Save Some Trees: Getting Rid of Paper Security Forms and Do Online Role Requests

Session #26395

March 23, 2009

ALLIANCE'09

Anaheim, California

Presenters

- **Chuck Stubbs**
  - Florida State University
  - Director of Technical Development
- **Amardeep Tekriwal**
  - Florida State University
  - ERP Analyst IV – Lead Developer

Agenda/Contents

- Configuration

- End User experience

- Live Demo

- Technical info

- Q&A

PeopleSoft at FSU

- HCM – 9.0
- FIN – 8.9 upgrading to 9.0 next week
- Portal – 9.0
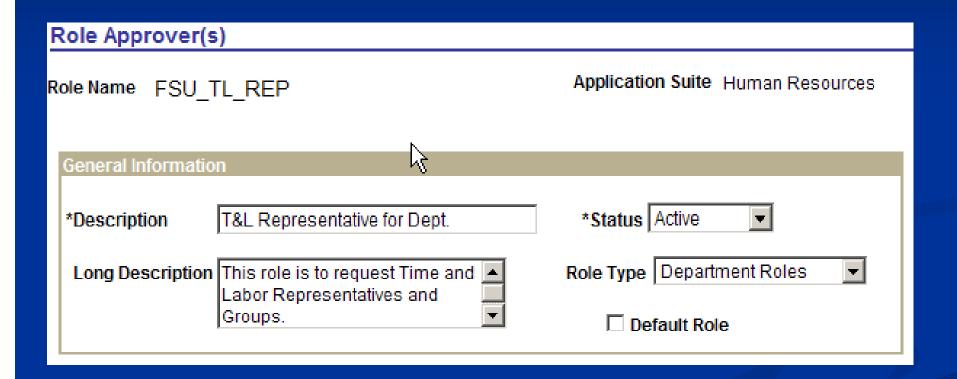- EPM - 9.0
- Tools – 8.49

Purpose of Customization

- Automate the existing paper process

- Speed up approval and granting/removal of roles

- Improve the Security request approval Audit trail

Additional Info

- Modification is in the HR Suite but manages roles in all environments
- Still a paper process for some Roles and User Preferences
- Mutually exclusive roles

Configuration

- Add Role
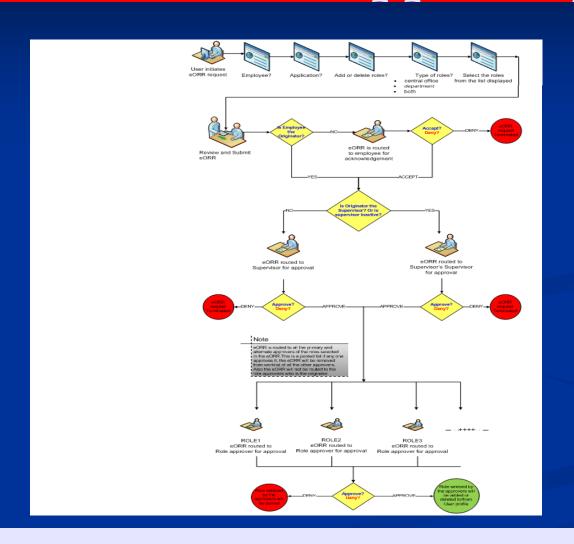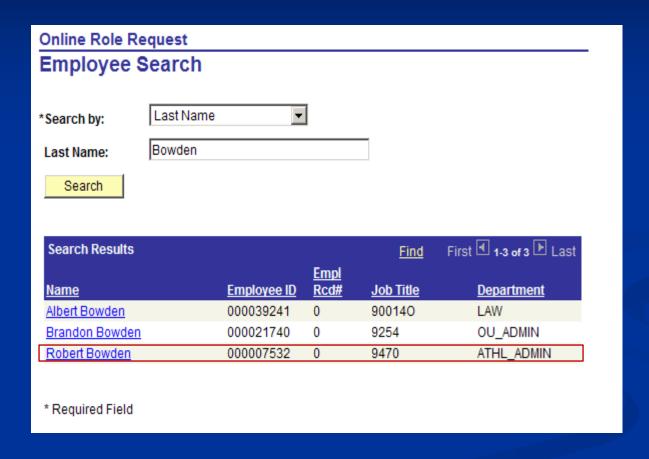- Add Role Approvers
- Set Mutually Exclusive Roles
- Add Messages

Configuration Pages

Configuration Pages
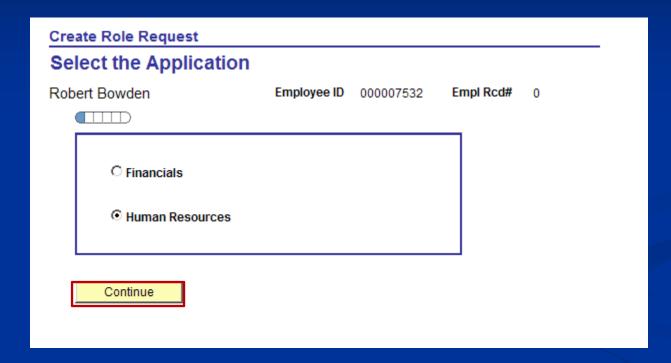
Configuration Pages

Configuration Pages

Processes performed by users

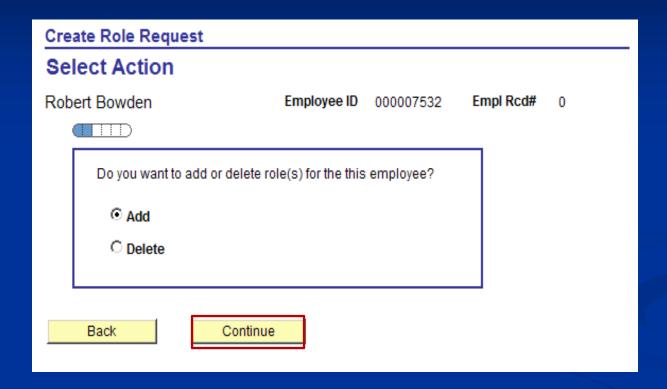- Online Role Request

- Approve Online Role Request

Workflow Approval path

Request Process

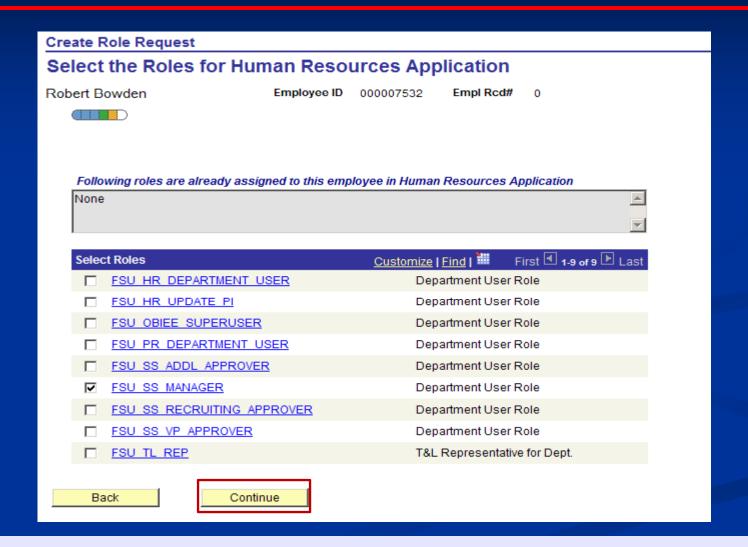# Request Process

Create Role Request

Select the Application

Robert Bowden   Employee ID   000007532   Empl Rcd#   0

○ Financials

◉ Human Resources

Continue

Request Process

Request Process

Request Process

Request Process

Request Process

Approve Request Process

Approve Request Process

Live Demo

PS Objects/Technology

- AWE (Approval Workflow Engine)
- Sync and Async service operations (Application Message)
- Component Interface
- Application Package
- Record/Page/Component/PeopleCode

Workflow Setup(AWE)

- Register Transaction

- Configure Transaction

- Setup Process Definition

- Email templates

- User list (Approvers/Reviewers)

Workflow Setup(AWE)

Workflow Setup(AWE)

Configure Transactions

Generic Template Definition	Blackberry Email Responses

Template: FSU eORR Approved

*Description: FSU eORR Approved

Instructional Text: Type names or email addresses in the To, CC, or BCC fields, using a semi-colon as a separator.
Click LOOKUP RECIPIENT to search for a name. Click DELIVERY OPTIONS to view or change the method of the

Priority: 1-High

*Sender: System   **Email ID:**

Subject: Approval of Online Role Request for %5, Employee ID %3

Message Text: Some of the roles in the online role request to %7 OMNI security roles for the below employee have been approved and/or denied.

Request ID: %9
Employee: %3
Employee Record: %4
Name: %5
Department: %8

Below is the list of available variables for this template.
You can use template variables within your subject or message text.
The following variables can also be used:
%Date, %DateTime, %Time, %ServerTimeZone, %EmailAddress, %NotificationPriority, %NotificationToList, %NotificationCCList

Template Variables

*Value	*Description		
%1	URL	+	−
%2	Role Request ID	+	−
%3	Employee ID	+	−
%4	Employee Record	+	−
%5	Name	+	−
%6	Application	+	−
%7	Action	+	−

Workflow Setup(AWE)

Workflow Setup(AWE)

User List Definition

User List: FSU_EORR_SUP_APR

*Description: FSU_EORR_SUP_APR

User List Source

- ○ Role
- ⦿ SQL Definition   SQL Object Identifier: FSU_E
- ○ Query
- ○ Application Class

☑ Include Users as Input
☑ Transaction Keys as Input

[Save] [Return to Search] [Notify]

FSU_EORR_SUP_APR.0.GBL..1900-01-01 (SQL Definition)

[default]

```
SELECT B.OPRID
  FROM PS_FSU_ROLEWF_HDR A
  ,PSOPRDEFN B
WHERE A.REQUEST_ID = :1
  AND A.SUPERVISOR_ID <> ''
  AND A.SUPERVISOR_ID = B.EMPLID
  AND B.OPRID <> A.ORIGINATORID
  UNION
SELECT B.OPRID
  FROM PS_FSU_ROLEWF_HDR A
  ,PSOPRDEFN B
WHERE A.REQUEST_ID = :1
  AND A.FSU_SUP_SUP_ID <> ''
  AND A.FSU_SUP_SUP_ID = B.EMPLID
  AND ( A.SUPERVISOR_ID = ''
  OR NOT EXISTS (
SELECT 'X'
  FROM PS_FSU_ROLEWF_HDR A
  ,PSOPRDEFN B
WHERE A.REQUEST_ID = :1
  AND A.SUPERVISOR_ID = B.EMPLID)
  OR EXISTS (
SELECT 'X'
  FROM PS_FSU_ROLEWF_HDR A
  ,PSOPRDEFN B
WHERE A.REQUEST_ID = :1
  AND A.SUPERVISOR_ID = B.EMPLID
  AND B.OPRID = A.ORIGINATORID) )
```

Code for Workflow

- Created Application Package that contained all the method to initialize,submit,clean-up transaction

code for Workflow

- Created another Application Package(event handler) that contained all the method to add/delete roles in User profile,create app message to integrate with other PeopleSoft applications, set transaction status to Approve/denied/in progress ,error handling etc

Code used to fire Wokflow (Submit)

- Peoplecode to submit transaction should be written in SavePostChange event

```
FSU_EORR_SUBMIT.GBL  (component)                    ▼  SavePostChange

import PTAF_MONITOR:MONITOR:awStatusMonitor;
import HRS_AWE:WRAPPERS:AdhocAccessLogicBase;

Declare Function UpdateStatus PeopleCode FSU_EORR_WRK.HMAF_STATUS FieldFormula;
Component string &strEORRAprvInd;
Global string &FSU_Resubmit;

Component HRS_AWE:INTERFACES:ILaunchManager &LaunchMgr;
Component HRS_AWE:INTERFACES:IApprovalManager &AprvMgr;
rem Component HRS_AWE:INTERFACES:IStatusMonitor &StatusMonitor;
Component PTAF_MONITOR:MONITOR:awStatusMonitor &StatusMonitor;
Component PTAF_MONITOR:MONITOR:awStatusMonitor &monitor;
rem Local string &strSetID = GetSetId(Field.BUSINESS_UNIT, FSU_ROLEWF_HDR.BUSINESS_UNIT, "SAC_AW_PRCS", "");
Local string &strSetID = FSU_ROLEWF_HDR.SETID;


&eORR_WF = create FSU_EORR_WORKFLOW:FSU_SUBMIT_EORR_WF();
&eORR_WF_UI = create FSU_EORR_WORKFLOW:FSU_EORR_UI();

Local Record &recEORRCopy = CreateRecord(Record.FSU_ROLEWF_HDR);
GetRecord(Record.FSU_ROLEWF_HDR).CopyFieldsTo(&recEORRCopy);

&AprvFactory = create HRS_AWE:ApprovalsFactory();

&AprvMgr = &AprvFactory.getApprovalManager("FSU_EORR", &recEORRCopy, %OperatorId);
&LaunchMgr = &eORR_WF.FSU_Init_eORR_Approval(&recEORRCopy);

If &strEORRAprvInd = "S" And
    &FSU_Resubmit <> "Y" Then

  /* This is the submit to the FSU approvals app package. */
  &eORR_WF.FSU_Submit_eORR_Approval(&recEORRCopy);    ◄━━━━━━━━

  /* Below is what does the UI formatting and displays things to the user. */
  &eORR_WF_UI.FSU_EORR_displayAprvMonitor();
  &eORR_WF.FSU_saveAdhocSteps();
  &eORR_WF.FSU_do_submit_cleanup();

End-If;
```

Code used to Submit transaction

- Use delivered method DoSubmit method to Submit transaction for Workflow

```
import HRS_AWE:INTERFACES:ILaunchManager;
import HRS_AWE:INTERFACES:IApprovalManager;
import HRS_AWE:ApprovalsFactory;
import FSU_EORR_WORKFLOW:COMMON:Exception;


/* Class FSU_SUBMIT_eORR_WF
   Module: eORR Workflow */

class FSU_SUBMIT_EORR_WF
    method FSU_Submit_eORR_Approval(&recEORR As Record);
    method FSU_Init_eORR_Approval(&recEORR As Record) Returns HRS_AWE:INTERFACES:ILaunchManager;
    method FSU_saveAdhocSteps();
    method FSU_do_submit_cleanup();
    /* private
    instance HRS_AWE:ApprovalsFactory &AprvFactory;
    instance HRS_AWE:INTERFACES:ILaunchManager &LaunchMgr; */
end-class;


Component string &streORRAprvInd;
Component HRS_AWE:ApprovalsFactory &AprvFactory;
Component HRS_AWE:INTERFACES:ILaunchManager &LaunchMgr;
Component HRS_AWE:INTERFACES:IApprovalManager &AprvMgr;


/* This Method is what submits the eORR for Approvals */
method FSU_Submit_eORR_Approval
    /+ &recEORR as Record +/

    &LaunchMgr = %This.FSU_Init_eORR_Approval(&recEORR);
    If &LaunchMgr.submitEnabled Then
        /* &LaunchMgr.requester = &HR.getOprID(&recJO.MANAGER_ID.Value); */
        &LaunchMgr.requester = %OperatorId;
        &LaunchMgr.DoSubmit();
    End-If;
end-method;
```

Code to Approve/Deny transaction

- **When Users Approve or Deny a Transaction**
  - Use DoApproveRowSet or DoDenyRowSet methods to approve or deny line level transactions
  - Use DoApprove or DoDeny methods to approve or deny header level transactions

```
If &Approve_transaction = "Y" Then
    &FSU_Global_Approve = "Y";
    &strEORRAprvInd = "A";
    &AprvMgr.DoApproveRowSet(&rsRoleWFLine);     ⬅
End-If;

If &Deny_transaction = "Y" Then

    &strEORRAprvInd = "D";
    &AprvMgr.DoDenyRowSet(&rsDenyRoleWFLine);    ⬅
End-If;
```

Application Message

- Created two Application messages (Services)
  - Synchronous message to fetch the existing roles for employee from other PeopleSoft apps
  - Asynchronous message to add/delete roles

Application Message

- Code to fetch existing roles for an employee in other PeopleSoft suites (Financials) using a Synchronous application message

```
PUSH  [field]                                    ▼  FieldFormula

Function PopulateExistingRoles()

    Local Message &request, &response;
    Local Rowset &req_rowset, &rep_rowset;
    Local Record &req_rec, &rep_rec;
    Local Row &req_row;
    Local string &LabelStr, &RemPrefix, &ProdRequested;
    Local any &PS_GL;
    Local SQL &getRoles;
    Local string &userid;

    &DisplayRolename = "";
    /*get userid of the employee*/
    &userid = getUserid();


    If FSU_ROLEWF_HDR.FSU_APP_SUITE = "FIN" Then

        If IsMessageActive(Message.FSU_ROLE_REQUEST) Then

            &request = CreateMessage(Message.FSU_ROLE_REQUEST);
            &req_rowset = CreateRowset(Record.FSU_ROLE_MSG_H);
            &req_rowset(1).GetRecord(Record.FSU_ROLE_MSG_H).GetField(Field.ROLEUSER).Value = &userid;

            &request.CopyRowset(&req_rowset);
            &response = &request.SyncRequest();        ◄━━━━━━━━
            If (&response.ResponseStatus = 0) Then
                &rep_rowset = &response.GetRowset();
                For &i = 1 To &rep_rowset.ActiveRowCount
                    &rolename = &rep_rowset(&i).GetRecord(Record.FSU_ROLE_MSG_H).GetField(Field.ROLENAME).Value;
                    /*Display only those roles that exists in Role Configuration tables*/
                    SQLExec("Select 'X' from PS_FSU_ROLEAPR_HDR A Where A.Rolename = :1 AND  A.FSU_APP_SUITE = :2 AND A.ROLESTATUS = 'A'
                    If All(&exists) Then
                        &DisplayRolename = &DisplayRolename | Char(10) | &rolename;
                    End-If;
                End-For;
            End-If;
        End-If;

    Else
        If FSU_ROLEWF_HDR.FSU_APP_SUITE = "HRMS" Then
            &getRoles = CreateSQL("SELECT ROLENAME FROM PSROLEUSER WHERE ROLEUSER = : 1", &userid);
            While &getRoles.Fetch(&rolename)
```

Application Message

- Code to Publish application messages on Final Approval to add roles in other PeopleSoft suites (Financials) using Asynchronous app message

```
EORR_Handler  (application_class)                              ▼    OnExecute

   end-try;

   Break;
When = "FIN"
   /* This part of the code sends a Async message to Fin to add the approved roles to the User profile*/
   /* If IsMessageActive(Message.FSU_USER_ROLEMAINTN) Then*/
   If %IntBroker.IsOperationActive(Message.FSU_USER_ROLEMAINTN) Then
      &EORR_ROLEMAINT_MSG = CreateMessage(Message.FSU_USER_ROLEMAINTN);
      &ROLEMAINT_RS = &EORR_ROLEMAINT_MSG.GetRowset();

      For &arrayIndex = 1 To &ApprovedRolesArray.Len
         If &m > 1 Then
            &ROLEMAINT_RS.InsertRow(&m - 1);
         End-If;
         &ROLEMAINT_RS.GetRow(&m).GetRecord(Record.FSU_RL_MANT_MSG).GetField(Field.ROLEUSER).Value = &oprid;
         &ROLEMAINT_RS.GetRow(&m).GetRecord(Record.FSU_RL_MANT_MSG).GetField(Field.ROLENAME).Value = LTrim(RTrim(&ApprovedRolesA
         &ROLEMAINT_RS.GetRow(&m).GetRecord(Record.FSU_RL_MANT_MSG).GetField(Field.FSU_ACTION_TYPE).Value = "ADD";
         &m = &m + 1;

      End-For;

      If None(&ROLEMAINT_RS.GetRow(1).GetRecord(Record.FSU_RL_MANT_MSG).GetField(Field.ROLENAME).Value) Then
         rem do nothing;
      Else
         %IntBroker.Publish(&EORR_ROLEMAINT_MSG);      ⬅
      End-If;
   Else
      WinMessage("Message FSU_USER_ROLEMAINTN is NOT active", 0);
   End-If;
   Break;

End-Evaluate;
```

37

Questions?

Contacts

- Chuck Stubbs

- Director of Technical Team

- ERP

- Florida State University

- E-mail: cstubbs@admin.fsu.edu

- Amardeep Tekriwal

- Senior Developer

- EPR

- Florida State University

- E-mail: atekriwal@admin.fsu.edu

This presentation and all Alliance 2009 presentations are available for download from the Conference Site

Presentations from previous meetings are also available