



How-to Guide  
SAP NetWeaver '04  
SAP Enterprise Portal 6.0

# How To... Configure the Universal Worklist

Version 1.00 – May 2004

Applicable Releases:  
SAP NetWeaver '04  
(SAP Enterprise Portal SP3, limited SP2 )

<p>© Copyright 2004 SAP AG. All rights reserved.</p> <p>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.</p> <p>Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.</p> <p>Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.</p> <p>IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.</p> <p>Oracle is a registered trademark of Oracle Corporation.</p> <p>UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.</p> <p>Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.</p> <p>HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.</p> <p>Java is a registered trademark of Sun Microsystems, Inc.</p> <p>JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.</p> <p>MaxDB is a trademark of MySQL AB, Sweden.</p>	<p>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.</p> <p>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.</p> <p>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. SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP</p>
---	---

---

	<p>NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.</p>
--	--

---

<b>1</b>	<b>SCENARIOS</b> .....	<b>1</b>
<b>2</b>	<b>RESULTS</b> .....	<b>2</b>
2.1	Minimal Configuration .....	2
2.2	Customize existing List Views .....	2
2.3	Item Launch Configuration.....	2
2.4	Custom Attributes .....	3
2.5	List Views with Custom Attributes.....	4
2.6	Decision Views .....	5
2.7	Tailored UWL iViews.....	5
<b>3</b>	<b>STEP BY STEP - MINIMAL CONFIGURATION</b> .....	<b>7</b>
3.1	Prerequisites.....	7
3.2	Creating the System in the System Landscape .....	7
3.3	Mapping the Portal User to the System User.....	9
3.4	Adding the UWL iView to the User Role .....	10
3.5	Registering the System with the UWL.....	12
3.6	Registering System Item Types with the UWL.....	14
3.7	Uploading Configuration Files .....	17
<b>4</b>	<b>STEP BY STEP – CUSTOMIZE EXISTING LIST VIEWS</b> .....	<b>19</b>
4.1	Get the definition of the View to be Customized .....	19
4.2	Create new configuration file .....	20
4.3	Copy the existing view definition to the new configuration file.....	20
4.4	Modify the view .....	21
4.4.1	Remove or add columns .....	21
4.4.2	Change column order .....	22
4.4.3	Change sorting.....	22
4.4.4	Change date format.....	23
4.5	Upload the config file .....	23

---

<b>5</b>	<b>STEP BY STEP – ITEM LAUNCH CONFIGURATION .....</b>	<b>25</b>
5.1	Inspecting Current Item Launch Configuration .....	25
5.2	Creating a New Configuration File.....	27
5.3	Adding Custom Launch Configuration for iViews .....	28
5.4	Overriding ITS MiniApp Launch Configuration.....	31
5.5	EP 5.0 Backward Compatibility: Converting Override XML File.....	32
<b>6</b>	<b>STEP BY STEP – CUSTOM ITEM ATTRIBUTES .....</b>	<b>34</b>
<b>7</b>	<b>STEP BY STEP – LIST VIEWS WITH CUSTOM ATTRIBUTES .....</b>	<b>37</b>
7.1	Prerequisites.....	37
7.2	Customization File.....	37
7.3	Adding List view with Custom Attributes.....	37
7.4	Upload the configuration file.....	40
7.5	Verify the Custom attributes List View.....	40
<b>8</b>	<b>STEP BY STEP – DECISION VIEWS .....</b>	<b>41</b>
8.1	Generic user decision .....	41
8.2	Terminating events.....	45
8.3	Work item completion with container update .....	48
<b>9</b>	<b>STEP BY STEP – TAILORED UWL IVIEW.....</b>	<b>52</b>
9.1	Prerequisites.....	52
9.2	Creating a new UWL IView .....	52
9.3	Adding UWL IView to Portal Users home page .....	52
9.4	Verify the Custom IView .....	53
9.5	Further tips on layout .....	53
<b>10</b>	<b>APPENDIX .....</b>	<b>54</b>
10.1	Configuration XML Tag Documentation.....	54
10.2	UWL Item Launch Configuration / UWL Actions.....	54

---

10.3	API for Registering UWL XML Configuration.....	54
10.4	Creating demo iView.....	55
10.5	Item visualization types in Business Workflow.....	57

# 1 Scenarios

This How-to guide describes how you configure and customize the Universal Worklist for Business Workflow scenarios:

1. **Minimal Configuration** – Use the UWL by generating required configuration from the backend systems. For this scenario, work items (tasks) are usually executed using the SAP GUI for HTML.
2. **Customize existing List Views** – Change standard UWL list views in the Portal installation: Remove or add columns, change column order, change sorting, and many other view parameters.
3. **Item Launch Configuration** – Typically, work items (tasks) are executed using the SAP GUI for HTML. Implement specific iViews (or other UI components like WebDynpro) for item completion to
  - Simplify task execution for the end user
  - Improve integration into the Enterprise Portal

For those tasks, your iViews are launched in place of the SAP GUI for HTML.

In the EP 5.0 UWL, the functionality was known as “override”.

The document also explains how to convert EP 5.0 UWL override xml files for use with the EP 6.0 UWL.

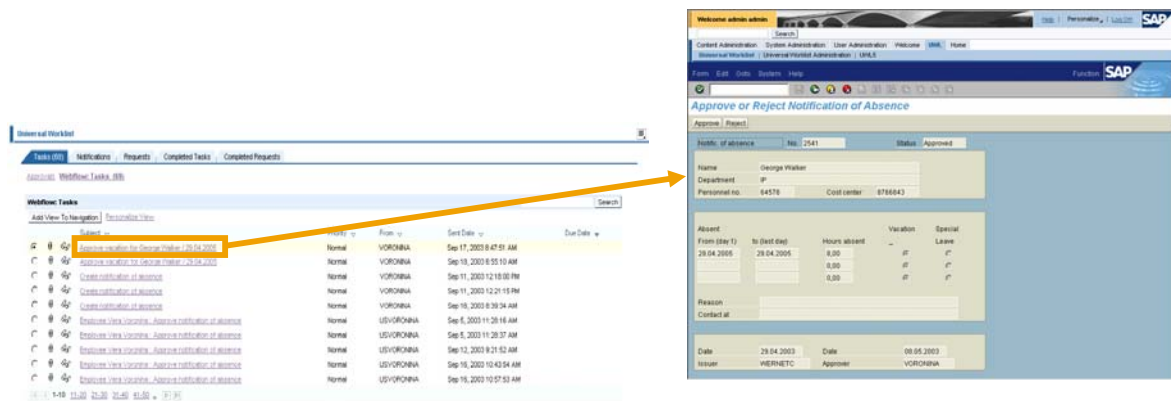
4. **Custom Attributes** – Enhance tasks items with additional attributes that are read from the leading Business Object of the task or from the work item container.
5. **List Views with Custom Attributes** – Display custom attributes in specific task list views.
6. **Decision Views** – Add decision actions to a list view to allow the user to efficiently complete multiple task items at once.
7. **Tailored UWL iView** - Create an iView, which contains only the work items and information you want the end user to see.

This guide refers to the UWL release that comes with EP 6.0 KMC SP3 or higher, but there are also specific notes for the SP2 release.

## 2 Results

### 2.1 Minimal Configuration

The Universal Worklist is operable within minutes. Work items are typically executed using the SAP GUI for HTML.



### 2.2 Customize existing List Views

Administrator can change the existing default list views and change several properties like the sort order, columns displayed

### 2.3 Item Launch Configuration

For specific task types, the default task launch as defined in the backend is overridden by iViews, WebDynpro applications or any URL, which represent the task completion transaction.



The image shows two screenshots from the SAP Universal Worklist. The left screenshot displays a list of tasks under the 'Workflow Tasks' section. A red 'X' is placed over the top right corner of this screenshot. A yellow arrow points from the first task in the list to the right screenshot. The right screenshot shows the 'Approve or Reject Notification of Absence' detail view for a specific task.

Task ID	Task Description	Priority	From	Sent Date	Due Date
1	Approve vacation for George Walker 7/20/2005	Normal	VORONNA	Sep 17, 2003 04:51 AM	
2	Approve vacation for George Walker 7/20/2005	Normal	VORONNA	Sep 18, 2003 6:55:10 AM	
3	Create notification of absence	Normal	VORONNA	Sep 11, 2003 12:18:00 PM	
4	Create notification of absence	Normal	VORONNA	Sep 11, 2003 12:21:15 PM	
5	Employee Vera Voronina - Approve notification of absence	Normal	VORONNA	Sep 16, 2003 8:39:34 AM	
6	Employee Vera Voronina - Approve notification of absence	Normal	USVORONNA	Sep 5, 2003 11:28:16 AM	
7	Employee Vera Voronina - Approve notification of absence	Normal	VORONNA	Sep 5, 2003 11:28:37 AM	
8	Employee Vera Voronina - Approve notification of absence	Normal	USVORONNA	Sep 12, 2003 9:21:52 AM	
9	Employee Vera Voronina - Approve notification of absence	Normal	USVORONNA	Sep 16, 2003 10:43:54 AM	
10	Employee Vera Voronina - Approve notification of absence	Normal	USVORONNA	Sep 16, 2003 10:57:53 AM	

The detail view on the right shows the following information:

- Notification of absence No: 0000002541
- Name: George Walker
- Department: #
- Personnel no.: 0000426
- Cost center: 0700043
- Table with columns: Absent From (Date), Absent To (Date), Hours absent, Vacation, Special Leave.
- Reason: #
- Contact #
- Date: 2003-04-29
- Date: 2003-05-08
- Issuer: HERNETIC
- Approver: VORONNA

## 2.4 Custom Attributes

The detail view shows additional attributes for the item.

**Item Detail**

[Launch SAP Task](#)
[New Ad Hoc Request](#)
[Reserve](#)
[View Detail in SAP Gui](#)
[View all of same category](#)

[priorityHigh](#)
[priorityLow](#)
[priorityNormal](#)
[priorityVeryHigh](#)

**Priority:** Normal  
**Status:** Created  
**Due Date:**  
**From:** USCARTMAN  
**Sent Date:** Feb 19, 2004 10:16 AM  
**Absence From:** Oct 10, 2003  
**Absence To:** Oct 10, 2003  
**Cost Center:** 7777  
**processor:** No Agent Available

**Subject:** [Employee CARTMAN: Approve notification of absence](#)

Employee CARTMAN submitted a notification of absence on 10.10.2003.  
 Make a decision about the request.  
 If you reject the request, create an attachment explaining your reason for the rejection. generous! generous!

**Attachment(s)**

[UWL Info Page 11/12/03 By CARTMAN](#)  
[UWLWorkItemViewExecution \(DOC 245KB\) 11/12/03 By CARTMAN](#)  
[Notif. of AbsenceCARTMAN of SP from 10.10.2003 to 10.10.2003](#)

**RawTextSample 11/12/03 By CARTMAN**

This is a sample of an raw text attachment, which was edited with the R/3 text editor.  
 This note describes the enhancements of Universal Worklist (UWL) for EP 6.0 SP2. It covers a major re-implementation with many new features.  
 XThe Universal Worklist enables SAP Portal end users to manage all their task items by bringing together tasks from multiple sources, including SAP R/3 Business Workflow and EP Ad Hoc Workflow applications. Ad Hoc Workflow allows users to create and structure tasks on the fly.

## 2.5 List Views with Custom Attributes

Custom attributes for the item in a list view give the user additional information.

**Universal Worklist**

Tasks (18 / 1 New) Notifications (18 / 12 Unread) Requests (10) Completed Tasks (27) Completed Requests (2)

Webflow Tasks (14 / 1) - **Vacation Approvals (5)** - TE Demo Approval (3)

**Vacation Approvals** Search

Refresh Add View To Navigation Personalize View

			Subject	Priority	Absence From	Absence To	Cost Center
<input checked="" type="radio"/>		4	<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Oct 10, 2003	Oct 10, 2003	7777
<input type="radio"/>		1	<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Oct 16, 2003	Oct 20, 2003	4567
<input type="radio"/>		1	<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Oct 16, 2003	Oct 20, 2003	4567
<input type="radio"/>		1	<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Jan 1, 2004	Jan 1, 2004	7755
<input type="radio"/>		1	<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal			

1-5

The list view is automatically added to the task list navigation if the user has items of a corresponding type. It can also be reached through the “View all of same category” link in the item detail view. In the above example “Absence From”, “Cost Center” and “Absence To” are custom attributes.

## 2.6 Decision Views

The user can directly execute decisions in the UWL.

**Universal Worklist**

Tasks (18 / 1 New) Notifications (18 / 12 Unread) Requests (10) Completed Tasks (27) Completed Requests (2)

Webflow Tasks (14 / 1) - Vacation Approvals (5) - **TE Demo Approval (3)**

**TE Demo Approval** Search

Submit Decisions Refresh Add View To Navigation Personalize View

			Subject	Priority	From	Absent From	Absent To	Reject	Approve
<input checked="" type="radio"/>		1	<a href="#">Approve vacation for YUE_demo / 04.04.2004</a>	Normal	CARTMAN	Apr 4, 2004	Apr 4, 2004	<input type="checkbox"/>	<input type="checkbox"/>
<input type="radio"/>		1	<a href="#">Approve vacation for CARTMAN / 20.02.2004</a>	Normal	CARTMAN	Feb 20, 2004	Feb 20, 2004	<input type="checkbox"/>	<input type="checkbox"/>
<input type="radio"/>		1	<a href="#">Approve vacation for CARTMAN / 23.02.2004</a>	Normal	CARTMAN	Feb 23, 2004	Feb 27, 2004	<input type="checkbox"/>	<input type="checkbox"/>

1-3

In the above example, user can choose *Reject* or *Accept* for the item and / or for multiple items and submit decisions.

## 2.7 Tailored UWL iViews

Applications and administrators can provide UWL iView instances, which deliver information to the user, which is tailored to their role.

**UWL Custom iView** ☰ ☒

---

**Vacation Approvals**  Search

[Personalize View](#) 🔊

		Subject	Priority	Absence From	Absence To
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Jan 1, 2004	Jan 1, 2004
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Oct 16, 2003	Oct 16, 2003
1		<a href="#">Employee YUE : Approve notification of absence</a>	Normal	Jun 6, 2004	Jun 6, 2004
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Feb 11, 2004	Feb 21, 2004
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Feb 23, 2004	Feb 27, 2004
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Feb 11, 2004	Feb 21, 2004
1		<a href="#">Employee CARTMAN : Approve notification of absence</a>	Normal	Feb 20, 2004	Feb 20, 2004
4		<b><a href="#">Employee CARTMAN : Approve notification of absence</a></b>	<b>Normal</b>	<b>Oct 10, 2003</b>	<b>Oct 10, 2003</b>

⏪ ⏩ 1-8 ⏪ ⏩

## 3 Step By Step - Minimal Configuration

The process is also well documented in the UWL administration documentation

[http://help.sap.com/saphelp\\_ep60sp2/helpdata/en/ff/83fd4d5a644336b13ae54528cfef57/content.htm](http://help.sap.com/saphelp_ep60sp2/helpdata/en/ff/83fd4d5a644336b13ae54528cfef57/content.htm),

(see 10.2 for navigation) but the description in this document also includes screenshots as well as a sample system landscape configuration and user mapping.

### 3.1 Prerequisites

- You must have administration rights for the Portal
- Each UWL user must have user accounts defined on the portal and the backend SAP systems they will use.
  - The users must be authorized for RFC access to function group SWK1 and transaction **SWK1**
- The executed Business Workflows must have the proper work item visualizations (see Appendix 10.5).
- The backend SAP systems you want to connect to have been configured in the system landscape, including the system alias. Each system has the following requirements:
  - Release 3.1H or higher
  - Enterprise Portal Plug-In 6.0 installed (WP-PI 6.0)
  - Internet Transaction Server
- The administrator must have done the user mappings for the backend systems. See section 3.3.

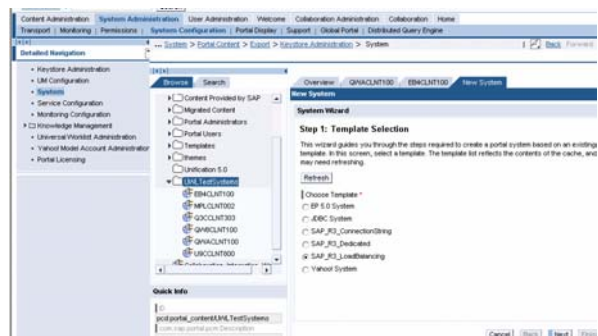
### 3.2 Creating the System in the System Landscape

Two options are available:

- Manual definition – use when creating the system for the first time.
- Importing an existing epa file – use when you want to import information (epa file) from a previously created system.

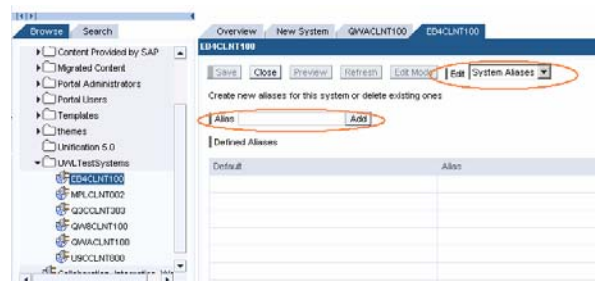
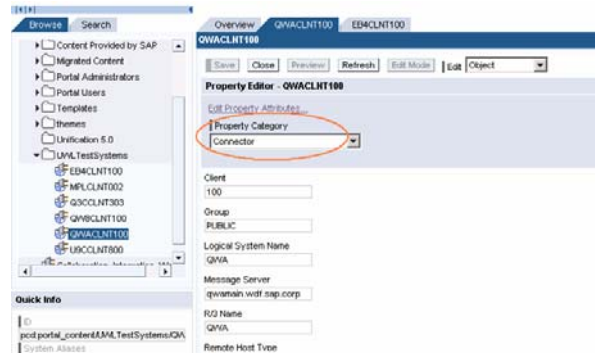
#### Manual definition

- *System Administration* → *System Configuration* → *System*  
(Optional: create your own folder under *Portal Content*)
- Pick a location under *Portal Content* where the new system should be created. Right click *New* → *System* (recommended: *SAP\_R3\_LoadBalancing*)
- Fill out the corresponding



system name and id

- Right click on the system name  
*Edit* → *Object*
- Make sure definition exists at least for the following *Property Category*:
  - Connector
  - User Management (also refer to 3.3 below)
  - ITS
- Define *system alias*
  - Choose *System Aliases* in the *Edit* pull down menu
  - Add the system alias and *save*



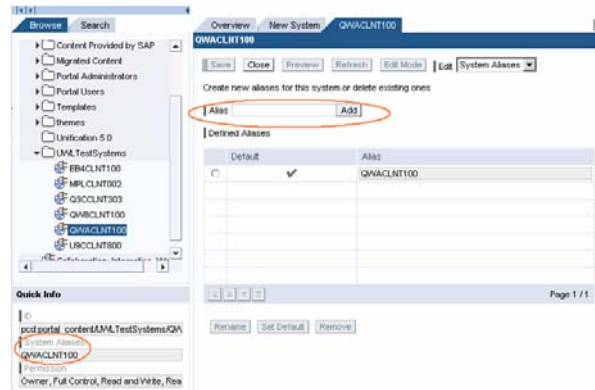
**Import an already defined EPA file**

System definition is already setup and saved in an EPA file.

- Place the epa file under the server "*<portalrelease>/global/config/pcd/Import*"
- Navigate to *System Administration* → *Transport* → *Import*
- Choose '*Server*' and use the *Browse* button to locate the epa file and click '*Select*' on the system epa file
- A list of import preview will be displayed. Click *Import*.

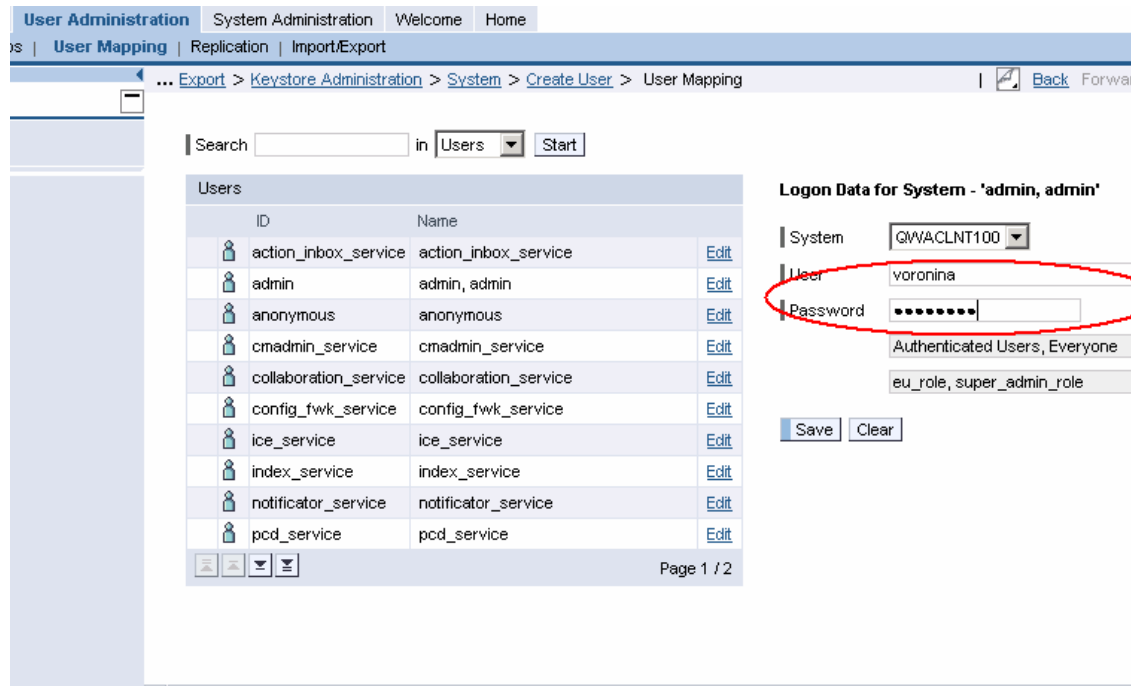


- Navigate to *System Administration* → *System Configuration* → *System*; you should be able to see the systems already created according to the definition in the EPA file.
- Make sure *system alias* is defined
  - Right click on the system name and choose *Edit* → *System Aliases*.
  - Add the system alias if not defined yet and choose *save*.



### 3.3 Mapping the Portal User to the System User

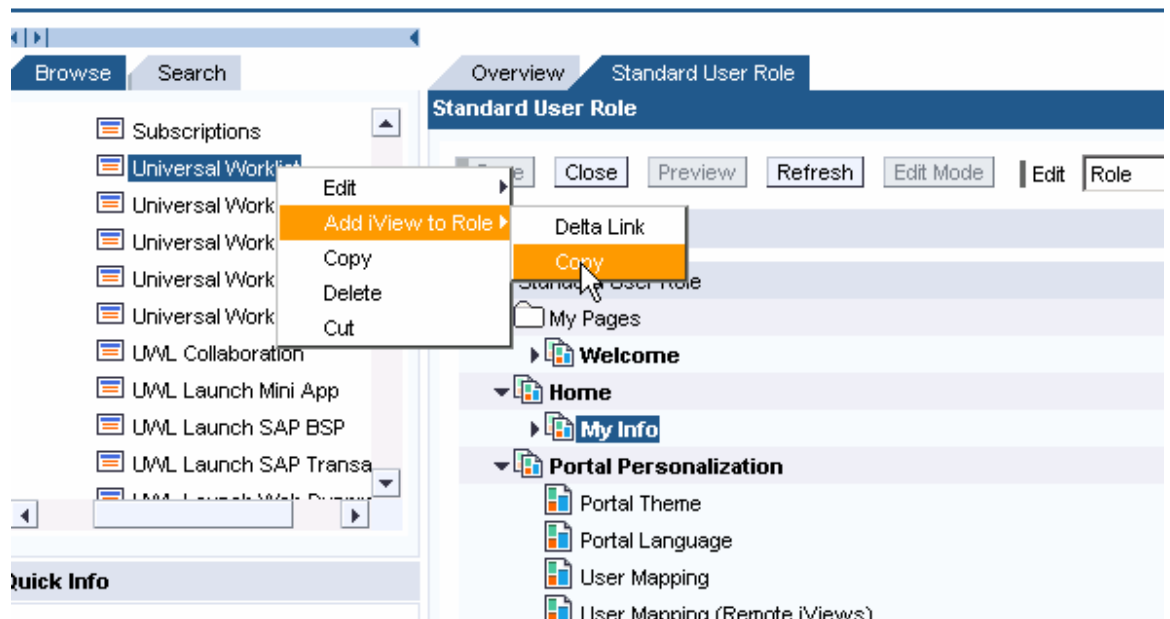
- Navigate to *User Administration* → *User Mapping*
- There are three scenarios:
  - Portal user id is different than the backend SAP user id and backend SAP user ids are different in all backend SAP systems connected to the portal (UIDPW)
    - Maintain the user mapping for EACH portal user corresponding to ALL the respective backend systems



- Portal user id is different than the backend SAP user id and backend SAP user ids are the same in all backend SAP systems connected to the portal (SAPLogon Ticket)
  - Maintain the user mapping for EACH portal user to the reference system
- Portal user id is the same as the backend SAP user and backend SAP user ids are the same in all backend SAP systems connected to the portal (SAPLogon Ticket)
  - NO user mapping is necessary

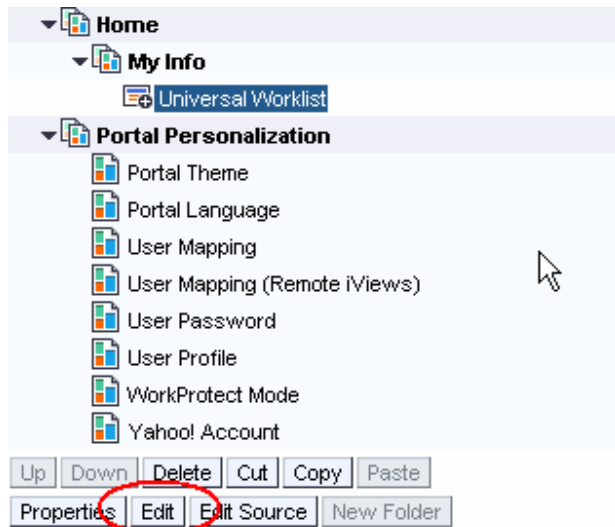
### 3.4 Adding the UWL iView to the User Role

- Locate the desired user role, for example *Content Administration* → *Portal Content*, expand the Navigation tree to *Portal Content* → *Portal Users* → *Standard Portal Users* → *Standard User Role*
  - Right click *Edit*
  - Choose *Home* → *My Info*. Choose *My Info*. The location to include the UWL is flexible; 'HOME' is an example.
- Locate the predefined UWL iView. From the same Navigation tree on the left, navigate to *Portal Content* → *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *iViews* → *Universal Worklist*
  - Select *Universal Worklist*, Right click *add iView to Role* → *Copy*
  - Make sure UWL iView is added as a COPY (not delta link)

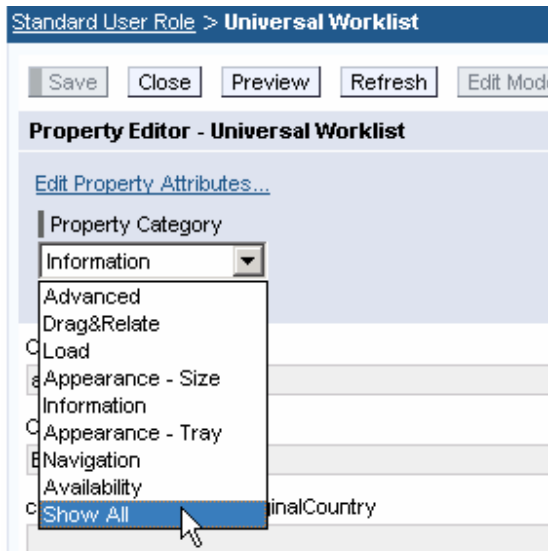


- (Optional and only available as of SP3 and higher): *System Group Configuration* in iView parameter. Specifying group name will limit the retrieval of work tasks under this iView to only those system(s) defined in the group. The group name is the optional value defined when registering system with UWL (refer to next section *Registering the System with the UWL*).
  - To edit iView parameters select the Universal Worklist iView and click 'Edit'

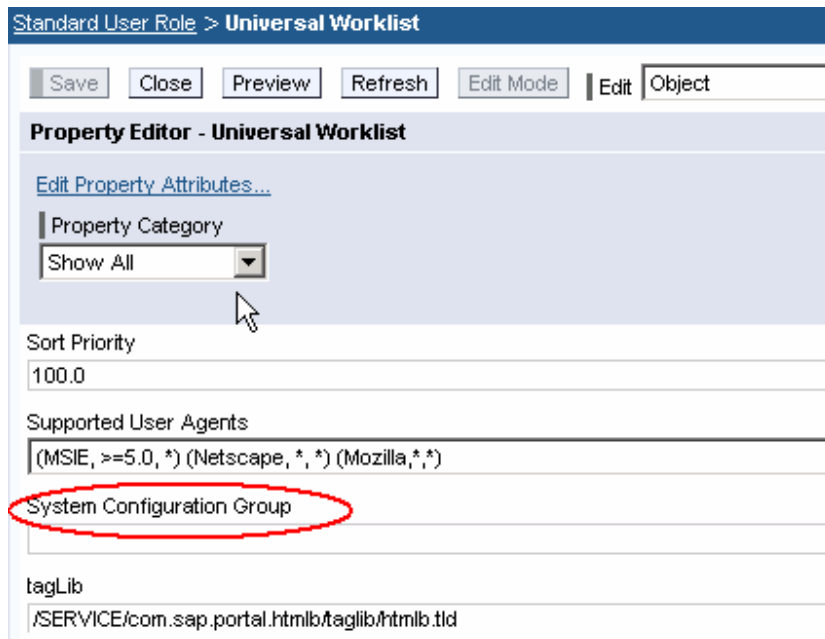




- o In the 'Property Category' select 'Show All'



- o Scroll down to 'System Configuration Group' property



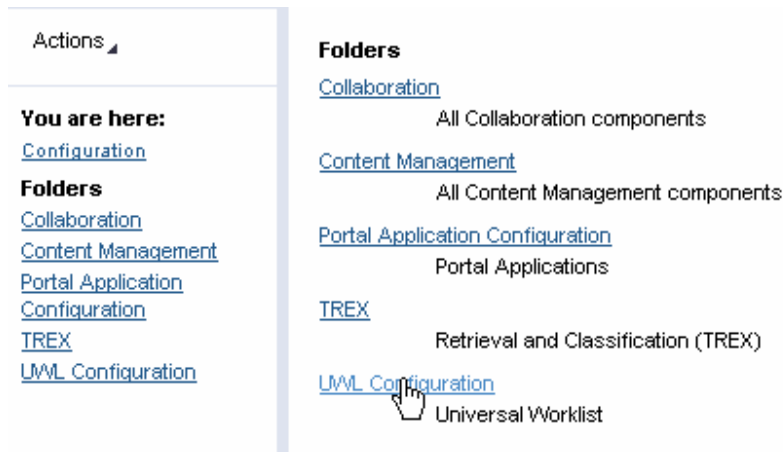
- Now, under *Home* → *My Info*, you should see the entry of Universal Worklist (there is no content yet, because configuration is not yet complete).



### 3.5 Registering the System with the UWL

Navigate to *System Administration* → *System Configuration* → *Universal Worklist Administration*. From the view *Registration of item types of Universal Worklist Webflow*, choose *Configuration Framework Page*.

- Choose *UWL Configuration*



- Choose *UWL Systems Configuration*
- Choose *New* and fill out the *Name*, *System Alias* and choose *WebFlowConnector*
- (Optional:) Fill out a configuration group id, for example; Test, Support, GroupA, GroupB, ... This can be referred by the corresponding UWL iview to restrict the scope of access to different backend systems by different users (refer to 3.4)

Here you can define connectors and systems as item providers for the Universal Worklist. The 'Connector' represents a unique id, which is used internally. The 'System Alias' is an alias for the system, defined in an alias editor in the connector, after defining a new system its item types have to be registered with the UWL service through

**Universal Worklist Systems**

Name	Configuration Group	System Alias	Connector
<input type="checkbox"/> SystemRef_1		AdHocSystem	AdHocWorkflowConnector

Page 1 / 1

New Duplicate Edit View Delete

**New "Universal Worklist Systems"**

Name \* SystemRef\_2

Configuration Group Test

System Alias \* QWACLNT100

Connector \* WebFlowConnector

OK Apply Cancel

Browse Search

- Portal Administrations
- Portal Users
- Templates
- themes
- Unification 5.0
- UWLTestSystems
  - EB4CLNT100
  - MPLCLNT002
  - Q3CCLNT303
  - QWACLNT100
  - QWACLNT100
  - U9CCLNT800

Overview QWACLNT100

QWACLNT100

Save Close Preview Refresh Edit Mode Edit System Aliases

Create new aliases for this system or delete existing ones

Alias  Add

Defined Aliases

Default	Alias
✓	QWACLNT100

Standard User Role

Standard User Role > Universal Worklist

Save Close Preview Refresh Edit Mode Edit Object

Property Editor - Universal Worklist

Edit Property Attributes...

Property Category

Show All

Sort Priority

100.0

Supported User Agents

(MSIE, >=5.0, \*) (Netscape, \*, \*) (Mozilla,\*,\*)

System Configuration Group

Test

tagLib

/SERVICE/com.sap.portal.htmlb/taglib/htmlb.tld

**3.6 Registering System Item Types with the UWL**

- Navigate to System Administration → System Configuration → Universal Worklist Administration.

Under the section of *Registration for Item Types of Universal Worklist Webflow*, register item types by clicking 'Register Item Types for All Systems'.

#### Universal Worklist Webflow Item Type Registration

Before registering item types in the UWL, Business Workflow systems have to be set up as item providers in the UWL Service Configuration:

**There are no item types from any of the system aliases of Business Workflow systems. Please generate the Business Wo systems.**

**Item types from the following new system aliases of Business Workflow systems are not yet registered:**

- QWACLNT100

[Register item types for All systems](#)

[Register item types for only new systems](#)

Item type generation requires a connection to the systems and may take a couple of minutes. Please be patient.

The generated item types can be viewed in the Universal Worklist content configuration iView. For each system, they are generated as conf item types should be re-registered after relevant changes in the workflow definitions of the Business Workflow system.

#### Universal Worklist Content Configuration

Verify the result under the section *Universal Worklist Content Configuration*. Choose option *Display or remove current configurations and their customizations*, click *Start*.

Note for SP2: The content configuration display page is directly on the admin page.

#### Universal Worklist Content Configuration

UWL Content Configuration includes definition of

- Item Types - The type of an item determines how it is executed, and if other actions can be applied. It also describes additional ir
- List Views - A view describes how the items of a specific type should be displayed in the worklist: visible attributes, sorting or c
- Navigation - The navigation defines the list views offered to the end user. The first level of the navigation hierarchy represents t

Related definitions are combined within a configuration.

**Choose one of the following options and click 'Start':**

- Display or remove current configurations and their customizations
- Upload a configuration and assign it to one or more systems
- Reset view personalizations from end users to the original views

[Start](#)

- Find the newly created configuration under the pull down menu.

**Universal Worklist Content Configuration**

Display the current content configuration in XML format. In the dropdown box, all configurations and their

**Select configuration:**

- Check that the configuration is created as “uwl.webflow.[systemalias]” with low override priority.

Note for SP2: There is no override priority. Select “Standard” as export mode.

**Universal Worklist Content Configuration**

Display the current content configuration in XML format. In the dropdown box, all configurations and their override priori

**Select configuration:**

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE UWLConfiguration PUBLIC '-
//SAP/UWL1.0/EN' 'http://www.sap.com/ctds/nw/uwl/uwl_configuration.dtd' [
]>
<UWLConfiguration version="1.0">
  <ItemTypes>
    <ItemType name="uwl.task.webflow.TS01000096.QWACLNT100" connector="WebFlowConnector"
defaultView="webflowView" defaultAction="launchSAPAction" executionMode="pessimistic">
      <ItemTypeCriteria systemId="QWACLNT100" externalType="TS01000096"
connector="WebFlowConnector"/>
    <Actions>
      <Action name="launchSAPAction" userDecision="no" handler="SAPMiniAppLauncher"
returnToDetailViewAllowed="yes" launchInNewWindow="no">
```

### 3.7 Uploading Configuration Files

Some SAP applications like Manager's Self Service (MSS) have released UWL configuration files, or you have created customized configurations as described in the following chapters.

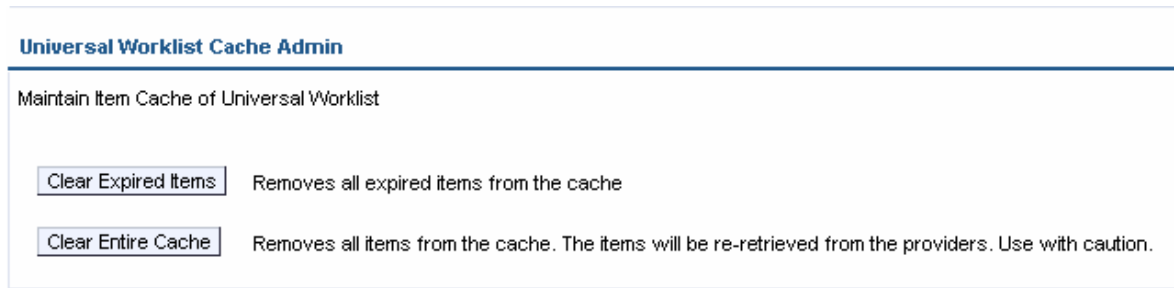
The following steps describe how to **upload** these configuration files into the UWL.

1. Navigate to *System Administration -> System Configuration -> Universal Worklist Administration*.
2. Under *iview Universal Worklist Content Configuration* select *Upload a configuration and assign it to one or more systems*. Click *Start*. (Note for SP2: the *iView* is already located on the admin page).
  - Enter the configuration name. If the name already exists, a confirmation will be prompted for overwriting. You cannot overwrite standard or generated configurations.
  - Do not check the *Override Priority* box if you upload predefined configuration shipped by an application. Unchecked, the configuration will be uploaded with priority "medium", which still overrides the "low" priority of generated Business Workflow types (see 3.6). For customization of other configurations, select high priority to override identical item types, system criteria or view names of other configurations.

Note for SP2: *Override Priority* is not available. Use "Upload Mode" instead. If you want to override existing item types or view names, "Customize" has to be selected as upload mode, and you have to specify the configuration name you want to customize (e.g. the specific webflow system configuration generated earlier)
  - Select the *Adapt to system* for which the configuration is valid. Only select "*None – use as is*" if you are sure that one of the following conditions for the configuration XML applies:
    - The configuration file contains already system specific information (the system criteria of item types have the attribute "systemId" referring to a specific system alias).
    - The configuration is valid for all Business Workflow systems, there are no conflicts with other configurations.

Note for SP2: *Adapt to system* is not available. The system has to be specified within the XML, as documented in the online help.
  - Use the *Browse* button to upload the configuration file. Click *Upload*.
3. Check for successful or error message. If successful, you can find your newly added configuration by:
  - Hitting enter (return button) and select *Display or remove current configurations and their customizations* and click *Start*.
  - Under the *Select configurations* pull down menu you can find your newly added configuration file, which corresponds to the configuration name you entered before.
4. When a configuration file is uploaded and if it contains customizations for List Views and if few users had already personalized that List View, you will be given an option to delete these personalizations. You can delete all personalized views or choose few Views. After this end users personalization are removed. Note that it is not required to delete all personalizations. However, if List Views are changed in the configuration file, it is necessary to delete the existing personalizations done by the end user for that List View. This is required for the end user to see the effect of the change.

5. Navigate to *Universal Worklist Admin* and *Cache Administration* section. Click *Clear entire Cache*. This removes all work items from the cache so the customizations done are reflected when items are refetched.

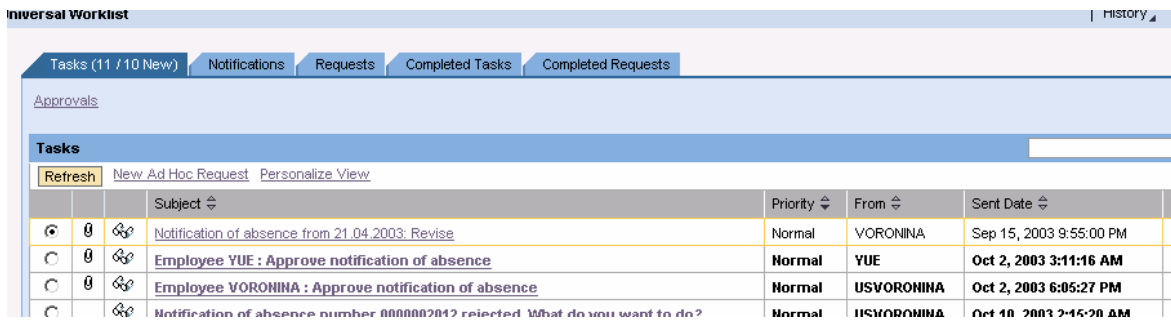


6. Test the changes if needed. Use the user id to logon to the portal and check the changes.



## 4 Step By Step – Customize existing List Views

This chapter describes how to change standard UWL list views in the Portal installation: remove or add columns, change column order, change sorting, and many other view parameters. There are five standard UWL list views defined for each tab of the UWL:



Each tab corresponds to a view name, as summarized in the following table:

Tasks	DefaultView
Notifications	NotificationsView
Requests	WorkitemRequestsView
Completed Tasks	WorkitemsCompletedView
Completed Requests	WorkitemsCompletedRequestView

The changes apply to all users.

### 4.1 Get the definition of the View to be Customized

The definition of each of the standard UWL views can be found in a standard UWL xml configuration.

1. Navigate to *System Administration -> System Configuration -> Universal Worklist Administration*. Under the section *Universal Worklist Content Configuration*, choose option *Display or remove current configurations and their customizations* and click *Start*.
2. Choose 'uwl.standard(Low)' in the drop-down menu and click either *Display Content as XML* or *Download XML* to view.
3. Locate the interested view name, for example, 'DefaultView':

**Configuration**

uwl.standard (Low)

Display Content as XML

Download XML

Delete Configuration

[Download DTD](#)

```
</ItemType>
</ItemTypes>
<Views>
  <View name="DefaultView" selectionMode="SINGLESELECT" width="98%"
supportedItemTypes="uwl.task" columnOrder="attachmentCount, detailIcon, subject, priority, creatorId,
createdDate, dueDate" sortBy="priority:descend, dueDate:ascend, createdDate:ascend"
tableDesign="STANDARD" visibleRowCount="10" headerVisible="yes"
tableNavigationFooterVisible="yes" tableNavigationType="CUSTOMNAV" actionRef="" refresh="300"
dueDateSevere="86400000" dueDateWarning="259200000" emphasizedItems="new"
displayOnlyDefinedAttributes="yes" dynamicCreationAllowed="yes" actionPosition="top"
referenceBundle="tasks">
  <Actions>
    <Action name="refresh" userDecision="no" handler="UIActionHandler"
referenceBundle="refresh" returnToDetailViewAllowed="yes" launchInNewWindow="no">
  <Properties>
```

**4.2 Create new configuration file**

Create a new configuration XML for example; **myuwlconf.xml** with the following outline structure:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE UWLConfiguration PUBLIC "-//SAP//UWL1.0//EN"
'uwl_configuration.dtd' [ ]>
<UWLConfiguration>
  <Views>
  </Views>
</UWLConfiguration>
```

Note: this file can be at any location on your system.

(Refer to Appendix 10.1 for DTD definition).

**4.3 Copy the existing view definition to the new configuration file**

Copy and paste the definition in a new xml file in a following way (according to the DTD):

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE UWLConfiguration PUBLIC "-//SAP//UWL1.0//EN"
'uwl_configuration.dtd' [
  ]>
<UWLConfiguration>
<Views>
```

```

<View name="DefaultView" selectionMode="SINGLESELECT" width="98%"
supportedItemTypes="uwl.task" columnOrder="attachmentCount, detailIcon,
subject, priority, creatorId, createdAt, dueDate"
sortBy="priority:descend, dueDate:ascend, createdAt:ascend"
tableDesign="STANDARD" visibleRowCount="10" headerVisible="yes"
tableNavigationFooterVisible="yes" tableNavigationType="CUSTOMNAV"
actionRef="" refresh="300" dueDateSevere="8640000"
dueDateWarning="25920000" emphasizedItems="new"
displayOnlyDefinedAttributes="yes" dynamicCreationAllowed="yes"
actionPosition="top" referenceBundle="tasks">

.....

</View>
</Views>
</UWLConfiguration>
    
```

#### 4.4 Modify the view

Change parameters of the target view (refer to DTD doc for valid parameters)

##### 4.4.1 Remove or add columns

To remove a column from the view the corresponding definition has to be deleted from 'columnOrder' string. For example, columnOrder="attachmentCount, detailIcon, subject, priority, createdAt, dueDate" will create a "DefaultView" with no "From" column.

		Subject	Priority	Sent Date	Due Date
		Test001	Normal	Mar 19, 2004 9:47 PM	Mar 26, 2004
		Workflow verification: Create notification of absence	Normal	Oct 10, 2003 8:52 PM	
		Notification of absence from 16.10.2003: Revise	Normal	Oct 23, 2003 4:29 PM	
		Employee CARTMAN : Approve notification of absence	Normal	Nov 17, 2003 7:37 PM	
		Approve vacation for YUE demo / 04.04.2004	Normal	Feb 11, 2004 5:23 AM	
		Approve vacation for CARTMAN / 20.02.2004	Normal	Feb 18, 2004 11:05 AM	
		Approve vacation for CARTMAN / 23.02.2004	Normal	Feb 18, 2004 11:16 AM	
		<b>Employee CARTMAN : Approve notification of absence</b>	Normal	<b>Feb 18, 2004 12:43 PM</b>	
		Notification of absence from 11.10.2003: Revise	Normal	Feb 19, 2004 11:14 AM	
		Employee CARTMAN : Approve notification of absence	Normal	Feb 19, 2004 11:16 AM	

#### 4.4.2 Change column order

To change the column order the corresponding definitions have to be rearranged in the 'columnOrder' string. For example,

columnOrder="attachmentCount, detailIcon, priority, subject, creatorId, createdDate, dueDate"

will change the order of "Subject" and "Priority" columns.

Tasks							Search
		Priority	Subject	From	Sent Date	Due Date	
<input checked="" type="radio"/>		Normal	Test001	admin, admin	Mar 19, 2004 9:47 PM	Mar 26, 2004	
<input type="radio"/>		Normal	Workflow verification: Create notification of absence	YUE	Oct 10, 2003 8:52 PM		
<input type="radio"/>		Normal	Notification of absence from 16.10.2003: Revise	CARTMAN	Oct 23, 2003 4:29 PM		
<input type="radio"/>		Normal	Employee CARTMAN : Approve notification of absence	USCARTMAN	Nov 17, 2003 7:37 PM		
<input type="radio"/>		Normal	Approve vacation for YUE demo / 04.04.2004	CARTMAN	Feb 11, 2004 5:23 AM		
<input type="radio"/>		Normal	Approve vacation for CARTMAN / 20.02.2004	CARTMAN	Feb 18, 2004 11:05 AM		
<input type="radio"/>		Normal	Approve vacation for CARTMAN / 23.02.2004	CARTMAN	Feb 18, 2004 11:16 AM		
<input type="radio"/>		Normal	Employee CARTMAN : Approve notification	CARTMAN	Feb 18, 2004 12:43 PM		

#### 4.4.3 Change sorting

To change the sorting order the corresponding definitions have to be modified in 'sortby' string. For example,

sortby="creatorId:descend "

will sort the Worklist by 'From' column.

Tasks							Search
		Priority	Subject	From	Sent Date	Due Date	
<input checked="" type="radio"/>		Normal	Workflow verification: Create notification of absence	YUE	Oct 10, 2003 8:52 PM		
<input type="radio"/>		Normal	Test001	admin, admin	Mar 19, 2004 9:47 PM	Mar 26, 2004	
<input type="radio"/>		Normal	Employee CARTMAN : Approve notification of absence	USCARTMAN	Nov 17, 2003 7:37 PM		
<input type="radio"/>		Normal	Notification of absence from 11.10.2003: Revise	USCARTMAN	Feb 19, 2004 11:14 AM		
<input type="radio"/>		Normal	Employee CARTMAN : Approve notification of absence	USCARTMAN	Feb 19, 2004 11:16 AM		
<input type="radio"/>		Normal	Notification of absence from 16.10.2003: Revise	USCARTMAN	Feb 24, 2004 10:32 AM		
<input type="radio"/>		Normal	Employee CARTMAN : Approve notification of absence	USCARTMAN	Feb 26, 2004 1:38 PM		

#### 4.4.4 Change date format

To change the date format, for example, of 'Send Date' column find its definition in 'uwl.standard' configuration and copy it into your custom xml file inside of <DisplayAttributes> node of your custom view:

```
<UWLConfiguration>
  <Views>
    <View name="DefaultView"....
      <DisplayAttributes>
        <DisplayAttribute name="createdDate"
          type="datetime"
          width=" " sortable="yes"
          format="medium"
          referenceBundle="created_date"
          hAlign="LEFT" vAlign="BASELINE" />
        .....
      </DisplayAttributes>
    </View>
  </Views>
</UWLConfiguration>
```

Changing type from "datetime" to "date" will remove the time portion of the date in the 'Send Date' column.

Priority	Subject	From	Sent Date	Due Date
Normal	Workflow verification: Create notification of absence	YUE	Oct 10, 2003	
Normal	Test001	admin, admin	Mar 19, 2004	Mar 26, 2004
Normal	Employee CARTMAN: Approve notification of absence	USCARTMAN	Nov 17, 2003	
Normal	Notification of absence from 11.10.2003: Revise	USCARTMAN	Feb 19, 2004	
Normal	Employee CARTMAN: Approve notification of absence	USCARTMAN	Feb 19, 2004	
Normal	Notification of absence from 16.10.2003: Revise	USCARTMAN	Feb 24, 2004	
Normal	Employee CARTMAN: Approve notification of absence	USCARTMAN	Feb 26, 2004	

#### 4.5 Upload the config file

After the change of the view definition (preferably only delta change included) upload the myuwlconf.xml file as described in section 3.7.

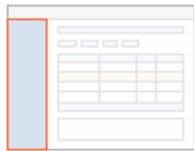
*Note: System Administrator is asked to delete the existing users personalizations by going to System Administration -> System Configuration -> Universal Worklist Administration. On Universal Worklist Content Configuration select 'Reset view personalization to original view' and click 'Start'. Enter the Personalized View Name and click "Delete Personalized View for All Users' button.*

**Universal Worklist Content Configuration**

Personalized views are administered here. Specifying \* for view name deletes all personalized views in the configuration.

Personalized View Name  \* removes personalizations for all views

*If the view was previously personalized by the user and the administrator doesn't perform the above step, then the changes to the view will not be visible until the user clicks on 'Personalize View' and then 'Restore View' button.*



**Page**

Personalize the following view

**Tasks**

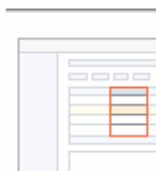
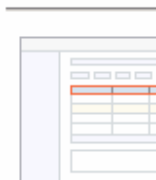
Page View Name:

**DefaultView**

Page View Name Space:

**test\_view**

[Advanced View](#)



## 5 Step By Step – Item Launch Configuration

When the user clicks on the task subject, a page opens in which the user can perform task related work. The pages can be rendered by different UI technologies, which include: SAP GUI Transaction, ITS template pages, iViews, WebDynpro applications, or just any URL.

The task type launch information is stored in the UWL configuration. It is created by the following two ways:

### 1. Registering the SAP system item types with the UWL

The backend system running the Business Workflow already contains item launch information. During step 3.5 of the UWL configuration, the UWL service connects to the backend, reads the complete launch configuration, converts it into the UWL format and stores it as `uwl.webflow.<systemalias>` configuration entry.

In the SAP system, Business Workflow launch configuration is specified at the task definition or the `SWFVISU` transaction. (see 10.5)

Initially, work items were executed either using the SAP GUI or ITS MiniApps. Later BSP pages were added through “Web Services”, and the recent addition of transaction `SWFVISU` (as of 6.20 SP19) allows also to specify iViews and WebDynpro applications as so-called work item visualization.

Edit done in this transaction should be followed by the item type registration again (see 3.6).

See 4.1 for the inspection of the generated configuration.

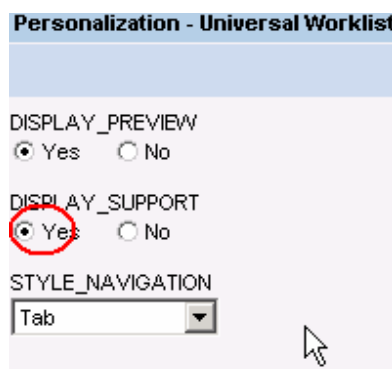
### 2. Uploading configuration files in XML format

- a) Predefined launch configuration from SAP applications (for example: MSS) – see 3.7
- b) Custom launch configuration – see 4.2 and following.

### 5.1 Inspecting Current Item Launch Configuration

Locate the interested *Item Type* and its associated **defaultAction** property as follows.

- In the UWL iview upper right corner, choose *Personalize*, a pop-up appears. Toggle the *Display UWL Support Information* to TRUE.



- Select the item interested

			<a href="#">Approve vacation for CARTMAN / 23.02.2004</a>	Normal	CARTMAN	Feb 18, 2004 11:10 AM
			<a href="#">Employee CARTMAN: Approve notification of absence</a>	Normal	CARTMAN	Feb 18, 2004 12:43 PM
			<a href="#">Notification of absence from 11.10.2003: Revise</a>	Normal	USCARTMAN	Feb 19, 2004 11:14 AM
			<a href="#">Employee CARTMAN: Approve notification of absence</a>	Normal	USCARTMAN	Feb 19, 2004 11:16 AM

and scroll down for support information and look up value for *Item Type Configuration Name* and *Item Type*.

**Current Item Information**

- Item Type Configuration Name: uwl.webflow
- Item Type: uwl.task.webflow.default
- Item Type associated view: DefaultView
- Internal Item ID: 47
- External Item ID: 000000550695
- App Context: null
- System ID: QWACLNT100
- Connector: WebFlowConnector
- Execution URL:

- Then, navigate to *System Administration -> System Configuration -> Universal Worklist Administration*. Under the section *Universal Worklist Content Configuration*, choose option *Display or remove current configurations and their customizations* and click *Start*.
- Choose the configuration corresponding to the value of *Item Type Configuration Name* and either *Display Content as XML* or *Download XML* to view.
- Locate the interested *Item Type* and its associated *defaultAction* property.



**Universal Worklist Content Configuration**

Display the current content configuration as XML.

**Select configuration:**

uw1.webflow (Low)

[Download DTD](#)

```

</ShortDescriptions>
<LongDescriptions>
  <Description Language="en" Description="View Detail in SAP Gui"/>
</LongDescriptions>
</Descriptions>
</MenuItem>
</Menu>
</ItemType>
<ItemType name="uw1.task.webflow.decision" connector="WebFlowConnector"
defaultView="DefaultView" defaultAction="viewDetail" executionMode="default"/>
<ItemType name="uw1.task.webflow.default" connector="WebFlowConnector"
defaultView="DefaultView" defaultAction="launchSAPAction" executionMode="default">
  <Actions>
    <Action name="launchSAPAction" userDecision="no" handler="SAPTransactionLauncher"
referenceBundle="launch_sap" returnToDetailViewAllowed="yes" launchInNewWindow="no">
  </Properties>

```

Note that the launch configuration for Business Workflow items does usually not include items which launch a transaction in SAP GUI for HTML. This launch is delegated to the Business Workflow system (transaction `SWK1`), which determines the transaction to be launched at runtime. These items have the general type `uw1.task.webflow.default`. To customize their launch, a new specific type can be added as described below.

## 5.2 Creating a New Configuration File

Create a new configuration XML for example; `myuw1conf.xml` with the following outline structure:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE UWLConfiguration PUBLIC "-//SAP//UWL1.0//EN"
'uw1_configuration.dtd' [ ]>
<UWLConfiguration>
  <ItemTypes>
  </ItemTypes>
  <Views>
  </Views>
</UWLConfiguration>

```

Note: this file can be at any location on your system.

(Refer to Appendix 10.1 for DTD definition).

It is recommended to follow the steps described in 5.1 "Inspecting Current Item Launch Configuration". With the feature *Display or remove current configurations and their customizations*, find the relevant section of the XML definition from an existing configuration (for example: `uwl.webflow`) and copy it to the file.

Add all your custom item types and view definitions to this file, as described in the following chapters.

Verify the validity of `myuwlconf.xml`.

1. Download the `uwl_configuration.dtd` as follows: See *System Administration* → *System Configuration* → *Universal Worklist Administration*.
2. Under the section of *Universal Worklist Content Configuration*, select *Display or remove current configurations and their customizations* and click *Start*.
3. Click *Download DTD*. Save the file in the same directory as `myuwlconf.xml`. View `myuwlconf.xml` in the Internet Explorer. It shows problems as detailed parsing errors.
4. Upload the configuration into the UWL as described in 3.7.

### 5.3 Adding Custom Launch Configuration for iViews

You wrote an iView to execute Business Workflow task TS91100019 and added it to the Portal Content catalog at

```
pcd://portal_content/com.sap.pct/every_user/general/iViews/myDemoIView.
```

(For information on how to implement such an iView, please see Appendix 10.4. The iView communicates directly with Business Workflow through the Java Connector to retrieve additional information and to send the terminating event once the work is done).

The following steps describe how to configure the UWL to launch this iView for task items of type TS91100019. It is assumed that this type was executed in SAP GUI for HTML, so no item type definition exists in the UWL yet (see previous section).

1. Open your existing configuration XML file, or create a new one as described in 5.2.
2. Add a new item type within the `<ItemTypes>` tag with a unique type name:

```
<ItemType name="uwl.task.webflow.TS91100019" connector="WebFlowConnector"
defaultView="webflowView">
</ItemType>
```

Note that the item type name defines a type hierarchy. The new type is defined of a sub type of `uwl.task.webflow`.

Sub types inherit the actions of their parent types. In addition, they are included in all list views which refer to their parent type.

3. Map the new type to a backend task id (which is the external type) by specifying the item type criteria within the <ItemType>:

```
<ItemTypeCriteria
externalType="TS91100019" connector="WebFlowConnector"/>
```

Do not specify the systemid attribute. It is recommended to keep the configuration independent of a specific system. The system, to which the configuration should apply to, is better specified in the upload page (“Adapt to system”), see 3.7.

Note for SP2: The systemid need to be specified since the “Adapt to system” is only available as of SP3 and higher. It is also required to add the systemid to the type name, e.g. “uwl.task.webflow.G9CCLNT001\_TS91100019”

4. Add the launch action to the <ItemType>:

```
<Actions>
  <Action name="launchDemoIView" handler="IViewLauncher">
    <Properties>
      <Property name="iview" value=
"pcd://portal_content/com.sap.pct/every_user/general/iViews/myDemoIView"/>
    </Properties>
    <Descriptions default="Launch Demo IView">
      <ShortDescriptions>
        <Description Language="en" Description="Launch Demo IView"/>
      </ShortDescriptions>
    </Descriptions>
  </Action>
</Actions>
```

The launch action requires an IViewLauncher as handler.

For details on this or other launch handlers e.g. WebDynpro applications and their properties see the appendix.

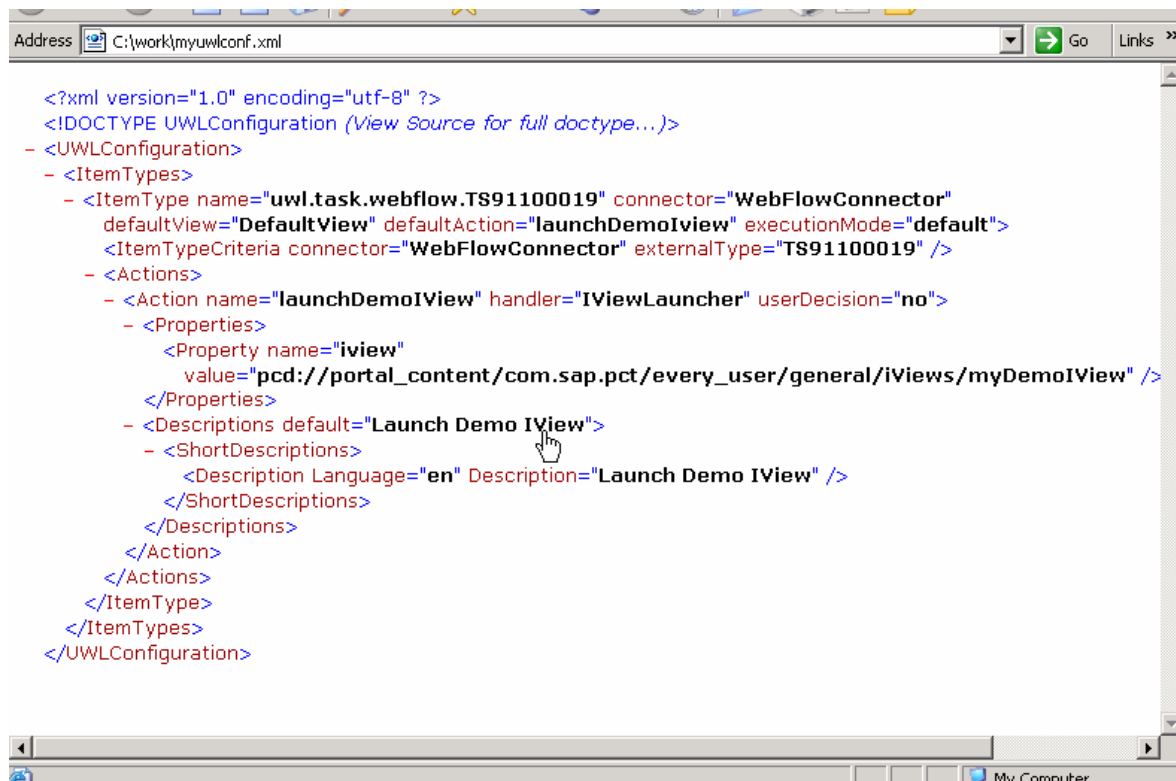
The description appears on the action button in the preview or details view and can be internationalized.

5. Add the launch action as default action to the item type:

```
<ItemType name="uwl.task.webflow.TS91100019" connector="WebFlowConnector"
defaultView="DefaultView" defaultAction="launchDemoIView">
```

The default action is the action which is launched when the user clicks on the “subject” of the task in the list view.

6. Save the myuwlconf.xml and verify the validity in Internet Explorer (see 5.2.3). It should look as follows:



```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE UWLConfiguration (View Source for full doctype...)>
- <UWLConfiguration>
- <ItemTypes>
- <ItemType name="uwl.task.webflow.TS91100019" connector="WebFlowConnector"
  defaultView="DefaultView" defaultAction="launchDemoIView" executionMode="default">
  <ItemTypeCriteria connector="WebFlowConnector" externalType="TS91100019" />
  - <Actions>
  - <Action name="launchDemoIView" handler="IViewLauncher" userDecision="no">
    - <Properties>
      <Property name="iview"
        value="pcd://portal_content/com.sap.pct/every_user/general/iViews/myDemoIView" />
    </Properties>
    - <Descriptions default="Launch Demo IView">
      - <ShortDescriptions>
        <Description Language="en" Description="Launch Demo IView" />
      </ShortDescriptions>
    </Descriptions>
  </Action>
  </Actions>
  </ItemType>
</ItemTypes>
</UWLConfiguration>
```

7. Upload the configuration as described in 3.7 “Uploading Configuration Files”. Select the system you would like the configuration to apply to.

Note you can also register UWL configuration files programmatically from other Portal components (so that manual upload is not required). Please refer to the appendix 10.3 for details.

8. Create a work item for a user, and verify the launch of the iView by clicking on the subject of the item in the UWL.

Demo Form				
<input type="button" value="Approve"/> <input type="button" value="Reject"/>				
Notific. of absence No. 0000002062				
Name	YUE_demo			
Department				
Personel no.	00000000	Cost center	3934	
Absent From (day 1)	Absent To (last day)	Hours absent	Vacation	Special Leave
2004-04-04	2004-04-04	0	X	
		0	X	
		0	X	
Reason				
Contact at				
Date	2004-02-04	Date	0000-00-00	
Issuer	YUE	Approver		

## 5.4 Overriding ITS MiniApp Launch Configuration

Locate the id of the interested task, either from the SAP Business Workflow or from the UWL as described in section 5.1. In this example, you want to trigger an ITS MiniApp instead of triggering an iview as done in the previous section. But steps are similar.

Then, follow the first few steps as listed in 5.3 to work on the <ItemType> of the corresponding task id, and in step 4, change the <Action> section for the task to indicate the MiniApp template, in this case "BWWF\_WI\_DECI":

```
<Action name="launchSAPAction" userDecision="no"
handler="SAPMiniAppLauncher" launchInNewWindow="no">
  <Properties>
    <Property name="template" value="BWWF_WI_DECI"/>
  </Properties>
  <Descriptions default="Launch SAP Task">
    <ShortDescriptions>
      <Description Language="en" Description=" Launch SAP Task"/>
    </ShortDescriptions>
  </Descriptions>
</Action>
```

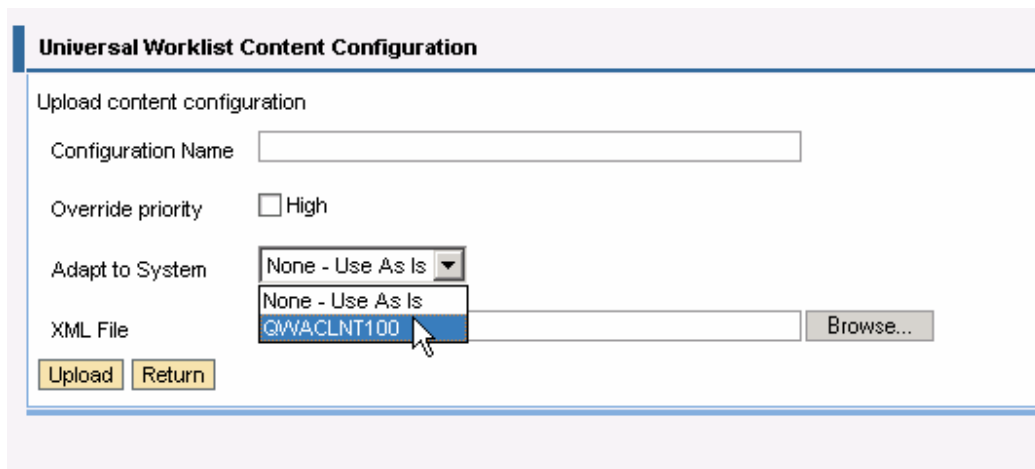
Finish up with step 6 and 7. Verify the execution of the corresponding item is launching the defined MiniApp template.

### 5.5 EP 5.0 Backward Compatibility: Converting Override XML File

If you have modified uwltaskmapping.xml or override.xml, you should convert the XML as follows and upload in the configuration of each Business Workflow system (example task TS20000193, system QWACLNT100).

Old	New
<pre>&lt;mappings&gt; ... &lt;/mappings&gt;</pre>	<pre>&lt;!DOCTYPE UWLConfiguration PUBLIC '- //SAP//UWL1.0//EN' 'uwl_configuration.dtd' [ ]&gt; &lt;UWLConfiguration&gt; &lt;ItemTypes&gt; ... &lt;/ItemTypes&gt; &lt;/UWLConfiguration&gt;</pre>
<pre>&lt;mapping logsys="QWACLNT100" task="TS20000193" template="" iview="com.sap.pct.ess.time.ws20 000081" javaiview="" url=""/&gt;</pre>	<pre>&lt;ItemType name="uwl.task.webflow. TS20000193" connector="WebFlowConnector" defaultView="defaultView" defaultAction="launchIView" &gt; &lt;ItemTypeCriteria connector="WebFlowConnector" externalType="TS20000193"/&gt; &lt;Actions&gt; &lt;Action name="launchIView" handler="IViewLauncher"&gt; &lt;Properties&gt; &lt;Property name="iview" value="com.sap.pct.ess.time.ws20000081"/ &gt; &lt;/Properties&gt; &lt;/Action&gt; &lt;/Actions&gt; &lt;/ItemType&gt;</pre>
<pre>&lt;mapping logsys="G9CCLNT001" task="TS20000193" template="" iview=' ' javaiview="ws20000081" url=""/&gt;</pre>	<pre>&lt;ItemType ... [same as cell above, the only difference:] &lt;Property name="iview" value="pcd:portal_content/...../ws200 00081" /&gt;</pre>

When uploading the modified configuration into the UWL select QWACLNT100 from the drop-down menu for 'Adapt to System':



Note for SP2: For configuration of backward compatibility for SP2 see [http://help.sap.com/saphelp\\_ep60sp2/helpdata/en/ff/83fd4d5a644336b13ae54528cfef57/frameset.htm](http://help.sap.com/saphelp_ep60sp2/helpdata/en/ff/83fd4d5a644336b13ae54528cfef57/frameset.htm)

(see 10.2 for navigation).

## 6 Step By Step – Custom Item Attributes

Additional custom attributes of an item can be retrieved at runtime. The source of the attributes can be configured and currently, the supported sources are SAP Business Object Repository (ABAP BOR), Business Workflow container (Webflow Container), User Management Engine (UME).

Definition can be specified in the configuration file. The setup of the source is associated with the item type of the item, and involves mainly the following block:

```
<CustomAttributes>
  <CustomAttributeSource id="..."
    objectIdHolder="..."
    objectType="..."
    cacheValidity="final">
    <Attribute name="..." type="..." displayName="..." />
    <Attribute name="..." type="..." displayName="..." />
  </CustomAttributeSource>
</CustomAttributes>
```

### ABAP BOR

```
<CustomAttributeSource id="ABAP_BOR"
  objectIdHolder="externalObjectId"
  objectType="FORMABSENC"
  cacheValidity="final">
  <Attribute name="COSTCENTER" type="string"
    displayName="Cost Center" />
  <Attribute name="FIRSTDAYOFABSENCE" type="date"
    displayName="First day of absence" />
  <Attribute name="LASTDAYOFABSENCE" type="date"
    displayName="Last day of absence" />
</CustomAttributeSource>
```

### Webflow Container

```
<CustomAttributeSource id="WEBFLOW_CONTAINER"
  objectIdHolder="externalId "
  objectType="WebflowContainer "
  cacheValidity="final">
  <Attribute name="FIRSTDAYABS" type="date"
    displayName="WF First day of absence" />
  <Attribute name="LASTDAYABS" type="date"
    displayName="WF Last day of absence" />
```



```
</CustomAttributeSource>
```

**UME**

```
<CustomAttributeSource id="UM"
    objectIdHolder="creatorId"
    objectType="user"
    cacheValidity="final">
  <Attribute name="lastName" type="string" displayName="Last Name" />
  <Attribute name="firstName" type="string" displayName="First Name" />
  <Attribute name="department" type="string" displayName="Department" />
  <Attribute name="telephone" type="string" displayName="Telephone" />
</CustomAttributeSource>
```

The following steps describe how to configure the UWL to include custom attributes for task items of type TS91100019

- Open your existing configuration XML file, or create a new one as described in 5.2.
- Add a new item type within the <ItemTypes> tag with a unique type name:

```
<ItemType name="uwl.task.webflow.TS91100019"
connector="WebFlowConnector" defaultView="webflowView">
</ItemType>
```

- Map the new type to a backend task id (which is the external type) by specifying the item type criteria within the <ItemType>:

```
<ItemTypeCriteria
externalType="TS91100019" connector="WebFlowConnector" />
```

Do not specify the systemid attribute.

- Depends on the custom attributes source for the <ItemType>, add the definition accordingly within the <CustomAttributes> block:

```
<ItemTypes>
  <ItemType name="uwl.task.webflow.TS91100019"
    connector="WebFlowConnector"
    .....other definitions go here>
  <ItemTypeCriteria connector="WebFlowConnector"
    externalType="TS91100019" />
</ItemTypes>
```

```
<CustomAttributes>
  <CustomAttributeSource id="..."
    objectIdHolder="..."
    objectType="..."
    cacheValidity="final">
    <Attribute name="..." type="..." displayName="..." />
    <Attribute name="..." type="..." displayName="..." />
  </CustomAttributeSource>
</CustomAttributes>
.....other definitions go here>
</ItemType>
...
</ItemTypes>
```

- Save the xml definition and verify the validity in Internet Explorer .
- Upload the configuration as described in 3.7 “Uploading Configuration Files”.
- Create a work item for a user, and verify the display of the custom attributes by clicking the detail screen of the item in the UWL.

## 7 Step By Step – List Views with Custom Attributes

### 7.1 Prerequisites

Custom Item attributes should be setup as described in the section 6.

### 7.2 Customization File

Navigate to *System Administration* → *System Configuration* → *Universal Worklist Administration*.

Download the configuration that was used for setting up the Custom attributes in section 4.

### 7.3 Adding List view with Custom Attributes

1. Open your existing configuration XML file
2. Change default view for item type `uwl.task.webflow.TS30000016` to `myApprovalView`. This `myApprovalView` will be defined shortly

```
<ItemType name="uwl.task.webflow.TS30000016" connector="WebFlowConnector"
defaultView="myApprovalView">
.....other definitions go here
</ItemType>
```

3. View you are going to define looks like below. Each section of this view is described in next steps.

```
<View name="myApprovalView" width="98%"
supportedItemTypes="uwl.task.webflow.TS30000016"
columnOrder="attachmentCount, detailIcon, subject, priority,
creatorId, createdDate, FIRSDAYOFABSENCE, LASTDAYOFABSENCE"
sortBy="priority, createdDate"
headerVisible="yes"
tableNavigationFooterVisible="yes"
displayOnlyDefinedAttributes="no">

<Descriptions default="Vacation Approval">
  <ShortDescriptions>
    <Description Language="en" Description="Vacation Approvals"/>
  </ShortDescriptions>
</Descriptions>
```

```

<DisplayAttributes>
  <DisplayAttribute name="COSTCENTER" type="string"
    width="" sortable="yes" format="medium">
    <Descriptions default="Cost Center">
      <ShortDescriptions>
        <Description Language="en" Description="Cost Center"/>
      </ShortDescriptions>
    </Descriptions>
  </DisplayAttribute>
  <DisplayAttribute name="FIRSTDAYOFABSENCE" type="date"
    width="" sortable="yes" format="medium">
    <Descriptions default="Absence From">
      <ShortDescriptions>
        <Description Language="en" Description="Absence From"/>
      </ShortDescriptions>
    </Descriptions>
  </DisplayAttribute>
  <DisplayAttribute name="LASTDAYOFABSENCE"
    type="date" width="" sortable="yes"
    format="medium">
    <Descriptions default="Absence To">
      <ShortDescriptions>
        <Description Language="en" Description="Absence To"/>
      </ShortDescriptions>
    </Descriptions>
  </DisplayAttribute>
</DisplayAttributes>
  <Actions>
    <Action reference="refresh" />
    <Action reference="removeFromNavigation" />
    <Action reference="addToNavigation" />
    <Action reference="personalize" />
  </Actions>
</View>

```

3. Define myApprovalView like below. Most of the attributes are default.

```

<View name="myApprovalView" width="98%"
  supportedItemTypes="uwl.task.webflow.TS30000016"
  columnOrder="attachmentCount, detailIcon, subject, priority,
  creatorId, createdDate, FIRSTDAYOFABSENCE, LASTDAYOFABSENCE"

```

```

        sortBy="priority, createDate"
        headerVisible="yes"
        tableNavigationFooterVisible="yes"
        displayOnlyDefinedAttributes="no">
</View>

```

4. Specify the supported itemType as uw1.task.webflow.TS30000016 which was used in the section 6 for adding custom attributes

```

<View name="myApprovalView" width="98%"
        supportedItemTypes="uw1.task.webflow.TS30000016"
.....

```

5. Add display name for the View

```

<Descriptions default="Vacation Approval">
<ShortDescriptions>
        <Description Language="en" Description="Vacation Approvals"/>
        </ShortDescriptions>
</Descriptions>

```

6. Since we need to display custom attributes, add these as display attributes for the view

```

<DisplayAttributes>
        <DisplayAttribute name="COSTCENTER" type="string"
                width="" sortable="yes"
                format="medium">
        <Descriptions default="Cost Center">
                <ShortDescriptions>
                        <Description Language="en"
                                Description="Cost Center"/>
                </ShortDescriptions>
        </Descriptions>
        </DisplayAttribute>
        <DisplayAttribute name="FIRSTDAYOFABSENCE" type="date"
                width="" sortable="yes"
                format="medium">
        <Descriptions default="Absence From">
                <ShortDescriptions>
                        <Description Language="en"
                                Description="Absence From"/>
                </ShortDescriptions>
        </Descriptions>
        </DisplayAttribute>
        <DisplayAttribute name="LASTDAYOFABSENCE" type="date"
                width="" sortable="yes"

```

```
        format="medium">
    <Descriptions default="Absence To">
        <ShortDescriptions>
            <Description Language="en"
                Description="Absence To" />
        </ShortDescriptions>
    </Descriptions>
</DisplayAttribute>
</DisplayAttributes>
```

## 7. Add the actions to be displayed in the view

```
<Actions>
    <Action reference="refresh" />
    <Action reference="removeFromNavigation" />
    <Action reference="addToNavigation" />
    <Action reference="personalize" />
</Actions>
```

### 7.4 Upload the configuration file

Refer to section 3.7.

### 7.5 Verify the Custom attributes List View

Go to Universal Worklist page that contains item types of task type `TS30000016`. You should be seeing a link with text that is set up in 7.3 step 5, if items are present. Clicking on the link leads to the Custom view just created.

Alternatively, in the list select any item of type `TS30000016`. Click on view similar types button. This also leads to the Custom attributes list view.

## 8 Step By Step – Decision Views

There are cases that item execution is fairly straightforward operation, such as user decisions. Selecting option for an individual task can easily be done along with the other similar tasks. For example, a user can choose to withdraw from or revise an application, an employer can approve or reject multiple requests at one time.

There are 3 common scenarios for providing options for users to choose and complete the task.

### 8.1 Generic user decision

- In Business Workflow, user can define a task as a decision task by assigning "DECISION" as the underlying business object type (refer to Appendix 10.5). The task definition allows user to create two or more decision options to complete the task. Refer to the standard generic user decision task 'TS00008267'.
- During runtime, UWL retrieves the available options from the workflow provider and these options are visible in the detail view of the item.
- It is also possible to display these decision options directly in the UWL list and execute the decisions, provided technical data of decision options is known in advance and can be configured in the configuration file, e.g. Revise or Withdraw (a rejected request)
- The main update is the definition under your customized view (<View> section). Below shows a sample to display 2 options as *checkbox* to either *revise a request* (with internal value 1) or *withdraw from a request* (with internal value 2):

```

<DisplayAttributes>
  <DisplayAttribute name="reviseCol" type="checkbox"
    width="" actionRef="1" vAlign="TOP"
    hAlign="CENTER">
    <Descriptions default="">
      <ShortDescriptions>
        <Description Language="en"
          Description="Revise"/>
      </ShortDescriptions>
    </Descriptions>
  </DisplayAttribute>
  <DisplayAttribute name="dropCol" type="checkbox" width=""
    actionRef="2" vAlign="TOP"
    hAlign="CENTER">
    <Descriptions default="">
      <ShortDescriptions>
        <Description Language="en"
          Description="Withdraw"/>
      </ShortDescriptions>
    </Descriptions>

```

```
</DisplayAttribute>
```

- (Optional): Include custom attributes in the view for each item to assist the decision making process. Refer to figure in Section 2.6 and technical details in Section 7.3.

The following steps describe how to configure the UWL to display decision options from items of the generic user decision type "TS00008267" in your own view "myRejectedRequestView"

- Open your existing configuration XML file, or create a new one as described in 5.2.

- Add a new item type within the <ItemTypes> tag with a unique type name:

```
<ItemType name="uwl.task.webflow.decision.TS00008267"
connector="WebFlowConnector" defaultView="myRejectedRequestView">
</ItemType>
```

- Map the new type to a backend task id (which is the external type) by specifying the item type criteria within the <ItemType>:

```
<ItemTypeCriteria
externalType="TS00008267" connector="WebFlowConnector" />
```

Do not specify the systemid attribute. It is recommended to keep the configuration independent of a specific system. The system, to which the configuration should apply to, is better specified in the upload page ("Adapt to system"), see 3.7.

Note for SP2: The systemid need to be specified since the "Adapt to system" is only available as of SP3 and higher. It is also required to add the systemid to the type name, e.g. "uwl.task.webflow.G9CCLNT001\_TS91100019",

- Define view to include the display of the decision options as check boxes

```
<Views>
<View name="myRejectedRequestView"
width="98%"
supportedItemTypes="uwl.task.webflow.TS00008267"
columnOrder="attachmentCount, detailIcon, subject, priority,
creatorId, createdAt, reviseCol, dropCol"
sortBy="priority"
selectionMode="SINGLESELECT"
tableDesign="STANDARD"
visibleRowCount="10"
headerVisible="yes"
tableNavigationFooterVisible="yes"
tableNavigationType="CUSTOMNAV"
```



```

actionRef="">
<Descriptions default="Rejected Requests">
  <ShortDescriptions>
    <Description Language="en"
      Description="Rejected Requests" />
  </ShortDescriptions>
</Descriptions>
<DisplayAttributes>
  <DisplayAttribute name="reviseCol" type="checkbox"
    width="" actionRef="1" vAlign="TOP"
    hAlign="CENTER">
    <Descriptions default="">
      <ShortDescriptions>
        <Description Language="en" Description="Revise" />
      </ShortDescriptions>
      <LongDescriptions>
        <Description Language="en"
          Description="Revise the request" />
      </LongDescriptions>
    </Descriptions>
  </DisplayAttribute>
  <DisplayAttribute name="dropCol" type="checkbox" width=""
    actionRef="2" vAlign="TOP"
    hAlign="CENTER">
    <Descriptions default="">
      <ShortDescriptions>
        <Description Language="en"
          Description="Withdraw" />
      </ShortDescriptions>
      <LongDescriptions>
        <Description Language="en"
          Description="Withdraw from the request" />
      </LongDescriptions>
    </Descriptions>
  </DisplayAttribute>
</DisplayAttributes>
.....other definitions go here

<Actions>
  <!-- the user decisions are for rendering in the column, not
in the tool bar. Option Revise has internal value 1 and option Withdraw

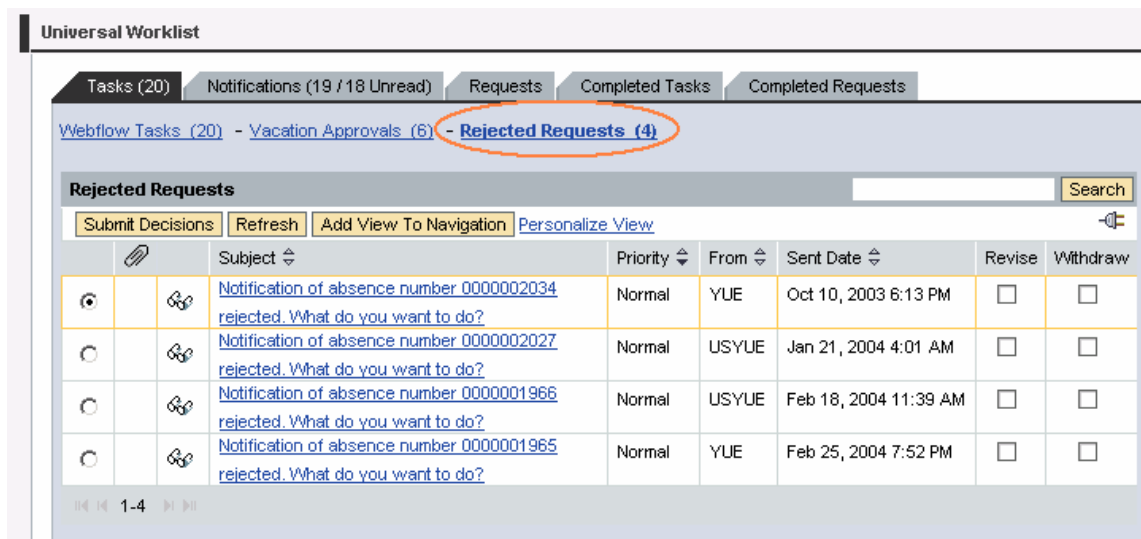
```

has internal value 2 and both are to be handled by the *UserDecisonHandler*->

```

<Action name="1" userDecision="yes"
      handler="UserDecisionHandler" />
<Action name="2" userDecision="yes"
      handler="UserDecisionHandler" />
<!-- the following referenced actions are defined in
uwl.standard -->
<Action reference="submitUserDecisions" />
.....other definitions go here
</Actions>
</View>
</Views>
    
```

- Save the xml definition and verify the validity in Internet Explorer .
- Upload the configuration as described in 3.7 “Uploading Configuration Files”.
- Create a work item of this task for a user, and go to UWL page. A link of the newly defined view ‘**Rejected Requests**’ should be visible. Click on the link and see something like this:



## 8.2 Terminating events

- Definite terminating events can also be displayed for users to pick as user decision options, e.g. Approve or Reject, and then complete the item. In Business Workflow, there is this section in task definition that user can create terminating event(s) .
- This can be specified in the configuration file to enable the options in the view with extra columns (refer to Section 4.4.1 for general concept of adding columns) when the terminating event id is known.
- The main update is the definition under the itemtype for such task and your customized view (<View> section), refer to previous section 8.1. Details in the following table.
- (Optional): Include custom attributes in the view for each item to assist the decision making process. Refer to figure in Section 2.6 and technical details in Section 7.3.

The following steps describe how to configure the UWL to include terminating events *Approved* and *Rejected* as *user decision* and display these 2 options as *checkboxes* in the view for task TS91100019

- Open your existing configuration XML file, or create a new one as described in 5.2.
- Add a new item type within the <ItemTypes> tag with a unique type name:

```
<ItemType name="uwl.task.webflow.TS91100019"  
connector="WebFlowConnector" defaultView="myApprovalRejectView_TE">  
</ItemType>
```

- Map the new type to a backend task id (which is the external type) by specifying the item type criteria within the <ItemType>:

```
<ItemTypeCriteria  
externalType="TS91100019" connector="WebFlowConnector" />
```

Do not specify the systemid attribute.

- Include the action definition for the <ItemType> with the precise event id as action name, add the definition accordingly within the <Action> block:

```

<ItemTypes>
  <ItemType name="uwl.task.webflow.TS91100019"
            connector="WebFlowConnector"
            .....other definitions go here>
  ...
  <Actions>
    <Action name="approved" userDecision="yes"
            handler="TerminatingEventHandler">
      <Descriptions default="Approve"/>
    </Action>
    <Action name="rejected" userDecision="yes"
            handler="TerminatingEventHandler">
      <Descriptions default="Reject"/>
    </Action>
  </Actions>
  .....other definitions go here>
</ItemType>
....
</ItemTypes>

```

- Define view to include the display of the decision options as check boxes

```

<Views>
  <View name=" myApprovalRejectView_TE"
        width="98%"
        supportedItemTypes="uwl.task.webflow.TS91100019"
        columnOrder="attachmentCount, detailIcon, subject, priority,
creatorId, createdAt, rejectCol, approveCol"
        sortBy="priority"
        selectionMode="SINGLESELECT"
        tableDesign="STANDARD"
        visibleRowCount="10"
        headerVisible="yes"
        tableNavigationFooterVisible="yes"
        tableNavigationType="CUSTOMNAV"
        actionRef="">
    <Descriptions default="My Reject Approval_TE">

```

```

        <ShortDescriptions>
        <Description Language="en"
        Description=" My Reject Approval_TE" />
        </ShortDescriptions>
    </Descriptions>
    <DisplayAttributes>
        <DisplayAttribute name="rejectCol" type="checkbox"
            width="" actionRef="rejected"
            vAlign="TOP" hAlign="CENTER">
            <Descriptions default="">
                <ShortDescriptions>
                <Description Language="en" Description="Reject" />
                </ShortDescriptions>
            </Descriptions>
        </DisplayAttribute>
        <DisplayAttribute name="approveCol" type="checkbox"
            width="" actionRef="approved"
            vAlign="TOP" hAlign="CENTER">
            <Descriptions default="">
                <ShortDescriptions>
                <Description Language="en" Description="Approve" />
                </ShortDescriptions>
            </Descriptions>
        </DisplayAttribute>
    </DisplayAttributes>
    .....other definitions go here

    <!-- the user decisions are for rendering in the column.
    Option Approve and option Reject are both to be handled by the
    TerminatingEventHandler -->
    <Actions>
        <Action name="approved" userDecision="yes"
            handler="TerminatingEventHandler">
            <Descriptions default="Approve" />
        </Action>
        <Action name="rejected" userDecision="yes"
            handler="TerminatingEventHandler">
            <Descriptions default="Reject" />
        </Action>
    <!-- the following referenced actions are defined in
    uwl.standard -->
    <Action reference="submitUserDecisions" />

```

```

.....other definitions go here
    </View>
</Views>

```

- Save the xml definition and verify the validity in Internet Explorer .
- Upload the configuration as described in 3.7 “Uploading Configuration Files”.
- Create a work item of this task for a user, and go to UWL page. A link of the newly defined view ‘My Reject Approval\_TE’ should be visible. Click on the link and view similar to figure in Section 2.6 should be displayed (depends on your configuration, with or without custom attributes)

-

### 8.3 Work item completion with container update

- A special handler in UWL to simply complete the task item and update the webflow container with the specified data in the configuration file under the <ItemType> section defined for the item.
- No particular prerequisite in Business Workflow required.
- The main update is the definition under the itemtype for such task and your customized view (<View> section), refer to previous section 8.1. Details in the following table.

The following steps describe how to configure the UWL to include options *Option1* and *Option2* as *user decision* and display them as *checkboxes* in the view. Each option, implicitly, will update the webflow container accordingly and complete the item of type TS91100019

- Open your existing configuration XML file, or create a new one as described in 5.2.
- Add a new item type within the <ItemTypes> tag with a unique type name:

```

<ItemType name="uwl.task.webflow.TS91100019"
connector="WebFlowConnector" defaultView="webflowView">
</ItemType>

```

- Map the new type to a backend task id (which is the external type) by specifying the item type criteria within the <ItemType>:

```

<ItemTypeCriteria
externalType="TS91100019" connector="WebFlowConnector" />

```

Do not specify the systemid attribute.

- Specify the action handler for the <ItemType> as 'UpdatingContainerHandler'.

```
<ItemTypes>
  <ItemType name="uwl.task.webflow.TS91100019"
            connector="WebFlowConnector"
            .....other definitions go here>
  <ItemTypeCriteria connector="WebFlowConnector"
                  externalType="TS91100019" />
  <Actions>
    <Action name="option1" userDecision="yes"
           handler="UpdatingContainerHandler" >
      <Descriptions default="Option1" />
      <Properties>
        <Property name="param1" value="value1" />
        <Property name="param2" value="value2" />
        <Property name="param3" value="value3" />
      </Properties>
    </Action>
    <Action name="option2" userDecision="yes"
           handler="UpdatingContainerHandler" >
      <Descriptions default="Option2" />
      <Properties>
        <Property name="param4" value="value4" />
      </Properties>
    </Action>
  </Actions>
  .....other definitions go here>
</ItemType>
...
</ItemTypes>
```

```

<Views>
  <View name=" myCompleteItemView_CU"
    width="98%"
    supportedItemTypes="uwl.task.webflow.TS91100019"
    columnOrder="attachmentCount, detailIcon, subject, priority,
creatorId, createdAt, option1Col, option2Col"
    sortBy="priority"
    selectionMode="SINGLESELECT"
    tableDesign="STANDARD"
    visibleRowCount="10"
    headerVisible="yes"
    tableNavigationFooterVisible="yes"
    tableNavigationType="CUSTOMNAV"
    actionRef="">

    <Descriptions default="My Simple Complete View">
      <ShortDescriptions>
        <Description Language="en"
          Description="My Simple Complete View"/>
      </ShortDescriptions>
    </Descriptions>

    <DisplayAttributes>
      <DisplayAttribute name="option1Col" type="checkbox"
        width="" actionRef="option1"
        vAlign="TOP" hAlign="CENTER">
        <Descriptions default="">
          <ShortDescriptions>
            <Description Language="en" Description="Option1"/>
          </ShortDescriptions>
        </Descriptions>
      </DisplayAttribute>
      <DisplayAttribute name="option2Col" type="checkbox"
        width="" actionRef="option2"
        vAlign="TOP" hAlign="CENTER">
        <Descriptions default="">
          <ShortDescriptions>
            <Description Language="en" Description="Option2"/>
          </ShortDescriptions>
        </Descriptions>
      </DisplayAttribute>

```



```

        </DisplayAttributes>
.....other definitions go here

        <!-- the user decisions are for rendering in the column.
Option Option1 and option Option2 are both to be handled by the
UpdatingContainerHandler -- >
        <Actions>
        <Action name="option1" userDecision="yes"
            handler="UpdatingContainerHandler" >
            <Descriptions default="Option1"/>
        <Properties>
            <Property name="param1" value="value1" />
            <Property name="param2" value="value2" />
            <Property name="param3" value="value3" />
        </Properties>
        </Action>
        <Action name="option2" userDecision="yes"
            handler="UpdatingContainerHandler" >
            <Descriptions default="Option2"/>
        <Properties>
            <Property name="param4" value="value4" />
        </Properties>
    </Action>

        <!-- the following referenced actions are defined in
uwl.standard -->
        <Action reference="submitUserDecisions" />

    <!-- other definition goes here -- >

```

- Save the xml definition and verify the validity in Internet Explorer .
- Upload the configuration as described in 3.7 "Uploading Configuration Files".
- After completing the item in UWL, verify the webflow container content in the SAP Webflow that it is updated with the specified values.

-

## 9 Step By Step – Tailored UWL iView

Applications and administrators can provide UWL iView instances, which deliver information to the user, which is tailored to his role. This is achieved by copying SAP shipped Universal Worklist iView to the user's workset in the portal administration and setting the iView properties appropriately.

### 9.1 Prerequisites

As an illustration add the View created and uploaded in section 7, that is "myApprovalView" to portal user's home page and switch off displaying preview and navigations. By doing this user would see only vacation approvals in a separate iView under *Home* → *MyInfo* tab.

In order to achieve this you must be having

- myApprovalView already loaded to UWL configuration
- myApprovalView should be working correctly

### 9.2 Creating a new UWL iView

1. Login to portal as administrator. Navigate to *Content Administration* → *Portal Content*. Go all the way to *Portal Content* → *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *Pages*
2. Right click and select new Page, enter page name say mypage and other properties, choose appropriate layouts and save the page
3. Expand Portal Content. Go all the way to *Portal Content* → *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *iViews*, right click on *Universal Worklist* and choose *addiViewtoPage* → *copy*
4. Save the page
5. In the Page editor, choose the *Universal Worklist iView*, click on *Properties*. In the Properties list on the right hand side, scroll down to *Display Preview*, select *false*. Scroll down and enter *sap\_uwl\_viewname* as "myApprovalView", choose *UWL Navigation Style* as none, Change the title if needed, for example; "My Vacation Approvals"
6. Click *Save*

### 9.3 Adding UWL iView to Portal Users home page

1. Login to portal as administrator. Navigate to *Content Administration* → *Portal Content*
2. Expand the Portal Content Directory to *Portal Content* → *Portal Users* → *Standard Portal Users*
3. Right click on “Standard user role” workset and select “Edit”. Now Standard Users workset should be visible in the right side. Click on *MyInfo*, *MyInfo* should now be highlighted.
4. Without closing the workset, expand *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *Pages*, locate the page created in section 9.2, Right click on the *myPage* and choose *addPageToRole* → *Copy*
5. This adds the custom Iview created to the end users worklist

#### 9.4 Verify the Custom IView

Log on to the portal as a Portal end user. Navigate to *Home* → *MyInfo*.

*myPage* that is added to the workset should now be visible. Click on *myPage*, only *Vacation Approvals* should be visible now.

#### 9.5 Further tips on layout

Layout of the page for example width etc. can be changed anytime by changing the *myPage* page under *Portal Content* → *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *Pages*.

If any of UWL IView properties need changes, it can be changed by editing the properties of IView in the page content of *myPage* located under expand *Content Provided by SAP* → *End User Content* → *Standard Portal Users* → *Pages* → *MyPage*.

## 10 Appendix

### 10.1 Configuration XML Tag Documentation

To be released.

SAP internal, see (draft to be revised and updated):

[http://aiokeh.wdf.sap.corp:1080/SAPIKS/~S~a97d5ef5456643d9bb975eaac94e54fe/KW/IWB\\_STRU\\_CT~0DDBBE61E8D04266BAF21D8E8BC95CB4/IWB\\_EXTHLP~D269C045994448ADAD27D36C4154FE74/?LANGUAGE=EN&RELEASE=646&IWB\\_INDUSTRY=&IWB\\_COUNTRY=&TMP\\_IWB\\_TASK=PREVIEW2](http://aiokeh.wdf.sap.corp:1080/SAPIKS/~S~a97d5ef5456643d9bb975eaac94e54fe/KW/IWB_STRU_CT~0DDBBE61E8D04266BAF21D8E8BC95CB4/IWB_EXTHLP~D269C045994448ADAD27D36C4154FE74/?LANGUAGE=EN&RELEASE=646&IWB_INDUSTRY=&IWB_COUNTRY=&TMP_IWB_TASK=PREVIEW2)

### 10.2 UWL Item Launch Configuration / UWL Actions

Navigation	Link
Navigation from <a href="http://help.sap.com">http://help.sap.com</a> : SAP Netweaver → SAP Enterprise Portal Click on the language. Under <i>SAP EP Documentation</i> , choose: <i>Administration Guide</i> → <i>Cross Platform</i> → <i>Default Initial Content</i> → <i>Default Portal Content</i> → <i>Default Iviews</i> → <i>Universal Worklist</i>	<a href="http://help.sap.com/saphelp_ep60sp2/helpdata/en/fff/83fd4d5a644336b13ae54528cfef57/content.htm">http://help.sap.com/saphelp_ep60sp2/helpdata/en/fff/83fd4d5a644336b13ae54528cfef57/content.htm</a>

### 10.3 API for Registering UWL XML Configuration

Follow the steps below to register a configuration XML file to UWL from a business package. Note: registration has to be done only once, it should not be done in every request cycle. It is recommended to place the following code example in a static initialization method.

1. Verify that xml adheres to the `uwl_configuration.dtd` document type definition.
2. Reference Universal Worklist service in `portalapp.xml`.
3. Check the sample code snippet. Code snippet for registering:

```
import com.sap.netweaver.bc.uwl.*;
import com.sap.netweaver.bc.uwl.config.*;
import com.sap.netweaver.bc.uwl.admin.IUWLConfigurationManager;

/////Get Hold of UWL Service

static {
    IUWLService uwlService = (IUWLService)
PortalRuntime.getRuntimeResources().getService(IUWLService.ALIAS_KEY);
    IUser user = request.getUser();
    userContext = new UWLContext();
    userContext.setUser(user);

    /////create a UWL session
    uwlService.beginSession(userContext, 10);
}
```

```

        IUWLConfigurationManager cm =
uwlService.getConfigurationManager(userContext);
        String fileName=" ;///fileName should contain the full path to
the xml configuration file
        String configurationName=;///Fill the Configuration name

        File file = new File(fileName);
        try {
            /// check if previously registration was done
            if(false==
cm.getConfigAdmin(userContext).isConfigurationRegistered(configurationN
ame)){

cm.getConfigAdmin(userContext).registerConfigurationFromXml(userContext
, configurationName,new
        FileInputStream(file),ConfigurationRoot.IConfigurationOwner.CUS
TOMER);
            }
        } catch (Exception e) {
            ///log the exception or handle the error accordingly
        }
        uwlService.endSession(userContext);
    }
}

```

## 10.4 Implementing Custom iViews for Business Workflow Item Execution

Suppose a Business Workflow is defined in a webflow system. The logic to trigger the execution of the workflow and communicating the result back to the user can be implemented as an iView. The UI of the iView should display the information relevant to the work item, created by the workflow, as well as UI elements (such as buttons) to allow the user to create an event, which will be handled in the `onAction(Event EventName)` event handler of the iView implementation code. Once such an iView is created, the task id of that workflow can be mapped to the iView (see 5.3).

The sample iView below is created for the approval vacation workflow, defined with 2 terminating events: approved and rejected. The UI displays a form with the data about the requested vacation requested and 2 buttons triggering terminating events set for the workflow (see the UI in 5.3.8). To display the form data the demo iView has to do the following:

- receive the current work item id from uwl (method `createFormBean()` below)
- call FM **SAP\_WAPI\_READ\_CONTAINER** to get the work item context (method `createForm()` below)
  - in table `simple_container` there's an entry `_WI_OBJECT_ID` containing the object's POR
  - take the key out of the POR (structure is `SWOTOBJID` as defined in DDIC)
- call FM **ZFORMABSENC\_READ** to fetch the object's data. Use the object key as `formnumber`. The object's data are stored in `DemoFormBean` (method `createForm()` below)
- render form data and approve / reject buttons

After the user clicks on “Approve” or “Reject” button the `onAction(Event)` method of `iView` implementation is called (below). That method determines what button was clicked, so that the correct event is communicated to the webflow system. The FM `SAP_WAPI_CREATE_EVENT` is called to complete the work item (method `createEvent()` below) .

```

public class DemoIview extends PageProcessorComponent {

    public DynPage getPage() {
        return new DemoPage();
    }

    public class DemoPage extends JSPDynPage {
.....
        private void createForm() {
            .....
            IFunctionTemplate m_read_container =
repository.getFunctionTemplate("SAP_WAPI_READ_CONTAINER");
            JCO.Function function = new JCO.Function(m_read_container);
            JCO.ParameterList importList =
function.getImportParameterList();
            importList.setValue(workitemID, "WORKITEM_ID");

            // call the function module
            client.execute(function);

            .....
            // get the formID;
            JCO.Table cnt =
function.getTableParameterList().getTable("SIMPLE_CONTAINER");
            String formNumber = "";
            for (int i =0; i < cnt.getNumRows(); i++) {
                cnt.setRow(i);

                if(cnt.getString("ELEMENT").equals("_WI_OBJECT_ID"))
                    String formNumber = cnt.getString("VALUE");

.....
            }
            // get the form data
            IFunctionTemplate m_read_form =
repository.getFunctionTemplate("ZFORMABSENC_READ");

            importList = function.getImportParameterList();
            importList.setValue(formNumber, "FORMNUMBER");
            client.execute(function);

            // get the result
            outputList = function.getExportParameterList();
            .....

            // populate the FormBean with the form data and display the
data with the buttons, triggering the terminating events (approved and
rejected)
.....
        }
    }
}

```

```

    }

    private void onAction(Event myName) throws PageException {
        String event = myName.getComponentName();
        createEvent(event);
    }

    private void createEvent(String event) {
        .....
        IFunctionTemplate m_create_event=
repository.getFunctionTemplate("SAP_WAPI_CREATE_EVENT");

        JCO.Function function = new JCO.Function(m_create_event);
        JCO.ParameterList importList =
function.getImportParameterList();
        importList.setValue("FORMABSENC", "OBJECT_TYPE");
        importList.setValue(formNumber, "OBJECT_KEY");
        importList.setValue(event, "EVENT");
        importList.setValue("X", "COMMIT_WORK");

        // call the function module
        client.execute(function);
        .....
        // check for errors and display the result
    }

    private void createFormBean() {
        String system = request.getParameter("logsys");
        String workitemID = request.getParameter("wi_id");
        String destination = request.getParameter("destination");
        String formnumber = request.getParameter("formnumber");
        bean = new DemoFormBean(destination, system, workitemID,
formnumber);
    }
}
}

```

## 10.5 Item visualization types in Business Workflow

A workflow definition is normally made up of a series of tasks. In Business Workflow, when dialog with a user is required to execute a task, this type of task is called the 'dialog work item'. In general, a task is defined with a business object type and its associated method and this determines the visualization type that the user will see when executing the work item.

(Business object type is defined with the Business Object Builder in the system and a full directory of the object types can be found in the Business Object Directory).

There are mainly 5 types:

- Standard business object ABAP form

- Vacation request/approval form, user decision, ...
- Eg. Type: "FORMABSENC"; Method: "CREATE"/"APPROVE"
- Eg. Type: "DECISION"; Method: "PROCESS" (check out the standard task "TS00008267")
- External Service
  - Type: "EXTSRV"; Method: "PROCESS"
- Web Service
  - Type: "WEBSERVICE"; Method: "PROCESSDIALOG"
- Web Form
  - Type: "FORM"; Method: "HTMLPROCESS"
  - A Web-enabled SAP transaction ("Web transaction") is called using the Internet Transaction Server (ITS)
- Business object associated with BSP blueprint
  - Business Partner, Product, Material
  - A business server page (BSP) will be called and displayed in the browser by the Web Application Server (WAS)
  - Eg. Business Partner → Type: "BUS1006"; Method: "CREATE/CHANGE/DISPLAY"
  - Product → Type: "BUS1178"; Method: "CREATE/CHANGE/DISPLAY"

In addition to the specification done during task definition, transaction **SWFVISU** is introduced (as of 6.20 sp19) to support a few more visualization types with corresponding parameters, particularly for portal:

- Portal component
  - Eg. Parameter (required): "NAME"; Value: "<full path of the portal component>"
- Portal IView
  - Eg. Parameter (required): "ID"; Value: "<full path of the iview>"
- Business object associated with BSP blueprint
  - Eg. Parameter (optional): "CRM\_OTYPE"; Value: "PRODUCT"
  - Eg. Parameter (optional): "CRM\_METHOD"; Value: "DEFAULT"
  - (The 2 parameters are optional, but "CRM\_METHOD" is expected together with the existence of "CRM\_OTYPE")
- WebDynpro application
  - Eg. Parameter (required): "APPLICATION"; Value: "WebDynproConsole"
  - Eg. Parameter (required): "PACKAGE"; Value: "sap.com/tc~wd~tools"

More information about this transaction can be found at the webflow documentation in:

[http://help.sap.com/saphelp\\_nw04/helpdata/en/a5/172437130e0d09e1000009b38f839/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/a5/172437130e0d09e1000009b38f839/frameset.htm)

Navigation from <http://help.sap.com/nw4> :

Click on the language. Under *SAP Library*, choose:

*SAP NetWeaver* → *Application Platform* → *Business Services* → *SAP Business Workflow* → *Select Reference Documentation* → *Workflow System Administration* → *Parameters for Work Item Visualization in the UWL*



[www.sap.com/netweaver](http://www.sap.com/netweaver)