Release 1974

Service Request 83181
Error Report 2363

Web Merit View-Only Access

Detail Design

April 22, 2011
Prepared by Nancy Field
Sreelekha Sen

Information Resources & Communications
Office of the President
University of California
Table of Contents

Introduction .................................................................................................................................................................................... 1
  Service Request 83181 .................................................................................................................................................................... 1
  Error Report 2363 ......................................................................................................................................................................... 1

Overview of PPS Modifications ................................................................................................................................................ 1

Programs ..................................................................................................................................................................................... 2
  Modified Programs .................................................................................................................................................................... 2
  UCPPPMRT .............................................................................................................................................................................. 2
  PPMDDPSPR ............................................................................................................................................................................ 2
  PPCSTSPR ................................................................................................................................................................................ 2

Control Table Updates ............................................................................................................................................................. 3
  UC0RUL .................................................................................................................................................................................... 3
  UC0ASC ................................................................................................................................................................................ 3

Java Changes ............................................................................................................................................................................. 4
  Merit.controllers .................................................................................................................................................................... 4
  ControllerServlet.java ............................................................................................................................................................. 4
  Merit.dataAccessBeans ........................................................................................................................................................ 4
  ARSMAccessBean.java ............................................................................................................................................................ 4
  CommonDataAccessBean.java ................................................................................................................................................ 4
  CycleIdDetailAccessBean.java ................................................................................................................................................ 4
  RosterDataAccessBean.java .................................................................................................................................................. 4
  Merit.displayBeans ............................................................................................................................................................... 4
  AddTitleRangeObject.java .................................................................................................................................................... 4
  CycleBaseObject.java ............................................................................................................................................................ 5
  CycleDetailObject.java ........................................................................................................................................................... 5
  CyclePerfObject.java ............................................................................................................................................................ 5
  DeptListObject.java ............................................................................................................................................................... 5
  RosterBaseObject.java ........................................................................................................................................................... 5
  RosterEmpDetailObject.java .................................................................................................................................................. 5
  RosterPageObject.java .......................................................................................................................................................... 5
  Merit.eventHandlers ............................................................................................................................................................... 5
  AdminCyclePerf.java ............................................................................................................................................................... 5
  AdminCycleProcessHandler.java ............................................................................................................................................ 5
  AdminCycleUpdate.java .......................................................................................................................................................... 5
  CycleListHandler.java ........................................................................................................................................................... 5
  DeptListHandler.java ............................................................................................................................................................ 5
  EventHandlerBase.java .......................................................................................................................................................... 6
  MainMenuHandler.java ........................................................................................................................................................... 6
  RosterHandler.java ................................................................................................................................................................. 6
  Merit.filters ............................................................................................................................................................................... 6
  AccessCheckFilter.java ............................................................................................................................................................ 6
  RedirectFilter.java ................................................................................................................................................................. 6
  TokenCheckFilter.java ............................................................................................................................................................ 6
  Merit.parameters .................................................................................................................................................................... 6
    constant.properties ............................................................................................................................................................... 6
    Parm.java ............................................................................................................................................................................... 7
  Merit.util ................................................................................................................................................................................ 7
    StringSanitizer.java ............................................................................................................................................................. 7
    Merit.util.taglib ................................................................................................................................................................. 7
    EmpDetailTokenTag.java .................................................................................................................................................... 7
    TokenTag.java ..................................................................................................................................................................... 7
<table>
<thead>
<tr>
<th>Web Pages</th>
<th>...............................................................................................................................................................................</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>RosterRangeBased.js</td>
<td>...............................................................................................................................................................................</td>
<td>7</td>
</tr>
<tr>
<td>RosterStepBased.js</td>
<td>...............................................................................................................................................................................</td>
<td>7</td>
</tr>
<tr>
<td>AdminCycleAddNew.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>7</td>
</tr>
<tr>
<td>AdminCycleEdit.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>AdminCycleList.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>AdminCycleUpdateStatus.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>AdminPerfEval.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>AdminPerfRatingRange.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>AdminPerfRatingStep.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>DisplayCbuc.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>DisplayRangeRateTable.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>DisplayStepRateTable.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>DisplaySubLoc.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>Error.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>MsgDisplay.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>RosterRangeBased.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>RosterRangeEmpDetail.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>RosterStepBased.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>RosterStepEmpDetail.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>ViewOnlyRosterRangeBased.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>ViewOnlyRosterRangeEmpDetail.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>ViewOnlyRosterStepBased.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>ViewOnlyRosterStepEmpDetail.jsp</td>
<td>...............................................................................................................................................................................</td>
<td>9</td>
</tr>
</tbody>
</table>
Introduction

Service Request 83181

Service Request 83181 asks that view-only access be provided for the Web Merit application. Users with the new view-only access should be able to view anything within the Merit Review/Input selection on the Merit Menu but should not be able to modify or save any data, regardless of cycle status. They should have the ability to navigate within the Merit Review/Input portion of the application and view information on pop-up windows (e.g. cycle information, title information, employee information).

Specifically, the SR asks for two types of view-only access:

- View-only access that is restricted to particular departments
- Universal view-only access

Error Report 2363

UCLA has reported that the Web Merit application has the following security vulnerabilities:

1. The application does not adequately scrub user submitted data which can potentially allow for cross-site scripting and cross-site request forgery.
2. The application is susceptible to cross-site request forgery (CSRF).
3. In terms of application error diagnostics, when the application encounters an error, it allows the user to see the stack trace; it should give the user a reference that the support team can use to find the stack trace in the application log. This is a minor threat but it is good practice to avoid revealing internal application details to end users.

Overview of PPS Modifications

To provide view-only access, modifications will be made to the Cobol stored procedure which is the ARSM security driver for access to Web Merit data. Changes will also be made to the stored procedure that identifies departments for the merit roster and inserts them into the merit department table. Two new rule types will be defined in the UCORUL rules table for view all access and view control point access. A new access intent for audit roster access will also be defined. Also, the code will be modified to return values of ‘U’ (for update), ‘R’ (for read) and ‘N’ (for no access), instead of ‘Y’ and ‘N’ in the current process.

On the java side, providing view-only access will be accomplished first by adding calls to the ARSM security driver in the program that handles the main menu for Web Merit to determine and save in session a user’s access. Second, a new filter program will be written to map, if necessary, a view-only version of the requested page, and to check for unauthorized access attempts.

Modifications to the java code to address the security vulnerabilities outlined in Error Report 2363 will be made as follows:

1. All the data entered on Web Merit pages goes through one specific routine in one program (ControllerServlet.java). Code will be added at this single point to scrub all incoming data.
2. To guard against CSRF, random tokens will be added to each web page where the user can enter data. When a web page comes back to the server, a program will check that the token matches the one saved in “memory” (session). In addition, the Web Merit application will be modified to use the post-redirect-get (PRG) approach for delivering web pages to the client. The PRG approach breaks post requests into 2 parts – post request incoming, then via a redirect, a get request outgoing.
3. A reference number will be generated at the time an error occurs. The reference number will be written to the logs and displayed on the error page for users to relay to the system administrator.
**Modified Programs**

**UCCPPPMRT**

UCCPPPMRT is a DB2 stored procedure which is the ARSM security driver for the Web Merit application.

Currently, when checking for user authorization, the program returns a PERMISSION flag of “Y” or “N”. “Y” currently implies update access since that’s the only type of access available. The program will be changed to return values of “U” for update, “R” for Read-only or “N” for no access.

Also, the program currently requires that an ARSM “PERSONAL” rule be present for cycle access. Since read-only users are not required to have a PERSONAL rule, the program will be changed to bypass the check for this rule for read-only users.

There will be two new rule types in the UC0RUL table, MERITVWA for view all and MERITVWC for view from a control point. For Access Intent of Cycle Access, the two new rules will be checked for cycle status of D, O, H, S and N.

A new Access Intent value should be added to the input request code to distinguish audit roster access from regular roster access, with value ‘AUDACC’.

In addition, to conform to best coding practices, the Insert SQL statement to insert into the Merit Department Table (PPPMDP) will be changed to reference individual columns of the table.

**PPMDPSPR**

PPMDPSPR is a stored procedure that identifies potential departments for merit roster selection and inserts them in the Merit Department table (PPPMDP).

The only change to this program is to include the two new rules MERITVWA and MERITVWC in the values for valid ARSM rules.

**PPCSTSPR**

PPCSTSPR is a stored procedure that is used to calculate costing for a Roster.

The change to this program is to include the two new rules MERITVWA and MERITVWC in the values for valid fund group. Also, in order to avoid changing the PPPFND table, when the ARSM rule is MERITVWA it is substituted by MERITADM and when it is MERITVWC it is substituted by MERITCTL. This achieves the same effect when the Cost Summary button is clicked on the roster page.
Control Table Updates

UC0RUL

The new MERITVWA and MERITWVC rules will be added to the UC0RUL table for access to MRT.

<table>
<thead>
<tr>
<th>ACCESS_RULE</th>
<th>RESOURCE_TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR(8)</td>
<td>CHAR(8)</td>
<td>CHAR(30)</td>
</tr>
<tr>
<td>PRIMARY-----</td>
<td>PRIMARY------</td>
<td>---------------------</td>
</tr>
<tr>
<td>MERITVWA</td>
<td>MRT</td>
<td>MERIT VIEW ALL</td>
</tr>
<tr>
<td>MERITWVC</td>
<td>MRT</td>
<td>MERIT VIEW CONTROL POINT</td>
</tr>
</tbody>
</table>

UC0ASC

For each user (or ARSM group) who will have read-only access, one or more rows must be added with either MERITVWA or MERITVWC and the appropriate control point. This can be done using the ARSM update screens or by directly updating the tables.

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>ACCESS_RULE</th>
<th>SEQ_NUMBER</th>
<th>ASSOCIATION_DATA</th>
<th>CHANGED_BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR(8)</td>
<td>CHAR(8)</td>
<td>SMALLINT</td>
<td>CHAR(18)</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PRIMARY----</td>
<td>PRIMARY-----</td>
<td>---PRIMARY</td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>PAYSXS</td>
<td>MERITVWA</td>
<td>1</td>
<td>222222</td>
<td>PAYSXS</td>
</tr>
</tbody>
</table>
Java Changes

Merit.controllers
Webmerit/src/merit/controllers/

ControllerServlet.java
This is the main controller for Web Merit. This program loads all parameters and constants used in the Web Merit application. A call to execute a new method to load a list used for mapping view-only access will be added.

This program contains a method through which all data passes. Code will be added to sanitize incoming data.

If an unexpected system error occurs, a reference number will be generated and written to the log along with the detailed error information. This reference number will be displayed on the error.jsp for the user to relay to the system administrator.

Merit.dataAccessBeans
Webmerit/src/merit/dataAccessBeans/

ARSMAccessBean.java
This program calls a COBOL stored procedure to determine the user’s access permissions. This program will be modified to incorporate changes necessary to support view-only access.

CommonDataAccessBean.java
This program gets code values used throughout the application. The code value for “ALL” sub locations and collective bargaining units is “@@”. This program will be modified to change the code for ALL to “ZZ” internally since the @ sign is one of the special characters designated to be stripped from data input because of security concerns. The code will remain “@@” in the database.

CycleIdDetailAccessBean.java
This program gets the data for a particular cycle from the database. This program will be modified to change the code for “ALL” for sublocation and CBU to “ZZ” from “@@” for internal use.

RosterDataAccessBean.java
This program gets roster data from the roster table PPPMLA. This program will be modified so search names will be positional parameters.

Merit.displayBeans
Webmerit/src/merit/displayBeans/

AddTitleRangeObject.java
This is the additional title code range object. This program will be serialized.
CycleBaseObject.java
This is the base cycle object. This program will be serialized and the code for “ALL” will be set to “ZZ” from “@@” for use in the drop down menus for sublocation and collective bargaining codes.

CycleDetailObject.java
This is the cycle detail object. This program will be serialized and the code for “ALL” will be set back to “@@” from “ZZ” for sublocation and collective bargaining codes so the data will be stored as expected in the database.

CyclePerfObject.java
This is the performance code object. This program will be serialized.

DeptListObject.java
This is the department list object. This program will be serialized.

RosterBaseObject.java
This is the roster base object. This program will be serialized.

RosterEmpDetailObject.java
This is the roster employee detail object. This program will be serialized.

RosterPageObject.java
This is the roster page object. This program will be serialized.

Merit.eventHandlers
Webmerit/src/merit/eventHandlers/

AdminCyclePerf.java
This is the handler for the “Performance Evaluation Table” page for Merit Administration. Changes will be made to support the PRG approach. These include saving objects to session and not to the request object.

AdminCycleProcessHandler.java
This is the handler for the “Merit Cycle Administration” page. Changes will be made to support the PRG approach. These include saving objects to session and not to the request object.

AdminCycleUpdate.java
This is the handler for updating the status for a merit cycle. Changes will be made to support the PRG approach. These include saving objects to session and not to the request object.

CycleListHandler.java
This is the handler for “Roster Cycle List” page. Changes will be made to support the PRG approach. These include saving objects to session and not to the request object.
DeptListHandler.java
This is the handler for the “Roster Department Selection List” page. Modifications will be made in how a user’s permission level is verified to support view-only access. In addition, changes will be made to support the PRG approach. These include saving objects to session and not to the request object.

EventHandlerBase.java
This is the main event handler for all pages in the Web Merit application. Code will be added to check if the user has view-only access, and if so, map the request if necessary to a new view-only version of the requested page. A change will also be made to support the PRG approach.

MainMenuHandler.java
This is the Merit Menu handler and it serves as an entry point to the application. A call to the Cobol stored procedure UCPPPMRT will be added to determine if the user has view-only access when first entering the application and changes will be made to support the PRG approach.

RosterHandler.java
This is the handler for the roster pages. Changes will be made to support the PRG approach. These include saving objects to session and not to the request object.

Merit.filters
Webmerit/src/merit/filters/

AccessCheckFilter.java
This will be a new program written to map view-only pages and check for unauthorized access.

RedirectFilter.java
This will be a new program written to support the PRG approach. The jsps are expecting data in the request object. This filter will move objects from session (saved by specific handlers) to the request object.

TokenCheckFilter.java
This will be a new program written to safeguard against CSRF attacks. The program TokenTag.java (new) will generate a hidden token on each jsp that allows data entry and will save the token in session. This filter will look for the handlers of those jsps and compare the token on return to the token saved in session. If the tokens are not the same, the user will be shown a blank message page.

Merit.parameters
Webmerit/src/merit/parameters/

constant.properties
This program contains the constant values used in the Web Merit application. A constant identifying view-only web pages will be added.
Parm.java
This program loads parameters and constant values. Code will be added to build a list of valid view-only pages.

Merit.util
Webmerit/src/merit/util

StringSanitizer.java
This will be a new program written to safeguard against CSRF attacks. StringSanitizer.java will be the utility class called to sanitize, or remove, potentially harmful characters.

Merit.util.taglib
Webmerit/src/merit/util/taglib

EmpDetailTokenTag.java
This will be a new program written to safeguard against CSRF attacks (see TokenTag.java below). EmpDetailTokenTag.java will generate a hidden token on the two employee detail pages and will save the token in session. The employee detail pages are singled out as they can be opened in a separate window while the user remains on the roster page.

TokenTag.java
This will be a new program written to safeguard against CSRF attacks. TokenTag.java will generate a hidden token on each jsp that allows data entry and will save the token in session. TokenCheckFilter.java will look for the handlers of those jsps and compare the token on return to the token saved in session. If the tokens are not the same, the user will be shown a blank message page.

Web Pages
Webmerit/WebContent

RosterRangeBased.js
This is the javascript file for the range based rosters. Changes will be made in how the employee detail screen is opened to support the PRG approach.

RosterStepBased.js
This is the javascript file for the step based rosters. Changes will be made in how the employee detail screen is opened to support the PRG approach.

AdminCycleAddNew.jsp
This is the “Add New Merit Cycle” page off the Administration link on the Merit Menu. Changes will be made to support the addition of tokens to plug security vulnerabilities.
AdminCycleEdit.jsp
This is the “Add New Merit Cycle” page off the Administration link on the Merit Menu. Changes will be made to support the addition of tokens to plug security vulnerabilities.

AdminCycleList.jsp
This is the “Merit Cycle Administration” page. Changes will be made to support the addition of tokens to plug security vulnerabilities.

AdminCycleUpdateStatus.jsp
This is the “Update Cycle Status” page for Merit Administration. Changes will be made to support the addition of tokens to plug security vulnerabilities.

AdminPerfEval.jsp
This is the “Performance Evaluation Table” page for Merit Administration. Changes will be made to support the addition of tokens to plug security vulnerabilities.

AdminPerfRatingRange.jsp
This is the “Range Performance Rating Table” page for Merit Administration. Changes will be made to support the addition of tokens to plug security vulnerabilities.

AdminPerfRatingStep.jsp
This is the “Step Performance Rating Table” page for Merit Administration. Changes will be made to support the addition of tokens to plug security vulnerabilities.

DisplayCbuc.jsp
This page is displayed when the user clicks “?” to display the list of code values for collective bargaining units. A change will be made to change the value for “ALL” from “@@” to “ZZ”.

DisplayRangeRateTable.jsp
This page is displayed when the user clicks a title code link on a range based roster. Changes will be made to support the addition of tokens to plug security vulnerabilities.

DisplayStepRateTable.jsp
This page is displayed when the user clicks a title code link on a step based roster. Changes will be made to support the addition of tokens to plug security vulnerabilities.

DisplaySubLoc.jsp
This page is displayed when the user clicks “?” to display the list of code values for sublocation. A change will be made to change the value for “ALL” from “@@” to “ZZ”.

Error.jsp
When there is an unexpected system error, this page is displayed and currently has a link to see the stack trace. Changes will be made to display a reference number for errors and not display the stack trace.
MsgDisplay.jsp
This is the generic message display page. Page will be modified to display no message if the message parameter is empty. Currently the page will display a message saying the message is null if there is no message.

RosterRangeBased.jsp
This is the “Range Based Merit Roster” page for Merit Roster Selection. Changes will be made to support the addition of tokens to plug security vulnerabilities. In addition, changes will be made to make search names (used for previous and next pages) positional parameters.

RosterRangeEmpDetail.jsp
This is the “Range Based Roster Employee Detail” page. Changes will be made to support the addition of tokens to plug security vulnerabilities.

RosterStepBased.jsp
This is the “Step Based Merit Roster” page for Merit Roster Selection. Changes will be made to support the addition of tokens to plug security vulnerabilities. In addition, changes will be made to make search names (used for previous and next pages) positional parameters.

RosterStepEmpDetail.jsp
This is the “Step Based Roster Employee Detail” page. Changes will be made to support the addition of tokens to plug security vulnerabilities.

ViewOnlyRosterRangeBased.jsp
This will be a new view-only version of the Range Based Merit Roster.

ViewOnlyRosterRangeEmpDetail.jsp
This will be a new view-only version of the Range Based Roster Employee Detail screen.

ViewOnlyRosterStepBased.jsp
This will be a new view-only version of the Step Based Merit Roster.

ViewOnlyRosterStepEmpDetail.jsp
This will be a new view-only version of the Step Based Roster Employee Detail screen.