Mandatory Field Check in Web Dynpro- ABAP

Applies to:
SAP ECC 6.0 Onwards. For more information, visit the Web Dynpro ABAP homepage.

Summary
Efficient way to Apply Mandatory/Obligatory field Check on Input fields in a View.

Author: Akash Rana
Company: Deloitte Touche Tohmatsu
Created on: 7 April 2011

Author Bio
Akash Rana is a SAP ABAP Technical Consultant with expertise in development across multiple modules. He is experienced in developing Web dynpro based applications, UI, forms, ALE / IDOCS and data transfer across multiple modules.
Table of Contents

Introduction ................................................................................................................................................. 3

Application Overview .................................................................................................................................. 3

Step by Step Process .................................................................................................................................... 3
  Step 1: Create a Web Dynpro Component. .............................................................................................. 3
  Step 2: Create Context for View V_MAIN. ............................................................................................... 3
  Step 3: Design Layout. .......................................................................................................................... 5
  Step 4: Implement Hook Method WDDOBEFOREACTION(). ................................................................. 6
  Step 5: Implement Event Handler Method, Activate and Create an Application. ................................. 7

Application .................................................................................................................................................. 8
  Mandatory Field Check ....................................................................................................................... 8

Related Content ......................................................................................................................................... 9

Disclaimer and Liability Notice ................................................................................................................ 10
Introduction

Sometimes we have a requirement to except entries on a view and perform a mandatory check on them. If all of the fields for mandatory check are filled, we need to process the data accordingly, if all fields are not filled then we need to issue a message and cancel the further processing.

There are many ways to perform the above task, here I will share a very efficient way to perform mandatory field check, without checking mandatory fields in related action handler methods manually.

Application Overview

Our application will accept Batch, Material and Plant (Mandatory fields in view) and display corresponding Sales Document's and its Item details - Sales Document, Item No., Net value and Document Currency in a table.

Input Fields are Batch, Material and Plant, which are mandatory fields.

We would apply mandatory check on these fields, if any of the above fields is empty, error message would be returned on the view and further processing would be canceled.

Step by Step Process

Step 1: Create a Web Dynpro Component.

Go to SE80 and create a WD Component.

Step 2: Create Context for View V_MAIN.

In the view context create node-

**ND_FILTER (Node attributes will be used as Input Fields on View V_MAIN)**

Attributes- CHARG (Type VBAP-CHARG), MATNR (Type VBAP-MATNR), WERKS (Type VBAP-WERKS).

Cardinality- 1..1
### ND_SALES_DOC (Node will be Attached to Data Source of UI Table in view V_MAIN)

Attributes: VBELN (Type VBAP-VBELN), POSNR (Type VBAP-POSNR), NETWR (Type VBAP-NETWR), WAERK (Type VBAP-WAERK)

Cardinality: 0..n

---

**Property** | **Value**
---|---
**Nodes**
Node Name | ND_SALES_DOC
Dictionary structure
Cardinality | 0..n
Selection | 0..1
Initialization Lead Selection | ✔
Singleton | ☐
Supply Function |
Step 3: Design Layout.

Create Input Fields using Node ND_FILTER for attributes - CHARG (Batch No.), MATNR (Material), WERKS (Plant) and set the value for the STATE Property of Input Fields as "required".

This would result a red asterisk in front of input fields, to indicate that it's a mandatory entry field. This is only for display option, we need to explicitly implement this check.

Create a Button and Action- GET DETAILS to fetch Sales Document details based on entries in input fields.

Create a Table UI with Data Source linked to context node ND_SALES_DOC and columns as Sales Document, Item No., Net value and Document Currency.
Step 4: Implement Hook Method WDDOBEFOREACTION().

Hook method WDDOBEFOREACTION is triggered before any Event Handler associated with an Action.

We need to check whether the user has entered all the mandatory fields once he triggers Event Handler Method ONACTIONGET_DETAILS associated with Button Submit.

Hook method WDDOBEFOREACTION would be triggered before Event Handler Method ONACTIONGET_DETAILS, here we can check for the mandatory field check using class CL_WD_DYNAMIC_TOOL, following is the implementation of wddobeforeaction() hook method.

```
METHOD wddobeforeaction.

  * Reference to View API
  DATA lo_api_controller TYPE REF TO if_wd_view_controller.

  * Reference to Action
  DATA lo_action TYPE REF TO if_wd_action.

  * Get API of View Controller
  lo_api_controller = wd_this->wd_get_api().

  * Get Current Action Called.
  lo_action = lo_api_controller->get_current_action().

  * Check Action
  IF lo_action IS BOUND.
      CASE lo_action->name.
          WHEN 'GETDETAILS'.
              CALL METHOD cl_wd_dynamic_tool->check_mandatory_attr_on_view
              EXPORTING
                  view_controller = lo_api_controller. " Export current view API

              END_CASE.
      END_CASE.
  ENDIF.
ENDMETHOD.
```

Implementation of Hook Method- WDDOBEFOREACTION-

METHOD wddobeforeaction.

  * Reference to View API
  DATA lo_api_controller TYPE REF TO if_wd_view_controller.

  * Reference to Action
  DATA lo_action TYPE REF TO if_wd_action.

  * Get API of View Controller
  lo_api_controller = wd_this->wd_get_api().

  * Get Current Action Called.
  lo_action = lo_api_controller->get_current_action().

  * Check Action
IF lo_action IS Bound.
  CASE lo_action->name.
   * Check of Action Get Details associated with button submit on View V_MAIN
     WHEN 'GET_DETAILS'.
     CALL METHOD cl_wd_dynamic_tool=>check_mandatory_attr_on_view
       EXPORTING
         view_controller = lo_api_controller. *Export current view API
   END_CASE.
  ENDIF.
ENDMETHOD.

Here we have used method CHECK_MANDATORY_ATTR_ON_VIEW of class CL_WD_DYNAMIC_TOOL to perform Mandatory check for input field UI elements with property STATE as 'required'.

Once the error message is raised Further processing of event handler method or navigation or wddomodifyview( ) method will not be executed as Error message raise by above method are error message with a context reference.

Step 5: Implement Event Handler Method, Activate and Create an Application.

<table>
<thead>
<tr>
<th>Application</th>
<th>zwd_mandatory_check</th>
<th>Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Mandatory Input Check</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>zwd_MANDATORY_CHECK</td>
<td></td>
</tr>
<tr>
<td>Interface View</td>
<td>wW_WIN</td>
<td></td>
</tr>
<tr>
<td>Plug Name</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>Help Menu Text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handling of Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Message Component on Demand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Always Display Message Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Administration Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Created By</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last changed by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>EN</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>http://***********/sap/bc/webdynpro/sap/zwd</td>
<td></td>
</tr>
</tbody>
</table>
**Application**

**Mandatory Field Check in Web Dynpro - ABAP**

**Filter**

- **Batch**: *
- **Material**: *
- **Plant**: *

Submit

**Mandatory Field’s**

Press Submit button without filling mandatory input field’s

**Sales Document’s**

<table>
<thead>
<tr>
<th>Sales Document</th>
<th>Item</th>
<th>Net value</th>
<th>Doc. Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Mandatory Field Check**

**Filter**

- **Batch**: *
- **Material**: *
- **Plant**: *

Submit

**Sales Document’s**

<table>
<thead>
<tr>
<th>Sales Document</th>
<th>Item</th>
<th>Net value</th>
<th>Doc. Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Related Content

Navigation in Web Dynpro ABAP
Using Dynamic ALV with Web Dynpro ABAP with Editable Fields
Assign Search Help to a Field at Runtime in Web Dynpro ABAP

For more information, visit the Web Dynpro ABAP 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.