd031-tabname,
      fieldname type dd031-fieldname,
      shlporigin type dd031-shlporigin,
end of ty_dd031.

tables:
  dd021, "SAP Tables
  dd031, "Table Fields
  dd301, "Search helps
  dd351, "Search help attachments to structures: Headers
  swotlq, "Runtime Table Parameters, Exceptions
  zshelpbor. "Assignment of search helps to BOR parameters

data: shlp_method type shlp_descr_t.
data: v_index_changed type c.

data: begin of w_dtel,
      rollname type dd031-rollname,
      tabname type dd031-tabname,
end of w_dtel,
t_dtel like hashed table of w_dtel with unique key rollname.
data: begin of w_shlp,
      shlpname type dd301-shlpname,
      issimple type dd301-issimple,
end of w_shlp,
t_shlp like table of w_shlp.

define error_fill.
  searchhelps-type = 'E'.
  searchhelps-id = &1.
  searchhelps-number = &2.
  searchhelps-message_v1 = &3.
  searchhelps-message_v2 = &4.
  searchhelps-message_v3 = &5.
  searchhelps-message_v4 = &6.
  message id &1 type 'S' number &2 with &3 &4 &5 &6 into
  searchhelps-message.
end-of-definition.

```

Copyright

© 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

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

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