

Release 1931

**Service Request 82583
IPEDS**

**Service Request 82806
Health Sciences Compensation Data**

**Service Request 82979
Add PPS Separation Reason Code to CPS Interface**

**Service Request 83016
Web New Hire – Add Appointment Dept Code**

**Error Reports 2276, 2279, 2280, 2281, 2283, 2285,
2286, 2287, 2288, 2290, 2293, 2297, 2307**

Detail Design

July 8, 2010
Prepared by Mary Meyer

Information Resources & Communications
Office of the President
University of California

Table of Contents

Introduction	1
Service Request 82583 – Race and Ethnicity Data Restructuring	1
Service Request 82806 – Health Science Compensation Data	1
Service Request 82979 – Add PPS Separation Reason Code to CPS Interface	1
Service Request 83016 – New Hire - Add Appointment Department Code	1
Error Reports	1
Overview of PPS Modifications	2
Initialization	2
EDB File Maintenance Process	2
Interface Files	2
Programs	3
New and Modified Programs	3
PPBASFET, PPBASUPD, PPBASHST	3
PPPERCHG, PPPERFET, PPPERUPD, PPPERUTL, PPPERUTW	3
PPEI740	3
PPP711	3
PS006	3
PS008	3
PS012	3
PPWEPD1	4
PPWIGEN	4
PPWHPER	4
PPWEHON	4
Copy Members	4
CPPDFKEY	4
CPWSRPER	4
CPWSRBAS	4
CPWSXECP	4
Include Members	4
PPPVBAS1, PPPVZBAS	4
PPPVPER1, PPPVPER2, PPPVZPER	4
DDL Members	5
EDB PPPPER Table	5
HDB PPPBAS Table	5
CICS Map Changes	6
PPEPD10	6
PPIGEN0	6
PPHPER0	6
CICS Help Updates	6
CICS Help Text	6
CICS Help Anchors	6
Forms	7
UPAY640A– MAINTENANCE WORKSHEET - 1	7
Control Table Updates	7
System Messages Table	7
Code Translation Table	7
Data Element Table	7
Data Element To Screens Table	8
History Control Table Updates	8
History Data Element Table	8
Data Dictionary Updates	8
EDB 0322 (new)	8
EDB 0323 (new)	8
EDB 0324 (new)	8

EDB 0325 (new)	8
EDB 0326 (new)	9
EDB 0327 (new)	9
EDB 0112 (changed)	9
Overview of PPS Web Modifications	9
PPSWebService Changes	9
EDBEmployee.java.....	9
MapBackedEDBEmpImpl.java	9
AppParamLookupFacadeImpl.java.....	9
CancelFacadeImpl.java	9
DocumentFacadeImpl.java	9
PanFacadeImpl.java.....	9
PPSWConEditFacadeImpl.java	10
PPSWRangeValueEditFacadeImpl.java	10
SuspendFacadeImpl.java	10
TemplateFacadeImpl.java.....	10
UserAuthorizationFacadeImpl.java	10
UserIdFacadeImpl.java	10
EDBInquiry Changes	10
Java Classes	10
AddressAction.java.....	10
PersonalAction.java	10
JSPs.....	10
personal_ppwiper.jsp	10
version.jsp.....	11
EDBUpdate New Hire Changes.....	11
Property Files.....	11
CampusCustomization.properties	11
Java Classes	11
Employee.java	11
ApplicationPropertiesLoader.java.....	11
CampusCustomizationData.java	11
CampusCustomizationDataImpl.java	11
DataElementMetadataManager.java	11
ITranslationService.java	11
JavaScripts	12
common.js.....	12
post-auth-notification.js	12
JSPs.....	12
add_appointment.jsp	12
show_appointment.jsp	12
conedit_rve.jsp.....	12
employee_identification.jsp.....	12
list_suspended_sessions.jsp	13
list_templates.jsp	13
personal_information.jsp	13
summary.jsp.....	13
pan_reviewers.jsp	13
CSS Files	13
New-hire.css	13
Help Files.....	13
0322.txt	13
0323.txt	13
0324.txt	13
0325.txt.....	14
0326.txt.....	14

0327.txt.....	14
0112.txt.....	14
2020.txt.....	14
2032.txt.....	14
Cross-Site Request Forgery (CSRF) Prevention in EDBUpdate	14

Introduction

Service Request 82583 – Race and Ethnicity Data Restructuring

Service Request 82583 requires a change to the collection and reporting of employee demographics to comply with new Federal standards for race and ethnicity data.

Service Request 82806 – Health Science Compensation Data

Service Request 82806 requires a change to establish reporting fields in PPS to capture data relevant to Health Sciences (HS) faculty. Phase I, changes to PPS, was completed under Release 1916. Phase II will be covered by this release to derive and move department type code, academic programmatic unit code, patient care flag, degree 1, degree 2 and degree 3 to the CPS file.

Service Request 82979 – Add PPS Separation Reason Code to CPS Interface

Service Request 82979 requires a change to pass the actual separation reason code (EDB0141) to the CPS file with no translation.

Service Request 83016 – New Hire - Add Appointment Department Code

Service Request 83016 requires a change to web new hire application to accept appointment department code as input at the appointment level. It requires this to be controlled by campuses using external property file.

Error Reports

The following error reports will be addressed in this release:

ER 2276 – This error report requires the Enlarge employee name fields on the employee identification screen of the web new hire application.

ER 2279 – This error report aligns the delete button under additional recipients on PAN notification screen of the web new hire application.

ER 2280 – This error report requires that the application retain incomplete field entry in various fields “masked” by a pattern in the web new hire application.

ER 2281 - This error report requires that the highlights be inconsistent on the summary screen of the web new hire application.

ER 2283- While testing SD new hire, the PAN screen was redisplayed with message “Could not complete hire...” After adding the update key to the ENPR screen in the UCOPFK table we were able to commit the update. In PS008, if there’s no update key available, it should cause an ABEND with useful messaging and logging. Internal documentation in PS008 should indicate that the update key must be available for the first function in the PANN bundle.

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

ER 2285 -When PS012 gets a fatal error condition from UCIIDASN, the soap message is "FATAL ERROR IN UCIIDASN". It would be helpful if this also included UCIDWSAS-RETURN-STAT which tells the nature of the error.

ER 2286 - This error report requires that the focus be on the "Appointment Type" field instead of "Title" when editing an appointment in the web new hire application.

ER 2287 - This error report requires proper formatting of the RVE screen in the web new hire application.

ER 2288 – This error report removes clickable columns from Select Recipient screen (PAN) in the web new hire application.

ER 2290 – This error report corrects the behavior of the tab key in modal dialogs in the web new hire application.

ER 2293 - This error report correct the "unexpected error" condition when there are no templates in the web new hire application.

ER 2297 – This error report requires fixes to the web new hire application related to a threat posed by a cross-site request forgery vulnerability.

ER 2307 – This error report requires fixes to the web EDB Inquiry application related to processing error when home phone number is not complete 10 digits.

Overview of PPS Modifications

Initialization

The newly created elements will not be populated for existing employees other than to initialize them to "blank" or "unknown" if necessary.

EDB File Maintenance Process

The EDB File Maintenance Process will be modified to include edits for six new elements Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. One new (user entered) value will be added for existing element Ethnic-Id. Additionally, if Ethnic-Id is not entered by user, an attempt will be made to derive the value based on the six new indicators.

Interface Files

The Corporate Personnel (CPS) Employee data interface file will be modified to include six new ethnicity indicators: Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, White-Ind as well as Department-Type-Code, Academic-Program-Unit-Code, Patient-Care-Flag, Degree-1, Degree-2, Degree 3 and Separation-Reason-Code.

Programs

New and Modified Programs

PPBASFET, PPBASUPD, PPBASHST

These programs are the utility programs for HDB table PPPBAS. They will be modified to include HISPANIC_IND, (EDB0322), AFRICAN_AMER_IND (EDB0323), AMER_INDIAN_IND (EDB0324), ASIAN_IND (EDB0325), NATIVE_HAWAI_IND (EDB0326), and WHITE_IND (EDB0327).

PPPERCHG, PPPERFET, PPPERUPD, PPPERUTL, PPPERUTW

These programs are the utility programs for EDB table PPPPER. They will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind

PPEI740

PPEI740 performs consistency edits and maintenance on ETHNIC_ID (EDB0112).

It will be updated so that before ethnic-id is set to 'unknown' in the current process, it will first try to derive a value for ethnic-id based on the values of the new indicators.

PPP711

PPP711 produces the CPS Interface File from the employees' data on the EDB. It will be modified to add two new fields for SR 82583.

The first new field will correspond to the Hispanic Indicator and will be passed as either 'Y', 'N', or spaces.

The second corresponds to the other five racial indicators and will pass a concatenated string of letters representing the indicators having code 'Y'.

For SR 82806, Department-Type-Code and Academic-Program-Unit-Code will be passed on the CPS file as well as these derived fields: Patient-Care-Flag, Degree-1, Degree-2 and Degree-3.

For SR 82979, the actual Separation-Reason-Code (with no translation) will be passed on the CPS file.

PS006

PS006 is the driver for the PS006 web service which handles consistency editing for web new hire. It invokes the consistency edits within PPS via calls to UCROUTER which mimic CICS functionality. It will be modified by inclusion of modified copy member CPPSFKEY.

PS008

PS008 is a web service program which commits an EDB update which is already in progress. It will be modified to contain section 9999-ABEND which is required by modified copy member CPPSFKEY.

PS012

PS012 is a web service program which calls the appropriate PPS modules to assign a new employee ID, or to use an existing IID system ID for a new employee. It will be modified to display content of ucidwsas-return-stat when ucidwsas-fatal-error is true, as aid in debug.

PPWEPD1

PPWEPD1 is the screen processor for the EPD1function (Employee Personal Data 1) It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind

PPWIGEN

PPWIGEN is the screen processor for the IGEN function (General Information 1). It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind and White-Ind.

PPWHPER

PPWHPER is the screen processor for the HPER function (Personal Data History). It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind and White-Ind.

PPWEHON

PPWEHON is the screen processor for the EHON function (Honors Data). It will be modified to left justify the honor-type-code field (EDB0713).

Copy Members

CPPDFKEY

CPPDFKEY is procedural code to set the logical function used to update. It will be modified to abend if perform of set-function-key does not return any aid key.

CPWSRPER

CPWSRPER defines a row of the PPPPER EDB table. It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

CPWSRBAS

CPWSRBAS defines a row of the PPPBAS HDB table. It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

CPWSXECP

CPWSXECP defines the record layout for the CPS Employee file. It will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind, Department-Type-Code, Academic-Program-Unit-Code, Patient-Care-Flag, Degree-1, Degree-2, Degree 3 and Separation-Reason-Code.

Include Members

PPPVBAS1, PPPVZBAS

These INCLUDE members are the record layouts for the standard views of the PPPBAS table. They will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

PPPVPER1, PPPVPER2, PPPVZPER

These INCLUDE members are the record layouts for the standard views of the PPPPER table. They will be modified to include Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

DDL Members

EDB PPPPER Table

The following columns will be added to the PPPPER table:

- HISPANIC_IND (CHAR(1) NOT NULL WITH DEFAULT)
- AFRICAN_AMER_IND (CHAR(1) NOT NULL WITH DEFAULT)
- AMER_INDIAN_IND (CHAR(1) NOT NULL WITH DEFAULT)
- ASIAN_IND (CHAR(1) NOT NULL WITH DEFAULT)
- NATIVE_HAWAI_IND (CHAR(1) NOT NULL WITH DEFAULT)
- WHITE_IND (CHAR(1) NOT NULL WITH DEFAULT)

The following DDL member will contain an ALTER statement for the new columns:

- TBP20A (new)

The following DDL member defines the PPPPER table and will incorporate the new columns:

- TBP00C

The following view DDL members will incorporate the new columns:

- PPPVPER1
- PPPVPER2
- PPPVZPER

HDB PPPBAS Table

The following columns will be added to the PPPBAS table:

- HISPANIC_IND (CHAR(1) NOT NULL WITH DEFAULT)
- AFRICAN_AMER_IND (CHAR(1) NOT NULL WITH DEFAULT)
- AMER_INDIAN_IND (CHAR(1) NOT NULL WITH DEFAULT)
- ASIAN_IND (CHAR(1) NOT NULL WITH DEFAULT)
- NATIVE_HAWAI_IND (CHAR(1) NOT NULL WITH DEFAULT)
- WHITE_IND (CHAR(1) NOT NULL WITH DEFAULT)

The following DDL member will contain an ALTER statement for the new columns:

- TBBAS14A (new)

The following DDL member defines the PPPBAS table and will incorporate the new columns:

- TBBAS00C

The following view DDL members will incorporate the new columns:

- PPPVBAS1
- PPPVZBAS

CICS Map Changes

PPEPD10

PPEPD10 is the CICS map for the screen function EPD1 - Employee Personal Data 1. It will be modified to add appropriate labels and data fields for entering and updating Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. Some existing elements will be rearranged to accommodate the new data.

PPIGEN0

PPIGEN0 is the CICS map for the screen function IGEN - General Information 1. It will be modified to add appropriate labels and data fields for displaying Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. Some existing elements will be rearranged to accommodate the new data.

PPHPER0

PPHPER0 is the CICS map for the screen function HPER - Personal Data History. It will be modified to add appropriate labels and data fields for displaying Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. Some existing elements will be rearranged to accommodate the new data.

CICS Help Updates

CICS Help Text

New CICS Help texts for the following will be created:

- EDB 0322 (Hispanic-Ind)
- EDB 0323 (African-American Ind)
- EDB 0324 (Amer-Indian-Ind)
- EDB 0325 (Asian-Ind)
- EDB 0326 (Native-Hawai-Ind)
- EDB 0327 (White-Ind)
- New group help for the ethnic indicators (EDB 0323 through EDB 0327) and the combination of EDB 0112 and EDB 0108 will be created

Existing CICS Help text for EDB 0112 (Ethnic-Id) will be modified.

CICS Help Anchors

CICS Help anchors will be adjusted on the following screens:

- EPD1: The Help text associated with EDB 0322 will be anchored to the Hispanic-Ind label and data fields, Help text associated with EDB 0323 will be anchored to the African-Amer-Ind label and data fields, Help text associated with EDB 0324 will be anchored to the Amer-Indian-Ind label and data fields, Help text associated with EDB 0325 will be anchored to the Asian-Ind label and data fields, Help text associated with EDB 0326 will be anchored to the Native-Hawai-Ind label and data fields, Help text associated with EDB 0327 will be anchored to the White-Ind label and data fields. Help text for rearranged elements will be re-anchored as needed.
- IGEN: The Help text associated with EDB 0322 will be anchored to the Hispanic-Ind label and data fields, Help text associated with EDB 0323 will be anchored to the African-Amer-Ind label and data fields, Help text associated with EDB 0324 will be anchored to the Amer-Indian-Ind label and data fields, Help text associated with EDB 0325 will

be anchored to the Asian-Ind label and data fields, Help text associated with EDB 0326 will be anchored to the Native-Hawai-Ind label and data fields, Help text associated with EDB 0327 will be anchored to the White-Ind label and data fields. Group Help text associated with EDB 0112 and EDB 0108 will be anchored to the Ethnic-Id and Sex-Code label and data fields. Group help associated with EDB 0322 through EDB 0327 will be anchored to the Ethnic/Race indicator label and data fields. Help text for rearranged elements will be re-anchored as needed.

- HPER: The Help text associated with EDB 0322 will be anchored to the Hispanic-Ind label and data fields, Help text associated with EDB 0323 will be anchored to the African-Amer-Ind label and data fields, Help text associated with EDB 0324 will be anchored to the Amer-Indian-Ind label and data fields, Help text associated with EDB 0325 will be anchored to the Asian-Ind label and data fields, Help text associated with EDB 0326 will be anchored to the Native-Hawai-Ind label and data fields, Help text associated with EDB 0327 will be anchored to the White-Ind label and data fields. Group help associated with EDB 0322 through EDB 0327 will be anchored to the Ethnic/Race indicator label and data fields. Help text for rearranged elements will be re-anchored as needed.

Forms

UPAY640A– MAINTENANCE WORKSHEET - 1

New indicators will be added in columns 29 – 34

HR will update form U5605- (refer to attachment K in SR)

Control Table Updates

System Messages Table

The following messages will be added to this table.

New Messages

08-349– (NO RACIAL INDICATOIRES ARE ‘Y’ – RECORDING NON-RESPONSE)- Severity Level of Warning for online.

12-349– (NO RACIAL INDICATOIRES ARE ‘Y’ – RECORDING NON-RESPONSE)- Severity Level of Warning for online.

Modified Messages

08-191 and 12-191 will be modified to read “ETHNIC ORIGIN FIELD IS BLANK, WILL BE DERIVED”.

Code Translation Table

Existing EDB data element ETHNIC_ID (EDB 0112) will have new code of “Z” and its translation will be added on this table as well as new derived codes O, P, S V, Q and T. Translations will be updated for codes A, 2 and B.

Data Element Table

The following new EDB data elements will be defined, and associated with the PPPPER Table.

- Hispanic-Ind (EDB 0322)
- African-American Ind (EDB 0323)
- Amer-Indian-Ind (EDB 0324)
- Asian-Ind (EDB 0325)

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

- Native-Hawai-Ind (EDB 0326)
- White-Ind (EDB 0327)

The new code of “Z” will be added to data element EDB 0112.

Data Element To Screens Table

The following new EDB data elements associated with the EPD1, HPER and IGEN screens will be defined on this table.

- Hispanic-Ind (EDB 0322)
- African-American Ind (EDB 0323)
- Amer-Indian-Ind (EDB 0324)
- Asian-Ind (EDB 0325)
- Native-Hawai-Ind (EDB 0326)
- White-Ind (EDB 0327)

History Control Table Updates

History Data Element Table

The following Data Elements will be added to the History Data Element (HDE) Table.

- Hispanic-Ind (EDB 0322)
- African-American Ind (EDB 0323)
- Amer-Indian-Ind (EDB 0324)
- Asian-Ind (EDB 0325)
- Native-Hawai-Ind (EDB 0326)
- White-Ind (EDB 0327)

Data Dictionary Updates

The following new Data Dictionary entries will be added:

EDB 0322 (new)

Hispanic-Ind

EDB 0323 (new)

African-Amer-Ind

EDB 0324 (new)

Amer-Indian-Ind

EDB 0325 (new)

Asian-Ind

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

EDB 0326 (new)

Native-Hawai-Ind

EDB 0327 (new)

White-Ind

EDB 0112 (changed)

Ethnic-Id

Overview of PPS Web Modifications

The PPS Web Services will be modified to accept Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind from EDB.

The EDB Inquiry system will be modified to display Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

The EDB Updates New Hire system will be modified to provide for entry of Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. This system will also be modified to address various error reports.

PPSWebService Changes

EDBEmployee.java

This is the interface class for all of the employee related properties. It will be modified to include new properties related to Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

MapBackedEDBEmpImpl.java

This is the implementation of EDBEmployee used in the PPS web applications. It will be modified to include new properties related to Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

AppParamLookupFacadeImpl.java

This is application parameters lookup facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

CancelFacadeImpl.java

This is cancel transaction facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

DocumentFacadeImpl.java

This is document facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

PanFacadeImpl.java

This is PAN facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

PPSWConEditFacadeImpl.java

This is consistency edit facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

PPSWRangeValueEditFacadeImpl.java

This is range value edit facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

SuspendFacadeImpl.java

This is suspend transaction facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

TemplateFacadeImpl.java

This is template facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

UserAuthorizationFacadeImpl.java

This is user authentication facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

UserIdFacadeImpl.java

This is user ID facade implementation. This will be modified to remove extraneous logging of messages. This is part of ER2293.

EDBInquiry Changes

Java Classes

AddressAction.java

This is Struts' action class related to address screen. It will be changed to format the home phone number only if all 10 digits all present. This is part of ER2307.

PersonalAction.java

This is Struts' action class related to personal screen. It will be changed to format the home phone number only if all 10 digits all present. This is part of ER2307.

JSPs

personal_ppwwiper.jsp

This is the html format for displaying personal data screen. This screen will be modified to display Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind. The lable for "Ethnicity" will be changed to "Primary Ethnic/Race".

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

version.jsp

This holds the current EDBInquiry version number. This will be updated to the next version number.

EDBUpdate New Hire Changes

Property Files

CampusCustomization.properties

A new property (displayApptDepartmentCode) will be added to control hide/display of the appointment department code on the appointment screen.

The default will be “false” (hide).

Java Classes

Employee.java

This is employee object.

A getter and setter methods will be added for the new properties related to Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind.

ApplicationPropertiesLoader.java

This class loads all of the properties and builds various drop-downs, radio buttons, auto complete maps.

A new drop down map will be created for the new ethnicity indicator with the value – Blank, Yes and No.

The new campus customization property, displayApptDepartmentCode, will be read from the external customization file.

CampusCustomizationData.java

This is the interface class for campus customization data.

A getter and setter methods will be added for the new property - displayApptDepartmentCode.

CampusCustomizationDataImpl.java

This is java class to hold campus customization data.

A getter and setter methods will be added for the new property - displayApptDepartmentCode.

DataElementMetadataManager.java

This is java object for managing meta data information related to new hire system.

This class will be modified to include the meta data information related to the new data elements (Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind).

ITranslationService.java

This is java class to provide translation services for code/value pair.

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

New constants will be added related to the new data elements (Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind).

JavaScripts

common.js

This is common JavaScript file for the new hire application.

Masking for number fields will be changed so that partially entered values are not wiped out. This is part of ER2280. The following field will be impacted with this change:

- SSN
- All dates
- Home phone number
- Campus phone numbers
- All zip codes
- Year education completed
- Appointment number
- Distribution number
- Next salary review date

Summary screen color for blank field will be made consistent. This is part of ER2281.

post-auth-notification.js

This is JavaScript pertaining to PAN screen.

This will be changed to remove column sorting. This is part of ER2288.

JSPs

add_appointment.jsp

This is the JSP page for adding/editing an appointment.

This will be changed to include conditional appointment department code.

This will also be changed to focus cursor on “appointment type” instead of “appointment title. This is part of ER2286.

show_appointment.jsp

This is the JSP page for displaying an appointment.

This will be changed to include conditional appointment department code.

conedit_rve.jsp

This is the JSP page for displaying range value edit errors.

This will be changed to format the display in a more presentable way. This is part of ER2287.

employee_identification.jsp

This is the JSP page for entering employee identification data.

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

This will be changed to make the name fields longer. This is part of ER2276.

list_suspended_sessions.jsp

This is the JSP page for listing all suspended transactions.

This will be change to remove “tabIndex” for individual fields for proper tabbing. This is part of ER2290.

list_templates.jsp

This is the JSP page for listing all user templates.

This will be change to remove “tabIndex” for individual fields for proper tabbing. This is part of ER2290.

personal_information.jsp

This is the JSP page for entering personal information data.

New data elements, Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind, will be added. Title for “ethnicity” will be changed to “Primary ethnic/race”.

summary.jsp

This is the JSP page for displaying summary data.

New data elements, Hispanic-Ind, African-Amer-Ind, Amer-Indian-Ind, Asian-Ind, Native-Hawai-Ind, and White-Ind, will be added. Conditionally, appointment department code will be displayed.

pan_reviewers.jsp

This is the JSP page for displaying PAN reviewers table.

This will be changed to align the “delete” button properly for the additional PAN reviewers. This is part of ER2289.

CSS Files

New-hire.css

This is the CSS file for the new hire system.

Additional style will be added to align the “delete” button properly for the additional PAN reviewers. This is part of ER2289.

Help Files

0322.txt

This will be Hispanic Indicator help file.

0323.txt

This will be Black/African American Indicator help file.

0324.txt

This will be American Indian or Alaska Native Indicator help file.

Release 1931
IPEDS - Race and Ethnicity Data Restructuring, SR82583
Detail Design
July 8, 2010

0325.txt

This will be Asian Indicator help file.

0326.txt

This will be Native Hawaiian or Other Pacific Islander Indicator help file.

0327.txt

This will be White Indicator help file.

0112.txt

This is help file for Primary Ethnic/Race. It will be modified to change the header.

2020.txt

This is help file for appointment type. It will be modified to remove unrecognized characters.

2032.txt

This will be appointment department code help file.

Cross-Site Request Forgery (CSRF) Prevention in EDBUpdate

The following changes will be made to new hire application to address ER2297. These changes encompass all JSPs and most of the JavaScripts using AJAX calls.

The new hire application will use the commonly recommended synchronizer token pattern to protect itself from CSRF attacks. In general, this pattern involves adding a hard-to-guess randomly generated string token with each page that submits data that could be the focus of an attack. The token passed by the browser is matched against the expected token, and if they do not match, the request is rejected. This token is refreshed on a per-request basis.

The application will implement this pattern by leveraging off of an existing Struts interceptor, `TokenSessionStoreInterceptor`, which implements the unique-token-per-request as described above.

The interceptor will be configured to run near the top of the stack so that the framework will not set any parameters on the cached data before having validated the request. This configuration is done in the struts configuration file (`newhire.xml`). The interceptor configuration supports globally included and excluded methods, and included and excluded methods at the individual action mapping level. The application will globally include any `update()` method, adding individually included methods as needed at the action level.

For basic form submits, for example, all navigation away from a view that contains input elements, the token data will be embedded in the form as hidden fields, using the struts `<s:token>` tag. This tag writes two hidden fields, `struts.token.name` and `struts.token`. Both are required. When the browser submits the form, the token fields are automatically included in the request data.

In some XHR cases, the application uses the `jQuery.ajaxSubmit` function, which submits all data fields on the referenced form. This function offers a "data" option, which allows passing additional parameters. For requests of this type that are protected by the CRSF token scheme, the application will this data option, retrieving the token data as a JSON object.

For all XHR requests that are not form submits, the application will find the current token, append it to the request data, and return a fresh token with every response. The application will use a variety of patterns to accomplish this.

The javascript file `common.js` will contain one new constant and four utility functions to help with token handling:

```
var STRUTS_TOKEN_NAME_KEY = 'struts.token.name';

//get the token data as querystring params
//i.e.: struts.token.name=struts.token&struts.token=[UID]
function getSessionTokenQS()

//get the token data as a JSON object
//i.e.:
//{ "struts.token.name" : "struts.token", "struts.token" : [UID]}
function getSessionTokenJSON()

//retrieve the refreshed token from the message
//which contains it as a JSON string
function updateSessionTokenFromJSON( html )

//retrieve the refreshed token from hidden
//input elements on the current DOM
function updateSessionTokenFromHiddenField()
```

Request Patterns

Any view that contains employee-level input fields will embed the token somewhere in its template using the `<s:token/>` tag. The token data will then, automatically, be included in the request when the user navigates to another view. The basic form pattern will be used on:

- Personal Information
- Address & Disclosures
- Citizenship & Taxes
- Employment Information
- Education*
- Review Messages and Update
- Post Authorization Notification: Submit to PPS**

*The education template will embed the token in its licenses tile (`licenses_and_certs.jsp`), since that is what will be returned as an HTML fragment when a license is added or deleted, making the refresh in the XHR context simpler.

**The PAN template will embed the token in its `pan_reviewer` tile, since that is what will be returned as an HTML fragment when adding and removing additional reviewers.

For XHR requests that use a querystring, or HTML as the data option, the application will call the utility function that finds the token name and token value elements on the page, and will return them as a concatenated querystring.

For XHR requests that use JSON as the data option, the application will use the utility function that returns the token data as a JSON object. XHR calls that use this method, will append this data to the existing JSON object via `jQuery.extend(existingJSONObj, tokenJSONObj)`.

In some cases, the javascript makes a new request by setting the `window.location` property. In these cases, it will attach the token data as querystring parameters to the URL that is set on the `location` field.

Chain pattern

Some actions, such as restoring a transaction, and proceeding to PAN from conedits, forward to another action on the same request. These redirects need to handle the token in a special manner as well. The source action must generate a new token, and include the token parameters in its action mapping as a parameter to the next action, so that the framework will include the data as querystring parameters in the redirect response header. The base action class will include a utility method, `setSessionTokenForRedirect()` which will generate the new token and stores it in a form field for reference by the mapping parameter. Occurrences of this pattern will include:

- Restore → Employee Entry
- Conedit Proceed to PAN → Submit Employee
- PAN → Submit Employee

Response Patterns

Some XHR actions return text fragments. In these cases, the new token will be appended to the response message as a JSON string. The application will call the update from JSON function to parse the string as JSON before setting the token input element with the new value.

Other XHR actions return HTML fragments which are displayed using DHTML. If the fragment makes up part of the main page view, such as the license table on the education view, then the new token data will be embedded using the `<s:token/>` tag, and no further processing is required by the XHR handler. In other cases, the fragment will not be part of the main page, such as when it is meant as the content for a modal. In these cases, the new token value will be stored as a hidden input with a different key, "newSessionToken". The XHR handler must call the `updateSessionTokenFromHiddenField` utility in order to set the new value on the existing input with the real key, `struts.token`.

Non-form Views

Views without a form, such as the summary or the transaction completed views, must also embed the token, since a user can restore a transaction from those views, and the restore action will be protected by the token scheme. The utility functions in `common.js` will find and set the token for these pages as well.

Implementation Details

Employee Entry

Action:	User fills in name, DOB, SSN, then clicks "Next".
Request Type:	XHR
JavaScript:	employee_identification.js
Function:	assignNewID
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	No token refresh needed, since the personal info view loads, which will have token embedded in its form.
Failure:	<p>Hidden field/html fragment pattern Existing EDB match Matches found Existing ID match Validation errors</p> <p>JSON pattern ID no longer available</p>

Action:	User selects existing match.
Request Type:	XHR
JavaScript:	employee_identification.js
Function:	useExistingId
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	No token refresh needed, since the personal info view loads, which will have token embedded in its form.
Failure:	<p>JSON pattern ID no longer available.</p>

Action:	User chooses to assign new ID when matches exist.
Request Type:	XHR
JavaScript:	employee_identification.js
Function:	createNewIdRegardless
Token source:	Page form hidden inputs.
Passed:	Added as JSON data option to \$.ajax().
Success:	No token refresh needed, since the personal info view loads, which will have token embedded in its form.
Failure:	Only backend unrecoverable errors can occur, which lead to auto-suspension.

Templates

Action:	User creates new template.
Request Type:	XHR
JavaScript:	templates.js
Function:	saveTemplate
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token appended to XHR response as JSON string.
Failure:	New token appended to XHR response as JSON string.

Action:	User uses a template.
Request Type:	XHR
JavaScript:	templates.js
Function:	useTemplate
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token appended to XHR response as JSON string, however the utility function is not called, since the template data is returned as a JSON string, so the new token is already parsed at this point. The token input value is set directly here.
Failure:	New token appended to XHR response as JSON string.

Action:	User deletes a template.
Request Type:	XHR
JavaScript:	templates.js
Function:	deleteTemplate
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token embedded in HTML fragment (full table of existing templates).
Failure:	New token appended to XHR response as JSON string.

Licenses

Action:	User adds a license.
Action Class & Method	Education.addLicense
Request Type:	XHR
JavaScript:	education.js
Function:	openAddLicenseDialog
Token source:	Page form hidden inputs.
Passed:	JSON object passed as data option to ajaxSubmit().
Success:	New token embedded in HTML fragment (full table of existing templates, licenses_and_certs.jsp).
Failure:	New token embedded in HTML fragment (marked up error message or range value div).

Action:	User deletes a license.
Action Class & Method	Education.deleteLicense
Request Type:	XHR
JavaScript:	education.js
Function:	confirmDeleteLicense
Token source:	Page form hidden inputs.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token embedded in HTML fragment (full table of existing templates).
Failure:	JSON string appended to error msg (ajax_error.jsp).

Action:	User edits a license.
Action Class & Method	Education.editLicense
Request Type:	XHR
JavaScript:	education.js
Function:	openAddLicenseDialog
Token source:	Page form hidden inputs.
Passed:	JSON object passed as data option to ajaxSubmit().
Success:	New token embedded in HTML fragment (full table of existing templates).
Failure:	JSON string appended to error msg (ajax_error.jsp).

Appointments

Action:	User submits an added or edited appointment.
Action Class & Method	EditAppointmentAction.addToEmpl
Request Type:	XHR
JavaScript:	appts-and-dists.js
Function:	Save
Token source:	Hidden inputs embedded in the appointment_table.jsp tile of the main page.
Passed:	JSON object passed as data option to ajaxSubmit().
Success:	New token embedded in HTML fragment (full table of existing appointments/distributions).
Failure:	New token embedded in the add/edit appointment form template (add_appointment.jsp), keyed by newSessionToken, and refreshed on the main page by updateSessionTokenFromHiddenField().

Action:	User deletes an existing appointment.
Action Class & Method	EditAppointmentAction.remove
Request Type:	XHR
JavaScript:	appts-and-dists.js
Function:	displayCRUD
Token source:	Hidden inputs embedded in the appointment_table.jsp tile of the main page.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token embedded in HTML fragment (full table of existing appointments/distributions).
Failure:	N/A

Distributions

Action:	User submits an added or edited distribution.
Action Class & Method	EditDistributionAction.addToEmpl
Request Type:	XHR
JavaScript:	appts-and-dists.js
Function:	Save
Token source:	Hidden inputs embedded in the appointment_table.jsp tile of the main page.
Passed:	JSON object passed as data option to ajaxSubmit().
Success:	New token embedded in HTML fragment (full table of existing appointments/distributions).
Failure:	New token embedded in the add/edit distribution form template (add_distribution.jsp), keyed by newSessionToken, and refreshed on the main page by updateSessionTokenFromHiddenField().

Action:	User deletes an existing distribution.
Action Class & Method	EditDistributionAction.Remove
Request Type:	XHR
JavaScript:	appts-and-dists.js
Function:	displayCRUD
Token source:	Hidden inputs embedded in the appointment_table.jsp tile of the main page.
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	New token embedded in HTML fragment (full table of existing appointments/distributions).
Failure:	N/A

Suspend Transaction

Action:	User suspends an active transaction.
Action Class & Method	SuspendAction.suspend
Request Type:	XHR
JavaScript:	modals.js
Function:	bindSuspendAndExitLink
Token source:	Main page inputs, either embedded in the update form or on the page if no form present.
Passed:	On views with an update form, which will contain the token inputs, the form is submitted programmatically in the JS function; on views without an update form (summary, transaction completed, employee entry display only), the token inputs are read using getSessionTokenQS and then appended to the URL set programmatically on window.location.
Success:	No refresh needed.
Failure:	Token embedded in error view, employee entry display.

Action:	User deletes a suspended transaction.
Action Class & Method	SuspendAction.delete
Request Type:	XHR
JavaScript:	modals.js
Function:	confirmDeleteSession
Token source:	Main page inputs, either embedded in the update form or on the page if no form present.
Passed:	JSON bind option bind object extended by return value from getSessionTokenJSON when binding the click handlers in bindIWantToRestoreButtons.
Success:	Token embedded suspended transaction list template (list_suspended_templates.jsp); the action method (RestoreAction.list() tests for the presence of the token in the session, and sets a new one if it is null).
Failure:	Appended to ajax error message, refreshed via updateSessionTokenJSON.

Restore Transaction

Action:	User restores an action.
Action Class & Method	RestoreAction.execute
Request Type:	XHR
JavaScript:	modals.js
Function:	restoreHandler
Token source:	Main page inputs, either embedded in the update form or on the page if no form present.
Passed:	JSON data bind option object extended by return value from getSessionTokenJSON when binding the click handlers in bindIWantToRestoreButtons.
Success:	Chain pattern: this action redirects to the EmployeeEntryAction. Since restoring a transaction requires reissuing an employee ID, the restore action forwards to the employee entry action on a successfully restore. Action method generates a new token, stores it in its currentSessionToken, by calling the superclass method NewhireBaseAction.setSessionTokenForRedirect(). The token fields are specified as additional parameters to the actionRedirect mapping that the restore action forwards to, so that framework will find the token data when it does its check on the protected EmployeeEntryAction.checkForMatch() call.
Failure:	Not needed.

Conedits

Action:	User fixes a conedit problem on the conedit view.
Action Class & Method	ConsistencyCheckAction.update
Request Type:	XHR
JavaScript:	review.js
Function:	submitConeditFix
Token source:	Main page inputs (conedit_messages.jsp).
Passed:	JSON object passed as data option to ajaxSubmit().
Success:	New token embedded in HTML fragment (conedit_messages.jsp).
Failure:	<p>RVE before invoking conedits: Error message fragment (conedit_rve.jsp) contains <s:token/> tag.</p> <p>General error: Error messgae fragment contains <s:token/> tag; old html containing token replaced by this message fragment.</p> <p>Field level RVE: new token returned as hidden input in html fragment (iid_validation_error.jsp) keyed as newSessionToken, real token input refreshed via updateSessonTokenFromHiddenField.</p>

Action:	User proceeds to PAN.
Action Class & Method	SubmitEmployeeAction.execute
Request Type:	XHR
JavaScript:	review.js
Function:	bindSubmitEmpButton
Token source:	Main page inputs (conedit_messages.jsp).
Passed:	JS binds event handler to button that concatenated to URL as querystring, which is set on window.location.
Success:	Chain pattern: this action redirects to the PAN action. SubmitEmpAction method generates a new token, stores it in its currentSessionToken, by calling the superclass method NewhireBaseAction.setSessionTokenForRedirect(). The token fields are specified as additional parameters to the actionRedirect mapping that the restore action forwards to, so that framework will find the token data when it does its check on the protected PAN action method.
Failure:	Redirects back to conedit, where the main page contains the token.

PAN

Action:	User adds reviewer by ID.
Action Class & Method	PostAuthNotificationAction.lookupByUserId
Request Type:	XHR
JavaScript:	post-auth-notification.js
Function:	bindLookupByUIDButton
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	pan_reviewers.jsp returned as HTML fragment replacing existing table DOM elements.
Failure:	JSON string appended to error msg (ajax_error.jsp) (action method must refresh token in these cases).

Action:	User adds reviewer by email.
Action Class & Method	PostAuthNotificationAction.addReviewer
Request Type:	XHR
JavaScript:	post-auth-notification.js
Function:	postSelectedRvwr0
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	pan_reviewers.jsp returned as HTML fragment replacing existing table DOM elements.
Failure:	JSON string appended to error msg (ajax_error.jsp) (action method must refresh token in these cases).

Action:	User adds reviewer by selecting item from name grid.
Action Class & Method	PostAuthNotificationAction.addReviewer
Request Type:	XHR
JavaScript:	post-auth-notification.js
Function:	postSelectedRvwr0
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	pan_reviewers.jsp returned as HTML fragment replacing existing table DOM elements.
Failure:	N/A

Action:	User deletes an additional reviewer.
Action Class & Method	PostAuthNotificationAction.removeReviewer
Request Type:	XHR
JavaScript:	post-auth-notification.js
Function:	postSelectedRvwr0
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	Concatenated to querystring that becomes data option to \$.ajax().
Success:	pan_reviewers.jsp returned as HTML fragment replacing existing table DOM elements.
Failure:	N/A

Action:	User submits employee to PPS.
Action Class & Method	PostAuthNotificationAction.submit
Request Type:	XHR
JavaScript:	post-auth-notification.js
Function:	javacript line embedded in button's href attribute, which submits the page's form.
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	Normal form submit.
Success:	Chain pattern: this action redirects to the SubmitEmployee action. The PAN.submit method generates a new token, stores it in its currentSessionToken, by calling the superclass method NewhireBaseAction.setSessionTokenForRedirect(). The token fields are specified as additional parameters to the actionRedirect mapping forwarding to the submit action, so that framework will find the token data when it does its check on the protected submit emp hire action.
Failure:	PAN view is reloaded, which contains a fresh token in its form, via the <s:token/> tag in its pan_reviewers.jsp tile.

IDOC

Action:	User confirms IDOC.
Action Class & Method	ConfirmIdocAction.execute
Request Type:	XHR
JavaScript:	idoc.js
Function:	bindConfirmIdocButton
Token source:	Main page inputs (pan_reviewers.jsp tile).
Passed:	JS binds event handler to button that concatenated to URL as querystring, which is set on window.location.
Success:	Token included in confirmation page's template via the <s:token/> tag.
Failure:	N/A