How-To Guide: Dropdown Boxes

SAP® CRM 7.0

Target Audience

- System administrators
- Technology consultants

Document version: 1.4 – December 2008

Terms for Included Open Source Software

This SAP software contains also the third party open source software products listed below. Please note that for these third party products the following special terms and conditions shall apply.

1. This software was developed using ANTLR.

2. gSOAP

Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3. SAP License Agreement for STLport

SAP License Agreement for STLPort between

SAP Aktiengesellschaft

Systems, Applications, Products in Data Processing

Neurottstrasse 16

69190 Walldorf, Germany

(hereinafter: SAP)

and

you

(hereinafter: Customer)

a) Subject Matter of the Agreement

A) SAP grants Customer a non-exclusive, non-transferable, royalty-free license to use the STLport.org C++ library (STLport) and its documentation without fee.

B) By downloading, using, or copying STLport or any portion thereof Customer agrees to abide by the intellectual property laws, and to all of the terms and conditions of this Agreement.

C) The Customer may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrictions.

D) Customer shall maintain the following copyright and permissions notices on STLport sources and its documentation unchanged:

Copyright 2001 SAP AG

E) The Customer may distribute original or modified STLport sources, provided that:

o The conditions indicated in the above permissions notice are met;

o The following copyright notices are retained when present, and conditions provided in accompanying permission notices are met:

Copyright 1994 Hewlett-Packard Company

Copyright 1996,97 Silicon Graphics Computer Systems Inc.

Copyright 1997 Moscow Center for SPARC Technology.

Copyright 1999,2000 Boris Fomitchev

Copyright 2001 SAP AG

Permission to use, copy, modify, distribute and sell this software and its documentation for any purposes is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purposes is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Moscow Center for SPARC makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Boris Fomitchev makes no representations about the suitability of this software for any purpose. This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk. Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purposes is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. SAP makes no representations about the suitability of this software for any purpose. It is provided with a limited warranty and liability as set forth in the License Agreement distributed with this copy. SAP offers this liability and warranty obligations only towards its customers and only referring to its modifications.

b) Support and Maintenance

SAP does not provide software maintenance for the STLport. Software maintenance of the STLport therefore shall be not included.

All other services shall be charged according to the rates for services quoted in the SAP List of Prices and Conditions and shall be subject to a separate contract.

c) Exclusion of warranty

As the STLport is transferred to the Customer on a loan basis and free of charge, SAP cannot guarantee that the STLport is error-free, without material defects or suitable for a specific application under third-party rights. Technical data, sales brochures, advertising text and quality descriptions produced by SAP do not indicate any assurance of particular attributes.

d) Limited Liability

A) Irrespective of the legal reasons, SAP shall only be liable for damage, including unauthorized operation, if this (i) can be compensated under the Product Liability Act or (ii) if caused due to gross negligence or intent by SAP or (iii) if based on the failure of a guaranteed attribute.

B) If SAP is liable for gross negligence or intent caused by employees who are neither agents or managerial employees of SAP, the total liability for such damage and a maximum limit on the scope of any such damage shall depend on the extent to which its occurrence ought to have anticipated by SAP when concluding the contract, due to the circumstances known to it at that point in time representing a typical transfer of the software.

C) In the case of Art. 4.2 above, SAP shall not be liable for indirect damage, consequential damage caused by a defect or lost profit.

D) SAP and the Customer agree that the typical foreseeable extent of damage shall under no circumstances exceed EUR 5,000.

E) The Customer shall take adequate measures for the protection of data and programs, in particular by making backup copies at the minimum intervals recommended by SAP. SAP shall not be liable for the loss of data and its recovery, notwithstanding the other limitations of the present Art. 4 if this loss could have been avoided by observing this obligation.

F) The exclusion or the limitation of claims in accordance with the present Art. 4 includes claims against employees or agents of SAP.

4. Adobe Document Services

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and / or other countries. For information on Third Party software delivered with Adobe document services and Adobe LiveCycle Designer, see SAP Note 854621.

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.
	Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Meaning
⚠	Caution
⚙	Example
💡	Note
⬆	Recommendation
⟨⟩	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Contents

1 Introduction

A common task for a CRM developer is providing a dropdown box for a business object attribute displayed in a view on the CRM WebClient UI.

1.1 Solution Summary

To provide a dropdown box, you implement a value help getter (V getter) for your attribute in the context node class used for display. For example, GET_V_PARTNER_TYPE for PARTNER_TYPE.

The V getter returns an object by implementing the interface IF_BSP_WD_VALUEHELP_PLDESCR. It provides the dropdown box content by using one of the following methods:

- GET_SELECTION_TABLE

    This method returns the content as table data.
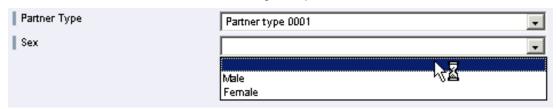
- GET_BINDING_STRING

    This method points to a context node attribute that provides the content.

This document includes examples for both approaches.

For this How-To Guide, the result of creating a dropdown box looks like this:

For the advanced search, the procedure is different. For more information, see *Provide a Dropdown Box for an Advanced Search* [page 17].

—

2 Detailed Procedure for Providing a Dropdown Box

2.1 Create P Getter for Business Object Attribute

For the UI to display a dropdown box, the field type of the attribute must be set as a dropdown box.

The attribute's context node class must have a property getter (P getter) method implemented. To create a P getter for the attribute, do the following:

1. Open the component workbench (transaction BSP_WD_CMPWB) and navigate to the component view containing the appropriate attribute.

2. Select the *Implementation Class* below the context node used for display to open the Class Builder.

3. Use the method _GET_P_XYZ as a template to create a P getter for your attributes by right clicking on it and choosing *Copy* from the context menu.

4. Rename the copied P getter to reflect the appropriate attribute (for example, GET_P_PARTNER_TYPE).

5. In *Implementation*, define the field type as follows:

   ‹› Syntax

```
method GET_P_PARTNER_TYPE.

    CASE iv_property.
            WHEN IF_BSP_WD_MODEL_SETTER_GETTER=>FP_FIELDTYPE.
*   field type: picklist
            RV_VALUE =
CL_BSP_DLC_VIEW_DESCRIPTOR=>FIELD_TYPE_PICKLIST.

    ENDCASE.

endmethod.
```

2.2 Create V Getter for Business Object Attribute

Create a value help getter (V getter) for your attribute by doing the following:

1. Open the component workbench (transaction BSP_WD_CMPWB) and navigate to the component view containing the appropriate attribute.

2. Select the *Implementation Class* below the context node used for display to open the Class Builder.

3. Copy the _GET_V_XYZ method and rename it to reflect the appropriate attribute (for example, _GET_V_PARTNER_TYPE). A similar process is described in *Create P Getter for Business Object Attribute* [page 9].

4. Once you have created the V getter, refresh the configuration so that the picklists are detected properly in the configuration tab.

2.3 Prepare the Value Help Descriptor

To prepare the business object, implement the IF_BSP_WD_VALUEHELP_PLDESCR interface.

It is a good idea to implement this object in the form of a local class in the corresponding context node as described below. Use a global class if you need the value help in more than one place.

2.3.1 Option A: Return Dropdown Content with a Table

In the following example, you prepare a value help for the attribute SEX based on the domain BU_SEX and its built-in attribute list. After setting the parameters for the class, you can use it to retrieve the values for any domain.

1. Select *Local Types* in the Class Builder to define the local class.

    The class definition is as follows:

    ‹› Syntax

    ```
    class LCL_VALUEHELP_FOR_DOMAIN definition friends
    CL_ZCRMCMP__TESTVIEW_CN01.

      public section

        interfaces: IF_BSP_WD_VALUEHELP_PLDESCR.

        methods: CONSTRUCTOR

                   importing

                     IV_SOURCE_TYPE type CHAR1.

      private section.

        data: GT_SELECTION_TABLE

                type BSP_WD_DROPDOWN_TABLE.

        methods: LOAD_SELECTION_TABLE

                  importing

                    IV_DOMAIN_NAME type STRING.

    endclass.
    ```

2. Select *Implementation* to implement the IF_BSP_WD_VALUEHELP_PLDESCR interface:

    ‹› Syntax

    ```
    *        CLASS lcl_valuehelp_for_domain IMPLEMENTATION

    *        Local class providing a value help for a domain value


    class LCL_VALUEHELP_FOR_DOMAIN implementation.

      method CONSTRUCTOR.

    *    Remember if content of dropdown listbox comes from table or
    context node

    *     ... SOURCE_TYPE_TABLE or SOURCE_TYPE_BINDING

        ME->IF_BSP_WD_VALUEHELP_PLDESCR~SOURCE_TYPE = IV_SOURCE_TYPE.

      endmethod.                    "CONSTRUCTOR

      method IF_BSP_WD_VALUEHELP_PLDESCR~GET_SELECTION_TABLE.
    ```

```
*    Returns cached selection table
*     ... called in case table content comes from table
(SOURCE_TYPE_TABLE)
     RT_RESULT = GT_SELECTION_TABLE.
  endmethod.
"

  method IF_BSP_WD_VALUEHELP_PLDESCR~GET_BINDING_STRING.
*    Value help does not come from context node here.
     raise exception type CX_BSP_WD_INCORRECT_IMPLEMENT.
  endmethod.
"

  method LOAD_SELECTION_TABLE.
*    Retrieves content of dropdown  listbox
*     ... called in case table content comes from table
(SOURCE_TYPE_TABLE)


*    Get content for selection table from database
     data: LV_DOMAIN_NAME type DOMNAME.
     LV_DOMAIN_NAME = IV_DOMAIN_NAME.
     data: LT_DOMAIN_ENTRIES type standard table of DD07V.
     call function 'DD_DOMVALUES_GET'
       exporting
         DOMNAME    = LV_DOMAIN_NAME
         LANGU      = SY-LANGU
         TEXT       = 'X'
       tables
         DD07V_TAB = LT_DOMAIN_ENTRIES.
     if SY-SUBRC <> 0.
       message E009(DMCLG).
     endif.


*    Load and remember selection table
     data: LS_SELECTION_TABLE       type line of
BSP_WD_DROPDOWN_TABLE,
           LT_SELECTION_TABLE       type BSP_WD_DROPDOWN_TABLE.
     field-symbols: <DOMAIN_ENTRY> type DD07V.
     loop at LT_DOMAIN_ENTRIES assigning <DOMAIN_ENTRY>.
       LS_SELECTION_TABLE-KEY = <DOMAIN_ENTRY>-VALPOS.
       LS_SELECTION_TABLE-VALUE = <DOMAIN_ENTRY>-DDTEXT.
       append LS_SELECTION_TABLE to LT_SELECTION_TABLE.
```
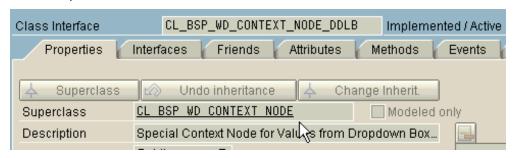
```
      endloop.
      GT_SELECTION_TABLE = LT_SELECTION_TABLE.


  endmethod.                          "SET_SELECTION_TABLE


endclass.
```

2.3.2 Option B: Return a Binding String

Alternatively, you can implement method GET_BINDING_STRING of your value help descriptor class pointing to a context node (of type CL_BSP_WD_CONTEXT_NODE_DDLB) that holds the content of the dropdown box.

The example below uses the *PARTNERTYPES* context node:

1. Select *Local Types* in the Class Builder to define the local class.

    The class definition is as follows:

    ‹› Syntax

    ```
    * Local class providing a context node based value help *

    class LCL_VALUEHELP_WITH_BINDING definition friends
    CL_ZCRMCMP__TESTVIEW_CN01.
      public section.
        interfaces: IF_BSP_WD_VALUEHELP_PLDESCR.
        methods: CONSTRUCTOR
                   importing
                     IV_SOURCE_TYPE type CHAR1.
      private section.
        data: GV_BINDING_STRING
                 type STRING.
        methods: SET_BINDING_STRING
                   importing
                     IV_BINDING_STRING type STRING.
    endclass.
    ```

2. Select *Implementation* to implement the IF_BSP_WD_VALUEHELP_PLDESCR interface:

◆❯ Syntax

```
*        CLASS lcl_partner_type_values IMPLEMENTATION

*        Local class providing a value help taken from a context
node

class LCL_VALUEHELP_WITH_BINDING implementation.


  method CONSTRUCTOR.

*    Remember if content of dropdown listbox comes from table or
context node

*     ... SOURCE_TYPE_TABLE or SOURCE_TYPE_BINDING

     ME->IF_BSP_WD_VALUEHELP_PLDESCR~SOURCE_TYPE = IV_SOURCE_TYPE.

  endmethod.                        "CONSTRUCTOR

  method IF_BSP_WD_VALUEHELP_PLDESCR~GET_SELECTION_TABLE.

*    Content for value help does not come from table here

     raise exception type CX_BSP_WD_INCORRECT_IMPLEMENT.

  endmethod.
"

  method IF_BSP_WD_VALUEHELP_PLDESCR~GET_BINDING_STRING.

*    Returns binding string

*     ... called in case table content comes from context node
(SOURCE_TYPE_BINDING)

     RV_RESULT = GV_BINDING_STRING.

  endmethod.
"

  method SET_BINDING_STRING.

*    Remember

     GV_BINDING_STRING = IV_BINDING_STRING.

  endmethod.                       "SET_SELECTION_TABLE


  endclass.
```
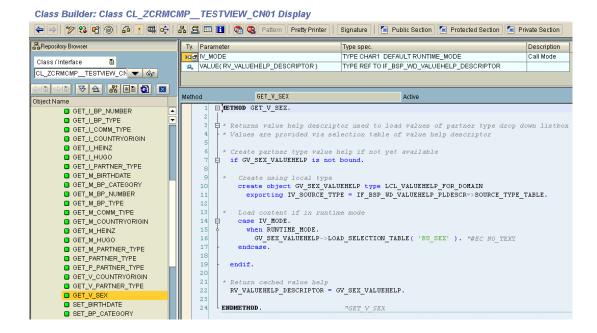
2.4 Return the Value Help Descriptor

2.4.1 Option A: Return Dropdown Content with a Table

To return the dropdown content as table data, implement the V getter to return the value help descriptor:

In this example, the descriptor is cached in the GV_VALUE_HELP attribute and is created only if it does not yet exist there. If the value help is operated in runtime mode, the selection table is loaded for the domain name (BU_SEX here). In design time mode there is no need to load the table content.

GV_VALUE_HELP is defined in the *Attributes* tab as GV_SEX_VALUEHELP, Private, with *Type Ref To* LCL_VALUEHELP_FOR_DOMAIN.

The code for the V getters for the SEX and PARTNER_TYPES attributes are as follows:

◇ Syntax

```
METHOD GET_V_SEX.

* Returns value help descriptor used to load values of partner
type dropdown  listbox

* Values are provided via selection table of value help
descriptor

* Create partner type value help if not yet available

  if GV_SEX_VALUEHELP is not bound.

*   Create using local type

    create object GV_SEX_VALUEHELP type LCL_VALUEHELP_FOR_DOMAIN

      exporting IV_SOURCE_TYPE =
IF_BSP_WD_VALUEHELP_PLDESCR=>SOURCE_TYPE_TABLE.

*   Load content if in runtime mode

    case IV_MODE.

      when RUNTIME_MODE.

        GV_SEX_VALUEHELP->LOAD_SELECTION_TABLE( 'BU_SEX' ). "#EC
NO_TEXT

    endcase.
```

```
      endif.


   * Return cached value help

      RV_VALUEHELP_DESCRIPTOR = GV_SEX_VALUEHELP.

ENDMETHOD.                         "GET_V_SEX


   method GET_V_PARTNER_TYPE.


   * Returns value help descriptor used to load values of partner
   type dropdown  listbox

   * Values are provided via context node

   * Create using local type

     data: LV_PARTNER_TYPE_VALUEHELP type ref to
   LCL_VALUEHELP_WITH_BINDING.

     create object LV_PARTNER_TYPE_VALUEHELP

       exporting IV_SOURCE_TYPE =
   IF_BSP_WD_VALUEHELP_PLDESCR=>SOURCE_TYPE_BINDING.


   *   Load content if in runtime mode

     case IV_MODE.

       when RUNTIME_MODE.

         LV_PARTNER_TYPE_VALUEHELP->SET_BINDING_STRING(
   '//PARTNERTYPES/VALUES' )."#EC NO_TEXT

     endcase.


     RV_VALUEHELP_DESCRIPTOR = LV_PARTNER_TYPE_VALUEHELP.

   endmethod.
```

You can keep your value help descriptors in a public location that serves as a cache.

2.4.2 Option B: Return a Binding String

The V getter for the partner type operates the value help descriptor in SOURCE_TYPE_BINDING mode and transfers the binding string to the value help descriptor.

You can create the corresponding context node PARTNERTYPES, which holds the content of the dropdown box, with the built-in wizard of the component workbench.

1. Select *Context Node* under *Context*, and choose *Create* from the context menu.

2. Select *Context Node for Dropdown Box* as the context node type, so that class CL_BSP_WD_CONTEXT_NODE_DDLB is used.

3. You can use the DO_INIT_CONTEXT method of the corresponding controller implementation (CL_ZCRMCMP_TESTVIEW_IMPL in this case) to initialize the

dropdown content using method SET_VALUE_TABLE of the dropdown box context node:

<> Syntax

```
method DO_INIT_CONTEXT.
  data:
      QS type ref to CL_CRM_COL_QUERY_SERVICE,
      COL1 type ref to IF_BOL_BO_COL.


  QS = CL_CRM_BOL_QUERY_SERVICE=>GET_INSTANCE( 'BuilHeaderSearch' ).
  QS->SET_PROPERTY( IV_ATTR_NAME = 'PARTNER' IV_VALUE = '*' ).
  QS->SET_PROPERTY( IV_ATTR_NAME = 'MAX_HIT' IV_VALUE = '1' ).


  COL1 = QS->GET_QUERY_RESULT( ).


  ME->TYPED_CONTEXT->BP->COLLECTION_WRAPPER->SET_COLLECTION( COL1 ).


* Load content for partner type drop down listbox
* ... Select partner types from database
  data: LT_PARTNER_TYPES        type standard table of TB004T,
        LS_PARTNER_TYPE         type line of BSP_WD_DROPDOWN_TABLE,
        LT_PARTNER_TYPE_VALUES type BSP_WD_DROPDOWN_TABLE.
  field-symbols: <PARTNER_TYPE> type TB004T.
  select * from TB004T into table LT_PARTNER_TYPES
           where SPRAS eq SY-LANGU.
  loop at LT_PARTNER_TYPES assigning <PARTNER_TYPE>.
      LS_PARTNER_TYPE-KEY = <PARTNER_TYPE>-BPKIND.
      LS_PARTNER_TYPE-VALUE = <PARTNER_TYPE>-TEXT40.
      append LS_PARTNER_TYPE to LT_PARTNER_TYPE_VALUES.
  endloop.
* ... Pass partner type values to context node for drop down listbox
  ME->TYPED_CONTEXT->PARTNERTYPES->SET_VALUE_TABLE(
LT_PARTNER_TYPE_VALUES ).
endmethod.
```

You can define your dropdown box context nodes in a custom controller context, so that you can reuse the cached content in various views with binding.

2.5 Trigger a Server Roundtrip After a Value is Selected

It is possible to trigger a server roundtrip when the dropdown box has been used. To do this, enhance the P getter of the attribute:

1. In *Implementation*, define the event (here PARTNER_TYPE_SELECTED) as follows:

◇ Syntax

```
method GET_P_PARTNER_TYPE.

   CASE iv_property.
…

…

…

      WHEN IF_BSP_WD_MODEL_SETTER_GETTER=>FP_SERVER_EVENT.
         rv_value = 'partner_type_selected'.

endmethod.
```

2. You can influence the event PARTNER_TYPE_SELECTED by implementing corresponding reactions in the controller's DO_HANDLE_EVENT method:

◇ Syntax

```
method DO_HANDLE_EVENT.


* Event handler dispatching


case HTMLB_EVENT_EX->EVENT_SERVER_NAME.


   when 'select'.                             *#EC NOTEXT
      EH_ONSELECT(
               HTMLB_EVENT    = HTMLB_EVENT
               HTMLB_EVENT_EX = HTMLB_EVENT_EX ).
* Partner type has been selected in drop down list box
   when 'partner_type_selected'.             *#EC NOTEXT
      EH_ON_PARTNER_TYPE_SELECTED(
               HTMLB_EVENT    = HTMLB_EVENT
               HTMLB_EVENT_EX = HTMLB_EVENT_EX ).


   when others.
   endcase.
endmethod.
```

For more information about P getters, see *Create P Getter for Business Object Attribute* [page 9].

2.6 Provide a Dropdown Box for an Advanced Search

For the advanced search you do not have to implement a V getter. Instead, you must redefine method GET_DQUERY_DEFINITIONS in the implementation class of the view controller. You can fill in table RT_RESULT with information relating to the value help (see structure CRMS_THTMLB_SEARCH_FIELD_INFO and attribute DDLB_OPTIONS).

The following is an example of the code that you can enter in the implementation class of the view controller:

〈〉 Syntax

```
method GET_DQUERY_DEFINITIONS.


  data:
   ls_ddlb type CRMS_THTMLB_SEARCH_DDLB_NVP.


  field-symbols: <rt_result> type CRMS_THTMLB_SEARCH_FIELD_INFO.


  CALL METHOD SUPER->GET_DQUERY_DEFINITIONS
    RECEIVING
      RT_RESULT = rt_result.


* DDLB for field "DESCRIPTION"
  read table rt_result with key field = 'DESCRIPTION' assigning
<rt_result>.
  if sy-subrc = 0.
    ls_ddlb-key   = 'key1'.
    ls_ddlb-value = 'value1'.
    append ls_ddlb to <rt_result>-DDLB_OPTIONS.
    ls_ddlb-key   = 'key2'.
    ls_ddlb-value = 'value2'.
    append ls_ddlb to <rt_result>-DDLB_OPTIONS.
  endif.


  endmethod.
```