

E-Recruitment: Adding Additional Tab to Requisition Maintenance



Applies to:

SAP E-Recruitment 6.0. E-recruitment Recruiter Scenario developed using BSP technology

Summary

In SAP E-recruitment, requisition maintenance, SAP has provided many standard fields to be used for requisition maintenance. However there are many business specific requirements where customer wants to add a whole new set of fields which are logically grouped together, so to provide flexibility SAP has provided provision to add a complete new additional tab, to display such logically grouped fields together. This document explains how such fields can be added to requisition maintenance, by creating a whole new tab. Apart from this, document will also explain how such fields can be stored in database and made search relevant.

Author: Hemendra Singh Manral

Company: SAP Labs India

Created on: 04 August 2009

Author Bio



Hemendra Singh Manral is a Sr. Consultant with Custom Development, SAP Labs India. He has worked on various customer projects in the area of HCM and E-Recruitment using SAP technologies like BSP, ABAP Web-Dynpro, XML for ABAP, Web services, Enhancement framework etc.

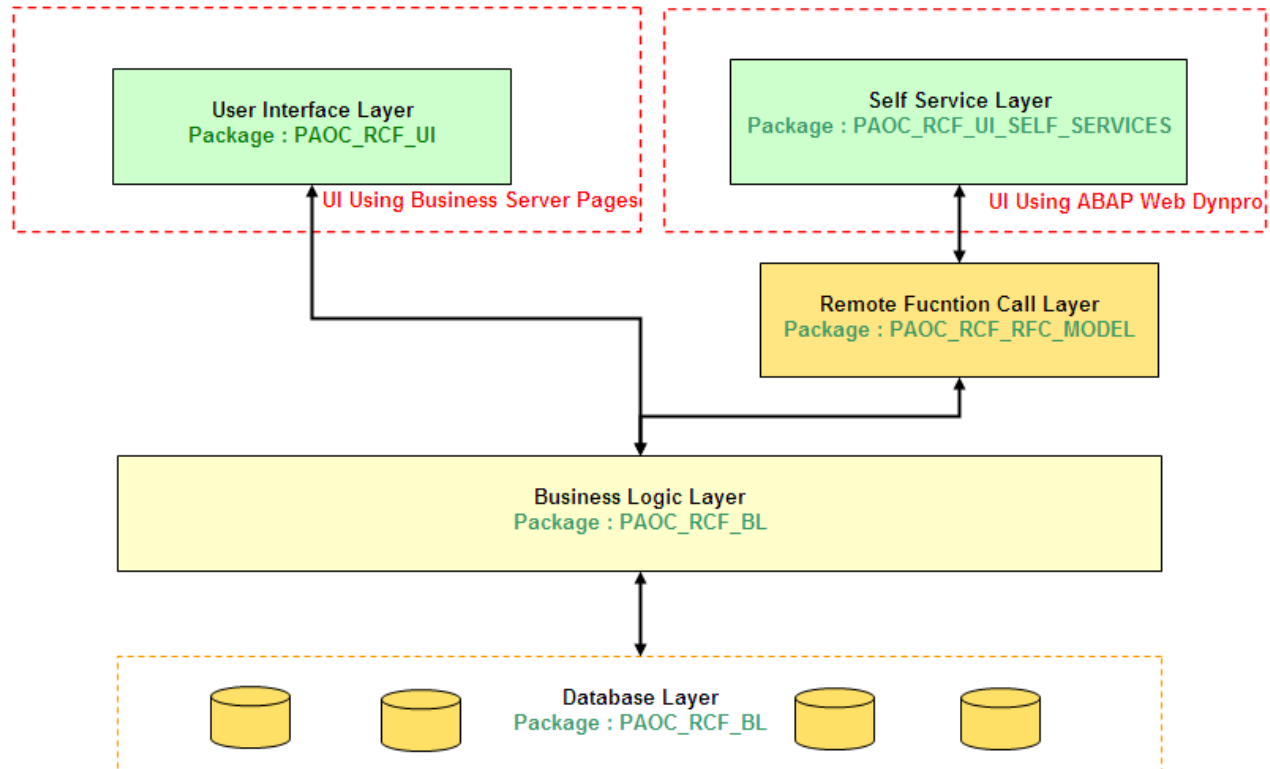
Table of Contents

Introduction	3
Description of Requirement	4
Creating a Custom Infotype	5
Creating a New BSP Application for Rendering New Fields	8
Maintaining Additional Customizing for Custom Fields	10
Customization Required for Adding New Tab.....	10
Field Level Customization for Newly Added Fields.....	12
Making fields Search Relevant	13
Enabling Search Profile Generation of Field.....	13
Enabling Search UI for Field	15
UI Output of Our Requirement.....	16
Related Content.....	16
Copyright.....	17

Introduction

In E-recruitment requisition maintenance, many times, there are requirement to add several additional fields which are logically grouped together. In such cases the need is to have a new infotype to store such fields and create a new Tab on UI for user maintenance.

Before going forward with the document its important to know about different layers in E-Recruitment, so that we can create new objects accordingly, at different layers and set up a good communication link using objects at different layer. Below is a layered architecture of E-Recruitment.



E-Recruitment Layered Architecture

This document will explain how to add additional custom tab to the requisition maintenance. We take an example requirement and frame the document accordingly.

The whole document is broadly categorized into following sections:

1. Description of requirement
2. Creating a custom Infotype
3. Creating a new BSP application for rendering new fields
4. Maintaining additional customizing for custom fields
5. Making fields search relevant
6. UI Output of our requirement

Description of Requirement

Let's say, a customer wants following fields to be added to requisition maintenance by creating a whole new tab. New tab will actually help in logically grouping all such fields on requisition maintenance. Let's say for a complete R&D lab, some additional basic requirements, might be required for some jobs, like as follows:

Field	Field Type
Worked with any government outside native country	Checkbox
Minimum Number of Thesis submitted	Inputfield
Minimum number of papers published	Inputfield
Minimum number of patents filed	Inputfield


Here we will not take much fields, but in actual scenario the count of fields are generally higher. We put the name of new tab as "Additional work detail"

Creating a Custom Infotype

Following are the steps to create a custom infotype.

1. Create a HRI structure

Structure	HRI9501	Active
Short Description	Additional Work Detail	

Attributes	Components	Entry help/check	Currency/quantity fields			
 Predefined Type 1 / 4						
Component	RTy...	Component type	Data Type	Length	Decim...	Short Description
ZZ GOVT WRK	<input type="checkbox"/>	ZDE GOVT WRK	CHAR	1	0	Government Work
ZZ MIN THESIS	<input type="checkbox"/>	ZDE MIN THESIS	NUMC	3	0	Min. no. of Thesis Submitted
ZZ MIN PAPER	<input type="checkbox"/>	ZDE MIN PAPER	NUMC	3	0	Min. no. of Papers Published
ZZ MIN PATENT	<input type="checkbox"/>	ZDE MIN PATENT	NUMC	3	0	Min. no. of Patents Filed

2. Go to Transaction PPCJ and create new HRP infotype.

Create infotype

Infotype	
Infotype	9501 <input type="button" value="Create"/> <input type="button" value="Check"/>
Infotype Name	Additional Work Detail
Namespace	

Infotype category	
<input checked="" type="radio"/> Field infotype	<input type="checkbox"/> Lang.-Dependent Infotype
<input type="radio"/> Table infotype	<input type="checkbox"/> Country-Specific Infotype

Conversion of Table TLOG	
Transparent table	<input type="button" value="Create"/>

Create infotype

Infotype

Infotype	9501	<input type="button" value="Create"/>	<input type="button" value="Check"/>
Infotype Name	Additional Work Detail		
Namespace			

Infotype category

Field infotype
 Table infotype

Conversion of Table TLOG
Transparent table

Create Infotype

Create infotype 9501?

Category




Infotype Lang.-Dependent Infotype
Infotype Country-Specific Infotype

Conversion of Table TLOG
Transparent table

Information


Infotype 9501 created

Newly Created infotype:

Transp. Table	HRP9501	Active					
Short Description	DB Table for Infotype 9501						
Attributes Delivery and Maintenance Fields Entry help/check Currency/Quantity Fields							
   Srch Help Predefined Type							
Field	Key	Initi	Data element	Data Ty...	Length	Decim...	Short Description
<u>.INCLUDE</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	HRIKEY	STRU	0	0	Extended Key Definition of PD Infotypes
<u>MANDT</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
<u>PLVAR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PLVAR	CHAR	2	0	Plan Version
<u>QTYPE</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	QTYPE	CHAR	2	0	Object Type
<u>OBJID</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	HROBJID	NUMC	8	0	Object ID
<u>SUBTY</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SUBTYP	CHAR	4	0	Subtype
<u>ISTAT</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ISTAT_D	CHAR	1	0	Planning Status
<u>BEGDA</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	BEGDATUM	DATS	8	0	Start Date
<u>ENDDA</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ENDDATUM	DATS	8	0	End Date
<u>VARYF</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	VARYF	CHAR	10	0	Variation Field for File PLOG
<u>SEQNR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SEQNR	NUMC	3	0	Number of Infotype Record with Same Key
<u>INFTY</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	INFOTYP	CHAR	4	0	Infotype
<u>QTJID</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	QTJID	CHAR	10	0	Concatenation of Object Type and Object ID
<u>.INCLUDE</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	HRIADMIN	STRU	0	0	Infotype Administration Data
<u>AEDTM</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AEDTM	DATS	8	0	Changed on
<u>UNAME</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	USRNAME	CHAR	12	0	User Name
<u>REASN</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	REASN	CHAR	2	0	Reason
<u>HISTO</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	HISTO	CHAR	1	0	Historical Record Flag
<u>ITXNR</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ITXNR	NUMC	8	0	Text Module for Infotype
<u>.INCLUDE</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	HRI9501	STRU	0	0	Additional Work Detail
<u>ZZ GOVT WRK</u>	<input type="checkbox"/>	<input type="checkbox"/>	ZDE GOVT WRK	CHAR	1	0	Government Work
<u>ZZ MIN THESIS</u>	<input type="checkbox"/>	<input type="checkbox"/>	ZDE MIN THESIS	NUMC	3	0	Min. no. of Thesis Submitted
<u>ZZ MIN PAPER</u>	<input type="checkbox"/>	<input type="checkbox"/>	ZDE MIN PAPER	NUMC	3	0	Min. no. of Papers Published
<u>ZZ MIN PATENT</u>	<input type="checkbox"/>	<input type="checkbox"/>	ZDE MIN PATENT	NUMC	3	0	Min. no. of Patents Filed

- Go to Transaction OOIT and maintain time constraint and objects definition for infotype

New Entries: Details of Added Entries

	
Dialog Structure ▾ Infotypes ▾ Time constraint ▾ Infotypes per object t	Object type NB Infotype 9501 Subtype Time constraint 3 Addl.condition

- Now create a new infotype access class for accessing data from the infotype.

Define class "CL_HRRCF_INFOTYPE" as its super class.

Creating a New BSP Application for Rendering New Fields

Now we are all set with the database of our new fields, so our next task is to create an application page which will provide UI for new fields. Here we will create a new BSP application which will be later added via customizing to the container sequence of requisition maintenance.

The best way to create a new BSP application is to copy a standard BSP application which closely meets your requirement, and then we will prune it to meet our exact requirements

In our case, we are going to copy standard application "HRRCF_REQ_MNT". We will keep the following views and controllers in the application and rename them as follows:

Controllers		Views	
Original name	New Name	Original name	New name
basicData.do	Addwork.do	basicData.bsp	Addwork.bsp
basicInfo_m.do	Addwork_m.do	basicInfo_m.bsp	Addwork_m.bsp
error.do	error.do	error.bsp	error.bsp

Now create two new controller classes with following configuration for controllers.

Controllers	Controller class configuration
Addwork_m.do	Super class: CL_HRRCF_CONTROLLER Interfaces: IF_HRRCF_REQ_MNT_EXCHG
Addwork.do	Super class: CL_HRRCF_CONTROLLER Interfaces: IF_HRRCF_SUBVIEW

Again here, the easiest way to create controller classes is to copy from one of the standard controller class, for e.g., here we can copy from controller classes of "basicdata.do" and "basicinfo_m.do".

Just go through the standard code to derive logic for methods of interface and to identify which all are required. If you have copied from standard controller class then additionally in your new controller class you have to make changes in methods like "DO_INIT", "DO_INITATTRIBUTES", "DO_HANDLE_EVENT", "ON_SAVE", to save and retrieve data from business logic layer (See Architecture diagram above).

Change/create main view for fields rendering, this main view will contain a call to your view with fields. An example in our case is as follows:

```

1  <@page language="abap" otrTrim="true" %>
2  <@extension name="htmlb" prefix="htmlb" %>
3  <@extension name="phtmlb" prefix="phtmlb" %>
4  <@extension name="bsp" prefix="bsp" %>
5  <@extension name="hrrcf bsp ext" prefix="hrrcf" %>
6  <!-- BSP_Application = ZHRRCF_ADD_WRK - HRRCF_Viewname = Addwork_m.bsp -->
7  <hrrcf:mainGroup title="<%= otr(ZEREC_WRK/ADDWORK_BASIC_INFO) %>" >
8  <phtmlb:matrix width="100%" >
9  <!-- Documentation -->
10 <phtmlb:matrixCell row = "1"
11 col = "1" />
12 <htmlb:textView id = "addworkInfo"
13 wrapping = "true"
14 text = "<%= otr(ZEREC_WRK/ADDWORK_INFO) %>" />
15 <!-- DetailScreen for Additional Work Data -->
16 <phtmlb:matrixCell row = "2"
17 col = "1" />
18 <br>
19 <bsp:call url = "Addwork.do"
20 comp_id = "addwork" />
21 <!-- Buttons -->
22 <phtmlb:matrixCell row = "3"
23 col = "1" />
24 <br>
25 <htmlb:button id = "save"
26 text = "<%= otr(ZEREC_WRK/ADDWORK_SAVE) %>"
27 width = "10%"
28 onClick = "onSave" />
29 <!-- Return Message -->
30 <%
31 if not controller->return_message is initial.
32 %>
33 <phtmlb:matrixCell row = "4"
34 col = "1" />
35 <hrrcf:messageBar id = "message"
36 type = "<%= controller->message_type %>"
37 rulerDisplay = "top"
38 text = "<%= controller->return_message %>" />
39 <%
40 clear: controller->return_message, controller->message_type.
41 endif.
42 %>
43 </phtmlb:matrix>
44 </hrrcf:mainGroup>

```

Add Comment like this, this will help to get the application and view name from internet explorer view source functionality.

Create your own OTR texts for labels and textview, this will help you change OTR's based on context's

Create a message bar to display any user errors. Do not display system errors, which are not understandable by user, such errors must be logged in error log for access by system Administrator.

Change/create BSP view for rendering of new fields. Use `<hrrcf:formLayout *>` tag for fields rendering on view, so that, each field can be customized with the context, check out the documentation of `<phtmlb:formLayout*>` before using.

Create OTR texts for your field labels, its better to create new OTR's rather than using existing ones, this will help you to have different text's for different context's with out effecting other objects

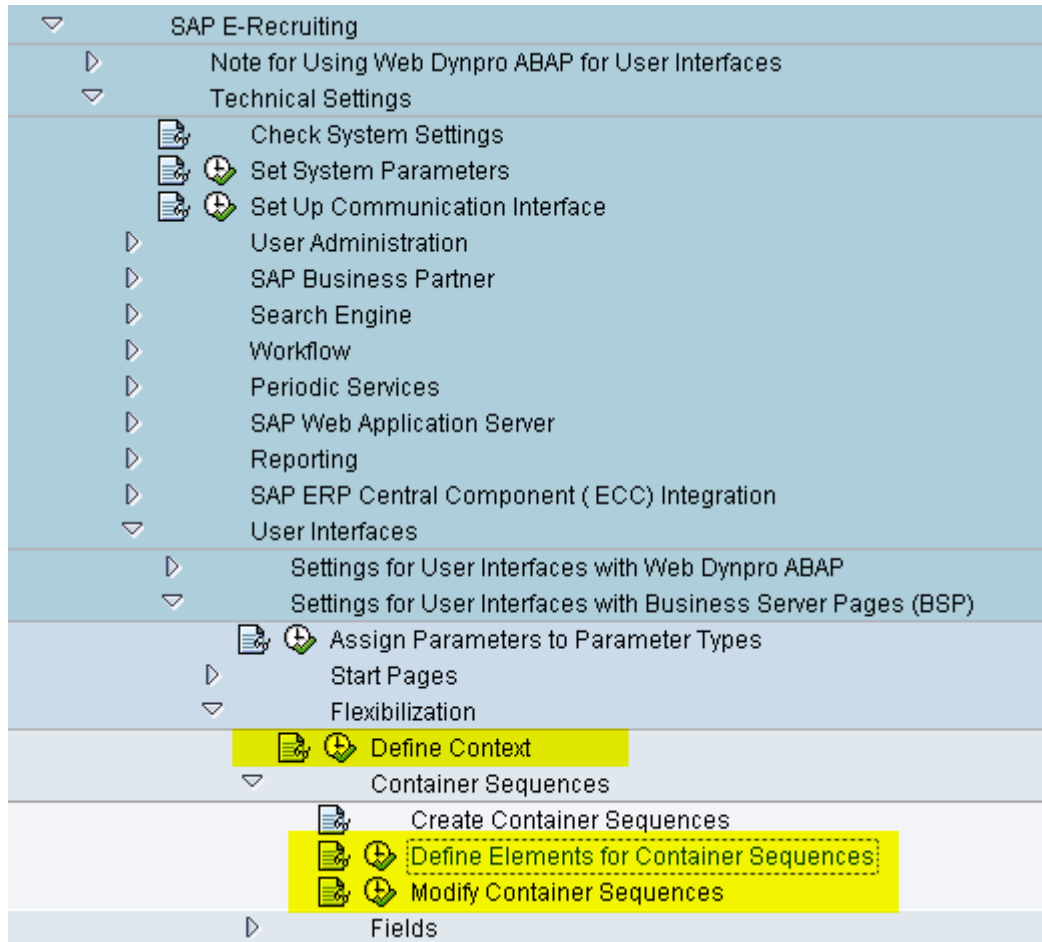
Maintaining Additional Customizing for Custom Fields

The required customization can be sub-divided into two parts:

1. Customization required for adding new tab
2. Field level customization for newly added fields

Customization Required for Adding New Tab

Follow path in IMG as mentioned below:



1. Define Context:

Here you will define a new context for which newly added tab will be rendered.

2. Define Elements for Container Sequences

Here a new element will be defined for new tab.

Element

Definition of Elements of Container Sequences	
Name	<input type="text" value="Additional work data requirement"/>
Namespace	<input type="text"/>
Application	<input type="text" value="ZHRRCF_ADD_WRK"/>
Controller URL	<input type="text" value="ADDWORK.DO"/>
OTR Text Alias	<input type="text" value="ZEREC_WRK/ADDWORK_BASIC_INFO"/> <input type="button" value="OTR"/>

3. Modify Container Sequences

Define a container sequence for requisition.

Dialog Structure

- Container Sequence
 - Assignment of Elements

Container Seq.

Container Sequence	
Name	<input type="text" value="Requisition Maintenance"/>
OTR Alias Title	<input type="text" value="PAOC_RCF_UI/REQUISITION_MAINTENANCE"/> <input type="button" value="OTR"/>
OTR Alias Descr.	<input type="text" value="PAOC_RCF_UI/REQUI_DOCU_1"/> <input type="button" value="OTR"/>

Bind elements to the container sequence defined above for requisition maintenance.

Dialog Structure

- Container Sequence
 - Assignment of Elements

Container Seq.

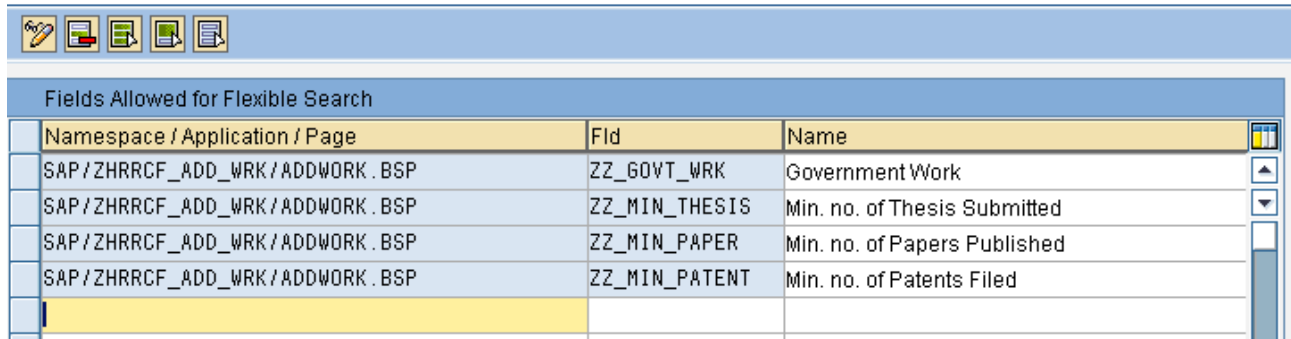
Name

Assignment of Elements				
Context	Name	No.	Element	Name
ZEREC	E Recruitment Test	0001	1000	Requisition: General
ZEREC	E Recruitment Test	0002	9501	Additional work data
ZEREC	E Recruitment Test	0003	1003	Requisition: Education
ZEREC	E Recruitment Test	0004	1002	Requisition: Job R
ZEREC	E Recruitment Test	0005	1001	Support Team
ZEREC	E Recruitment Test	0006	1004	Requisition: Status
ZEREC	E Recruitment Test	0007	1006	Job Postings

Field Level Customization for Newly Added Fields

Here we will enable all added custom fields for configuration:

For all custom added fields we have to make an entry in a customization view "VRCFPHTMLB_FLI" (customizing table PHTMLB_FLI)



Fields Allowed for Flexible Search		
Namespace / Application / Page	Fld	Name
SAP/ZHRRCF_ADD_WRK/ADDWORK.BSP	ZZ_GOVT_WRK	Government Work
SAP/ZHRRCF_ADD_WRK/ADDWORK.BSP	ZZ_MIN_THESIS	Min. no. of Thesis Submitted
SAP/ZHRRCF_ADD_WRK/ADDWORK.BSP	ZZ_MIN_PAPER	Min. no. of Papers Published
SAP/ZHRRCF_ADD_WRK/ADDWORK.BSP	ZZ_MIN_PATENT	Min. no. of Patents Filed

The entries in field (Fld) column should match the ID of elements on your BSP view for those elements.

Also refer customization part of document [E-Recruitment Adding Additional Custom Fields to Requisition Maintenance](#).

Making fields Search Relevant

The next step is to make our custom added fields, search relevant, i.e. to make fields available, whenever someone searches for postings or requisitions. This part can be subdivided into two as follows:

1. Enabling search profile generation of field
2. Enabling search UI for field.

Enabling Search Profile Generation of Field

To make a field relevant for search all we need a CEC (Content extractor class) and Infotype access class. We have already created infotype access class, so the left is CEC class.

Create a global class and assign following interfaces to it:

1. IF_HRRCF_CONSTANTS
2. IF_HRRCF_CONTENT_EXTRACTOR

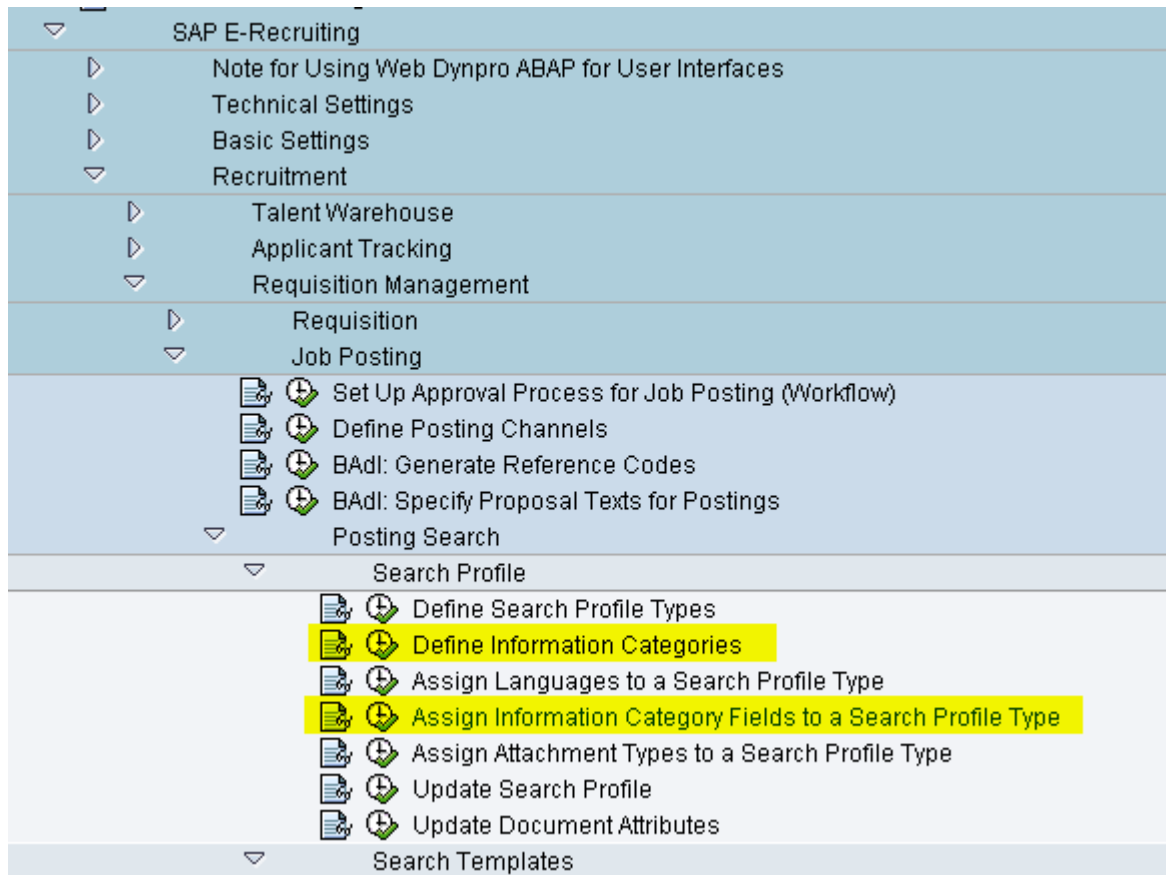
You can also refer standard CEC class like "CL_HRRCF_CEC_INFOTYPE" for creating CEC class of your own.

In most of the cases, where there are no package namespace, standard CEC class can be used rather than creating a new one. The reason being, the standard class uses offset "1" to get the infotype no. from the infotype P structure, this will actually fail in case of package namespace, as the P structure name is prefixed by package namespace, if it exist. So in all such cases its advisable to create a new CEC class.

Note: If you are copying the standard CEC class to create a new one, just check all the logic and change code where ever offset is read to get the infotype no. correctly.

Now register your infotype access class for search profile using maintenance view "T77RCF_SPT_INFOTY".

For rest of the customization follow the IMG path as below:



1. Define Information Categories

Define new information category using the CEC class and Infotype “P” structure created above. Remember the P structure will be pre-fixed by the package namespace, if exist. Assign this for object type “NB” i.e. requisitions.

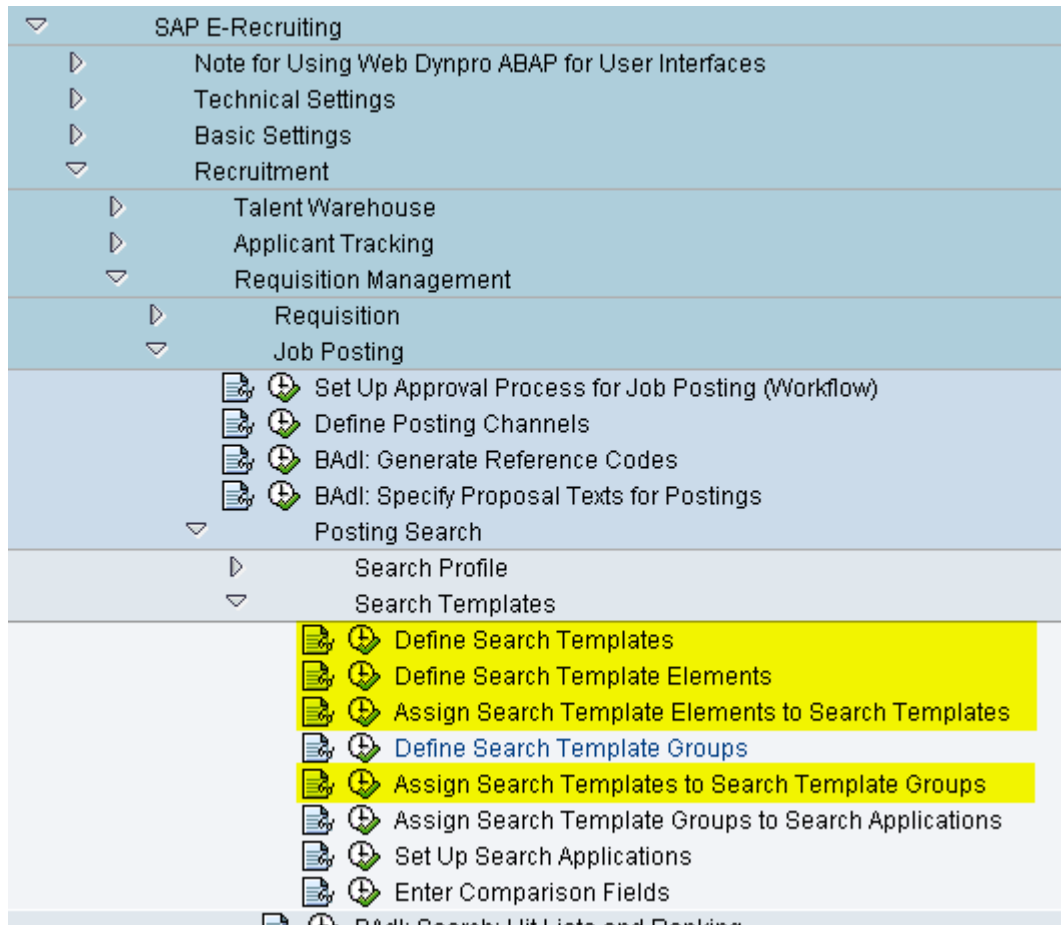
2. Assign Information Category Fields to a Search Profile Type

Make assignment for both “EXT_POST” and “INT_POER” i.e. internal and external postings and requisitions. Use the Information category defined above and select the custom added fields using F4 help (These fields are populated from the structure defined for information category).

Use “Type” either “Free text” (If it’s a free text field, as in our case), “Entity” (If it has a value table or check table) or both (if you want the field to be searched from both value selection or free text search, this is more relevant in case the field has values associated with it, and has a text table as well and you want search based on text but for this you must have implemented logic for method “ASSIGN_TEXT_TO_ENTRIES” in infotype access class)

Enabling Search UI for Field

Follow the IMG path as below



1. Define Search Templates

Define two new search templates one for each “EXT_POST” and “INT_POER”. Use type “General”.

2. Define Search Template Elements

Define new template elements for each new custom field. For each field define the attributes of information category defined above. If you have specified element type which require set of values like drop down, list box etc. then you also have to define Data class, for data access class standard class can be referred like “CL_HRRCF_T77RCF_DEGREE”.

3. Assign Search Template Elements to Search Templates

4. Assign Search Templates to Search Template Groups

Assign newly created search templates to existing search template groups which includes requisition search.

UI Output of Our Requirement

So here is the UI output in our case for requisition maintenance:

The screenshot displays the SAP Requisition Maintenance interface. At the top, the breadcrumb navigation shows 'Personal Pages > Requisition Maintenance > Requisition'. Below this, the requisition number '1850004220' is visible, along with a 'Status' field set to 'Draft'. A brief description states: 'A requisition is a formal request to fill one or more positions. Store the relevant data and release the requisition.' The main navigation bar includes tabs for 'Requisition', 'General Job Information', 'Additional Work related Detail', 'Education Requirements', 'Job Requirements', and 'Support Team'. The 'Additional Work related Detail' tab is currently active, showing a table with the following data:

Additional Work related Detail	
Government Work	<input checked="" type="checkbox"/>
Min. no. of Thesis Submitted	<input type="text" value="12"/>
Min. no. of Papers Published	<input type="text" value="3"/>
Min. no. of Patents Filed	<input type="text" value="5"/>

Related Content

[SAP E-Recruiting](#)

[Business Server Pages](#)

[Enhancements to Global Classes and Interfaces](#)

[E-Recruitment Adding Additional Custom Fields to Requisition Maintenance.](#)

Copyright

© Copyright 2009 SAP AG. All rights reserved.

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.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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 other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

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.

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.