**Picker as input for IBA**



Author:

Aman Somgade

Document Properties

|  |  |
| --- | --- |
| **File Name** | **Status** |
| Picker as input parameter for an IBA | Draft |

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Description** |
| 12-02-2015 | Aman Somgade | A.1 | Initial Draft Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Review History

|  |  |  |
| --- | --- | --- |
| **Reviewer** | **Review Date** | **Comments** |
| Dattupant Telang |  |  |
| Mayuresh Joshi |  |  |
|  |  |  |

Approval History

|  |  |  |
| --- | --- | --- |
| **Approver** | **Approval Date** | **Comments** |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Trademark Acknowledgement** |

Copyright© 2009 ITC InfoTech

All products or brand names used in this document are trademarks or registered trademarks of their respective companies. The information in this document is subject to change without notice.

ITC InfoTech shall not be liable for errors contained herein, or for consequential damages in connection with the furnishing, performance or use of the material. No part of this document may be reproduced or transmitted in any form or by any means, for any purpose, or translated to another language without the prior written consent of ITC InfoTech.

|  |
| --- |
| **Confidentiality Statement** |

By accepting this document the recipient agrees:

To keep permanently confidential all information which it contains.

To treat and retain as secret and confidential all information contained in this document or otherwise acquired by the receiving party from ITC InfoTech including, without prejudice to the generality of the foregoing, all handbooks, manuals, drawings, designs, specifications, charts, diagrams, tapes, disks, diskettes and any other documents or materials containing such information.

This may not be construed as legally valid and binding agreement between ITC InfoTech and the recipient of this document. If accepted, an agreement will be formalized incorporating mutually agreed terms and conditions.

Contents

[1. Overview 4](#_Toc405477334)

[2. To-Be Process 5](#_Toc405477335)

[3. Windchill Process 6](#_Toc405477336)

[4. Deployment Steps with Build 7](#_Toc405477337)

1. Overview

There are many types of business objects in Windchill, plus potentially many soft types or

modelled object types introduced by the user. These types would predominantly would have some

relationship with the other types. There are numerous instances when user creates or updates

objects of such types using an user interface, he would need to specify the property value(s)

derived from the other available objects of the same or different type. E.g. The business object

like Part or Document are owned by a specific organization and hence at the time of their

creation user needs to be able to specify an owning organization.

This sort of user interaction is achieved through the use of **property picker**. The property picker

simply aids in picking up a property from an existing business object. The user would pick an

object by launching the picker and have its properties used to update visible and/or nonvisible

properties on the page. The following sketch illustrates this sort of interaction pictorially.

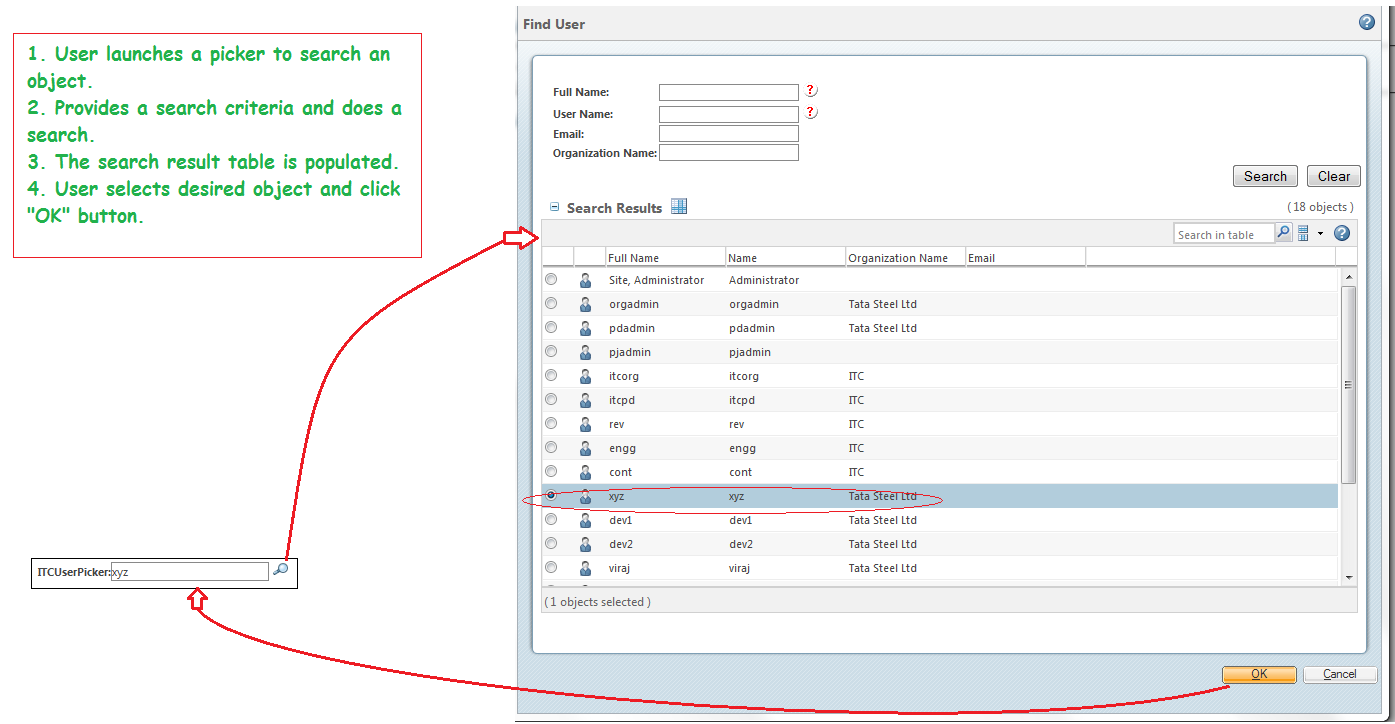
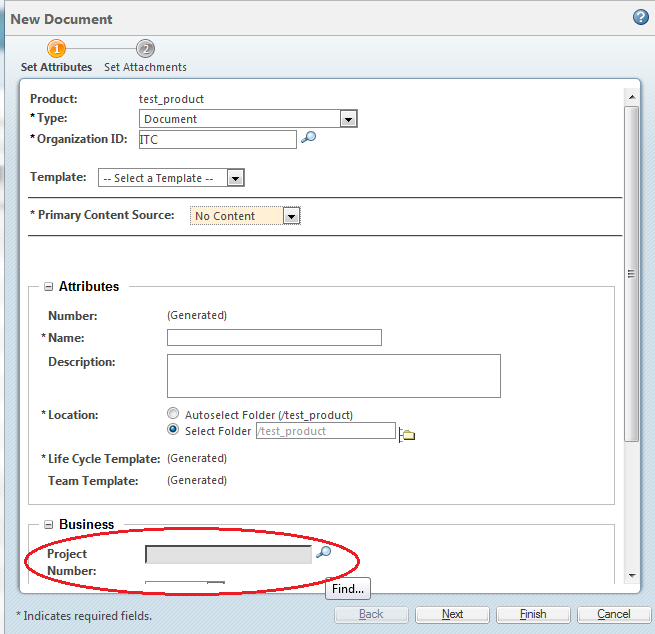


figure: In this example: a user picker is trigerred (left window) – the picker (right window), searchs all records, the picked row values are returned to the calling wixard step.

1. To-Be Process/Requirement

**Requirement:** Wewant to create a project picker with custom search criteria which is an input parameter for an IBA on create page.

There is an IBA for WTDocument named as “Project Number”. Whenever a new document is created we need to provide a value to this IBA. But, it will be very tedious for the user to either remember the project number or browse to the project and then view project number. So, we want to provide a picker adjacent to IBA, upon clicking it, a user can search for existing projects and select any project to directly fetch its number to our IBA. Additionally, we also need to customize the search criteria in this picker as per user convenience and requirement.



1. Windchill Process/Solution

**3.1 General information about picker:-**

Picker can be configured in two ways, one is jsp approach and other is coding approach.

**3.1.1 Picker using tags:-**

In windchill, there is readied system of OOTB Pickers which can be used on any jsp page simply by using some tags. Following is a list of such picker types:

1. **User picker:** The user picker is used when you have a requirement to select specific user(s) depending upon certain criteria and use them in your application. Typical use case could be that you may want to find parts which are created by certain user.In this case you can have a user picker and then through this you can selectthe user and pass it on to the search criteria to perform search for parts.

User Picker <wctags:userPicker id="testUserPicker" label="MyUserPicker" />

1. **Organization picker:** The organization picker is used when you have a requirement to select specific organization(s) depending upon certain criteria and use them in your application. Typical use case could be that you may want to find parts that are available in particular organization. In this case, you can have a organization picker and then through this you can select the organization and pass it on to the search criteria to perform search for parts.

Organization Picker <wctags:organizationPicker id="orgPicker" label="MyOrgPicker"/>

1. **Context picker:** The context picker is used when you have a requirement to perform an operation that is based on certain context.

Context Picker <wctags:contextPickerid="contextPicker"label="MyContextPicker"pickerTitle="ContextPicker" />

1. **Item picker:** The item picker is used when you have a requirement to select specific business object(s) depending upon certain criteria and use them in your application.

Item Picker <wctags:itemPicker id="itemPicker" label="MyItemPicker" pickerTitle="ItemPicker"/>

1. **Type picker:** Type Picker Common Component is to be used either for display or for assignment of type-able items. For example in a search application to select the type of objects that you are interested to do a search on or in a create application to create an item of a specific type. It can be used to display the type in case of edit or view applications. The component can be used in the context/mode of CREATE, EDIT, SEARCH or VIEW.

Type Picker in Drop Down format

<p:typePicker id="typePickerTest" label="Type : " mode="SEARCH" > <p:pickerParam name="format" value="dropdown" /> </p:typePicker>

Type Picker in table format

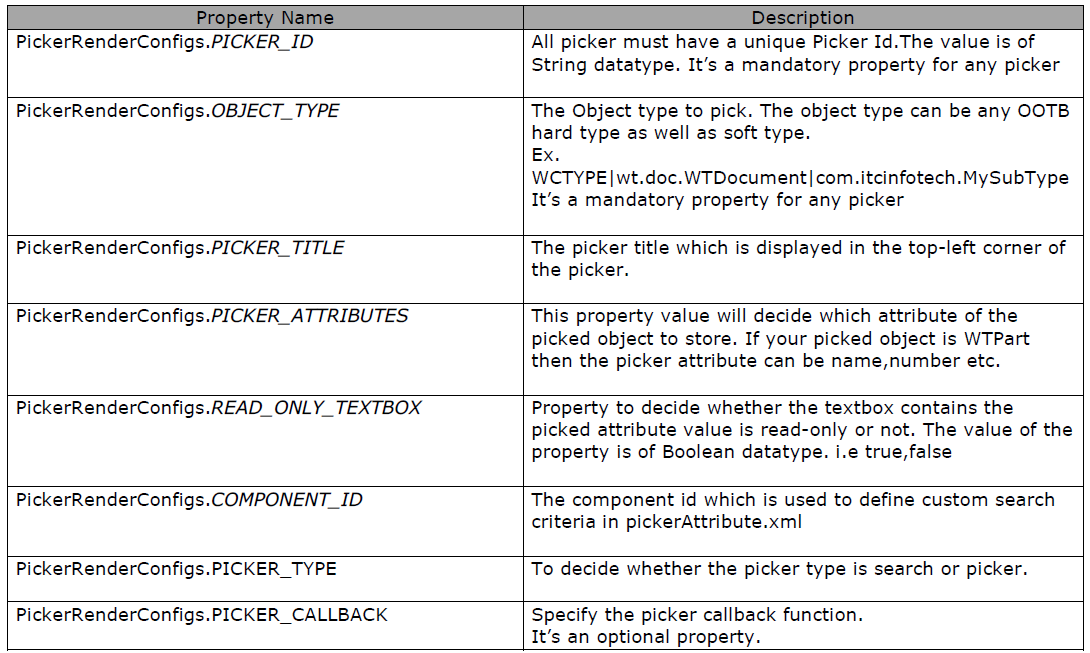
<p:typePicker id="typePickerTest" label="Type : " mode="SEARCH" > <p:pickerParam name="format" value="table" /> </p:typePicker>

Type Picker in tree format

<p:typePicker id="typePickerTest" label="Type : " mode="SEARCH" > <p:pickerParam name="format" value="tree" /> </p:typePicker>

**3.1.2 Picker using coding approach:-**

An API “PickerRenderConfigs” is used to set various picker properties in java code. The various picker properties which can be set using this API are as follows:



**3.2 Custom search criteria for picker:-**

We can define which attributes we want to search in our picker. To do this, we simply need to insert following line in <WINDCHILL>\codebase\pickerAttributes.xml:

<ComponentID id="TestPicker">

<ObjectType id="wt.part.WTPart">

<SearchCriteriaAttributes>

<Attributes>

<Name>contextRef</Name>

<DisplayName>CONTEXT\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>name</Name>

<DisplayName>NAME\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>number</Name>

<DisplayName>NUMBER\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>state.state</Name>

<DisplayName>STATE\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>thePersistInfo.modifyStamp</Name>

<DisplayName>LAST\_UPDATED\_ATTRIBUTE</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>thePersistInfo.createStamp</Name>

<DisplayName>CREATED\_ATTRIBUTE</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>versionInfo.identifier.versionId</Name>

<DisplayName>VERSION\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>iterationInfo.identifier.iterationId</Name>

<DisplayName>ITEM\_ITERATION\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>iterationInfo.creator</Name>

<DisplayName/>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>folderingInfo.cabinet</Name>

<DisplayName>CABINET\_TABLEVIEW\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>containerInfo.ownerRef</Name>

<DisplayName>OWNER\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

</SearchCriteriaAttributes>

</ObjectType>

</ComponentID>

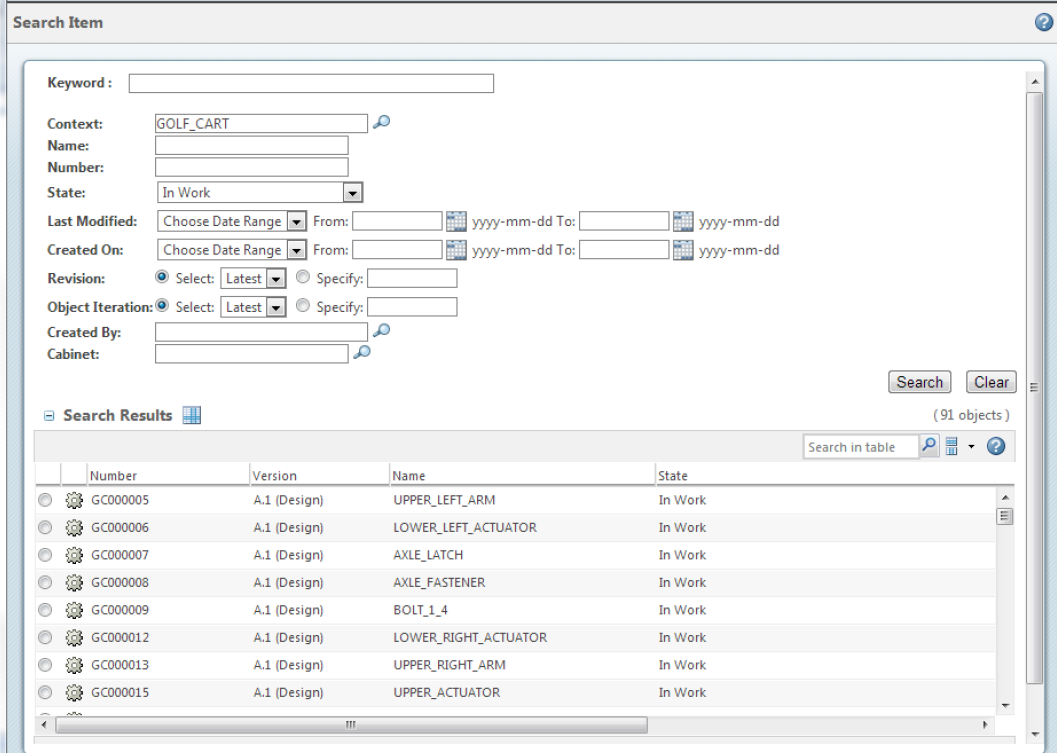


figure: picker input – attributes are defined in pickerAttributes.xml for the componentID = TestPicker – Note: this id will be used in DataUtility class

**3.3 Picker call back function:-**

A JavaScript function is used to handle picked row and put values in calling form (wizard step).

PickerRenderConfigs.setDefaultPickerProperty(map, PickerRenderConfigs.***PICKER\_CALLBACK***, "ProjectnumberCallBack");

The JavaScript function should be defined inside wizard step (for custom step wizard) or in appropriated position for OOTB wizard step.

Example:

<script Language="JavaScript">

function ProjectnumberCallBack(objects, pickerID, attr, displayFieldId ) {

var updateHiddenField = document.getElementsByName(pickerID)[0];

var updateDisplayField = document.getElementsByName(displayFieldId)[0];

var myJSONObjects = objects.pickedObject;

for (var i = 0; i < myJSONObjects.length; i++) {

var oid = myJSONObjects[i].number;

var displayAttr = eval("myJSONObjects[i].number");

updateHiddenField.value = oid;

updateDisplayField.value = displayAttr;

}

}

</script>

Here the number field is returned to the calling form.

1. Deployment Steps with Build

**Step 1:-**

Create a java class “ITCProjectPicker” which is a DataUtility.



And copy the compiled class file in <Windhcill\_home>\codebase\com\practice\DataUtility or any other location as per user preference.

**Step 2:-**

Define the search criteria in the file <Windhcill\_home>\codebase\pickerAttributes.xml and add your own search criteria with the same componentID mentioned in the dataUtility. (In this example it is “ItcPicker”)

<ComponentID id="ItcPicker">

<ObjectType id="wt.projmgmt.admin.Project2">

<SearchCriteriaAttributes>

<Attributes>

<Name>number</Name>

<DisplayName>NUMBER\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

<Attributes>

<Name>name</Name>

<DisplayName>NAME\_LABEL</DisplayName>

<IsSearchable>true</IsSearchable>

</Attributes>

</SearchCriteriaAttributes>

</ObjectType>

</ComponentID>

**Step 3:-**

Define the picker callback function in create.jspf or define it in a separate jsp and include the tag for this jsp in create.jspf. The location for create.jspf is <Windhcill\_home>\codebase\netmarkets\jsp\document\create.jspf. The call back function name should be same in jsp as well as java class. (In this example it is “ProjectnumberCallBack”)

<script Language="JavaScript">

function ProjectnumberCallBack(objects, pickerID, attr, displayFieldId ) {

var updateHiddenField = document.getElementsByName(pickerID)[0];

var updateDisplayField = document.getElementsByName(displayFieldId)[0];

var myJSONObjects = objects.pickedObject;

for (var i = 0; i < myJSONObjects.length; i++) {

var oid = myJSONObjects[i].number;

var displayAttr = eval("myJSONObjects[i].number");

updateHiddenField.value = oid;

updateDisplayField.value = displayAttr;

}

}

</script>

**Step 4:-**

Register the entry of data utility class in services.properties.xconf and run xconfmanager –p. The selector value should be unique. (In this example it is “test1”)

<Service context="default" name="com.ptc.core.components.descriptor.DataUtility">

<Option serviceClass="com.practice.DataUtility. ITCProjectPicker"

selector="test1"

requestor="java.lang.Object"

cardinality="duplicate" />

</Service>

**Step 5:-**

Navigate to type and attribute management> manage types> Select Document> click on Actions> Edit> select Layouts tab>select “Create new layout”> Select IBA (In this example it is “Project number”)> Click on  icon.

Provide the appropriate selector value in the “Data utility ID” field.

