

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

- Introduction3
- Application Overview3
- Step by Step Process3
 - Step 1: Create a Web Dynpro Component.3
 - Step 2: Create Context for View V_MAIN.3
 - Step 3: Design Layout5
 - Step 4: Implement Hook Method WDDOBEFOREACTION().6
 - Step 5: Implement Event Handler Method, Activate and Create an Application.7
- Application8
 - Mandatory Field Check8
- Related Content9
- Disclaimer and Liability Notice10

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.

Name	ZWD_MANDATORY_CHECK
Description	Mandatory Check on View
Type	<input checked="" type="radio"/> Web Dynpro Component <input type="radio"/> Web Dynpro Component Interface
Window Name	WN_WIN
View Name	V_MAIN

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

Context V_MAIN

- CONTEXT
 - ND_FILTER
 - CHARG
 - MATNR
 - WERKS

Property	Value
Nodes	
Node Name	ND_FILTER
Dictionary structure	
Cardinality	1..1
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input type="checkbox"/>
Supply Function	

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

Context V_MAIN

- CONTEXT
 - ND_FILTER
 - CHARG
 - MATNR
 - WERKS
 - ND_SALES_DOC
 - VBELN
 - POSNR
 - NETWR
 - WAERK

Property	Value
Nodes	
Node Name	ND_SALES_DOC
Dictionary structure	
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input type="checkbox"/>
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.

View Inactive(revised)

Properties | **Layout** | Inbound Plugs | Outbound Plugs | Context | Attributes | Actions | Methods

Favorites

text

action

selection

complex

layout

graphic

integration

Filter

Batch: *

Material: *

Plant: *

Sales Document's

Sales Document	Item	Net
V_MAIN.ND_SALES_DOC.VBELN	V_MAIN.ND_SALES_DOC.POSNR	V_
V_MAIN.ND_SALES_DOC.VBELN	V_MAIN.ND_SALES_DOC.POSNR	V_
V_MAIN.ND_SALES_DOC.VBELN	V_MAIN.ND_SALES_DOC.POSNR	V_
V_MAIN.ND_SALES_DOC.VBELN	V_MAIN.ND_SALES_DOC.POSNR	V_
V_MAIN.ND_SALES_DOC.VBELN	V_MAIN.ND_SALES_DOC.POSNR	V_

Row 1 of 5

CONTEXT_MENUS

ROOTUIELEMENTCONTAINER

GROUP_IO

CAPTION [Header]

GROUP_IO_CHARG_LBL

GROUP_IO_CHARG

GROUP_IO_MATNR_LBL

GROUP_IO_MATNR

GROUP_IO_WERKS_LBL

GROUP_IO_WERKS

BUTTON_DIS

GROUP_DETAILS

CAPTION_1 [Header]

TABLE_SL

Step 4: Implement Hook Method WDDOBEFOREACTION().

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

We need to check whether 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.

```

View      V_MAIN      Inactive(revised)
Properties  Layout  Inbound Plugs  Outbound Plugs  Context  Attributes  Actions  Methods
Method List  Method
Method      WDDOBEFOREACTION
.....
1  METHOD wddobeforeaction .
2
3  * Reference to View API
4  DATA lo_api_controller TYPE REF TO if_wd_view_controller.
5  * Reference to Action
6  DATA lo_action          TYPE REF TO if_wd_action.
7
8  * Get API of View Controller
9  lo_api_controller = wd_this->wd_get_api( ).
10
11 * Get Current Action Called.
12 lo_action = lo_api_controller->get_current_action( ).
13
14 * Check Action
15 IF lo_action IS BOUND.
16   CASE lo_action->name.
17    * Check of Action Get Details associated with button submit on View V_MAIN
18    WHEN 'GET_DETAILS'.
19     CALL METHOD cl_wd_dynamic_tool=>check_mandatory_attr_on_view
20     EXPORTING
21     view_controller = lo_api_controller. " Export current view API
22
23   ENDCASE.
24   ENDIF.
25 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

  ENDCASE.
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 wddommodifyview() 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.

Application Saved

Properties Parameters

Description	Mandatory Input Check		
Component	ZWD_MANDATORY_CHECK		
Interface View	WN_WIN		
Plug Name	DEFAULT		
Help Menu Text			
Help Link			

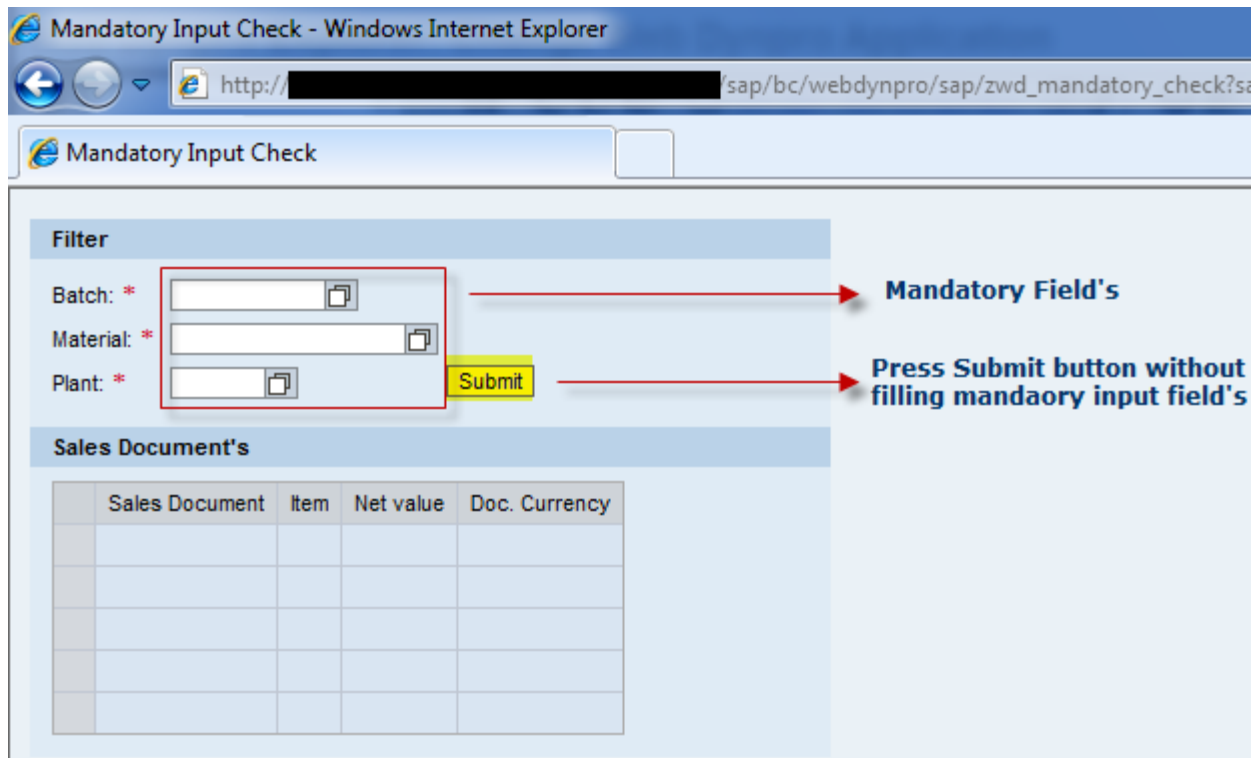
Handling of Messages

Show Message Component on Demand
 Always Display Message Component

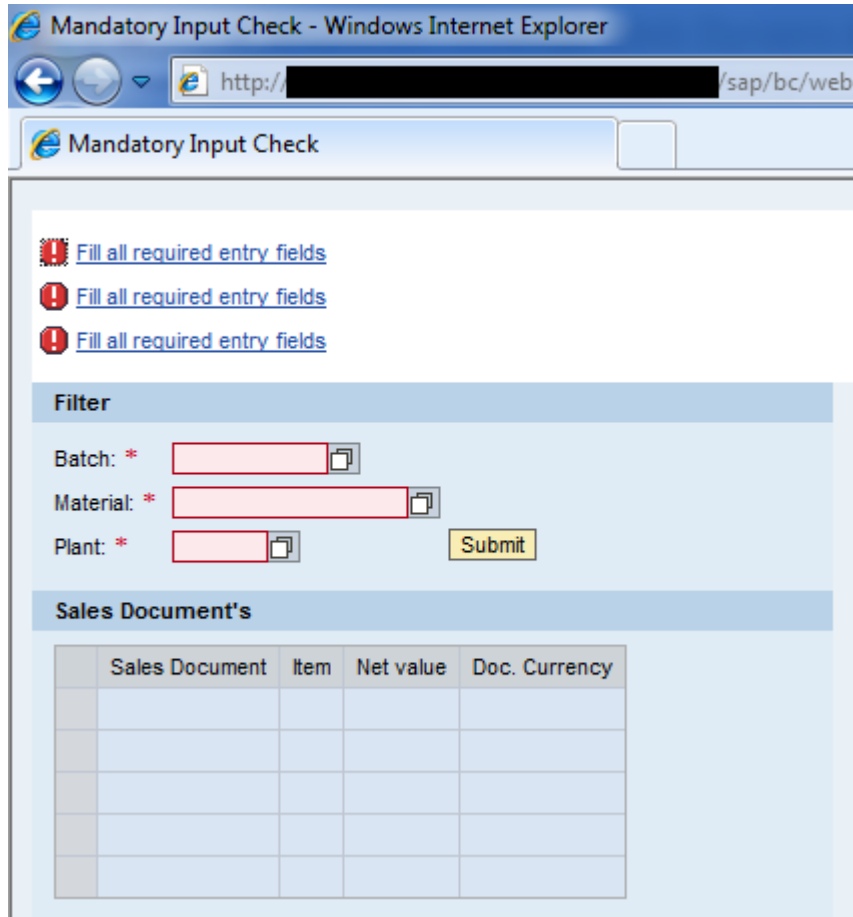
Administration Data

Created By	<input type="text"/>	Created on	06.04.2011
Last changed by	<input type="text"/>	Changed on	<input type="text"/>
Package	<input type="text"/>		
Language	EN		
URL	http:// <input type="text"/> /sap/bc/webdynpro/sap/zwd		

Application



Mandatory Field Check



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.