

Release 1873

**Service Request 82355
EDB Inquiry Java Rewrite**

Detailed Design

July 21, 2009
Prepared by Maxine Gerber
Peter Parker
Nancy Field
Rajeev Gupta
Damien Smith

Information Resources & Communications
Office of the President
University of California

Table of Contents

Introduction	1
Service Request 82355.....	1
Overview of PPS Modifications	1
PPS CICS Regions and New Batch Process	1
CICS Web Services Programs	1
Web Artifacts.....	1
Web Merit Changes	1
Web PAN Changes	1
PPSWeb Appplication (new).....	1
EDBInquiry Application (new).....	2
PPSWebServices Application (new).....	2
Differences Between Old and New Application.....	2
Security and Session Control in the Java PPS Applications	3
Certificates.....	3
Tokens	3
Session Timeout.....	3
CICS Web Services.....	3
PS001 (new).....	3
PS003 (new).....	3
PS004 (new).....	3
PS005 (new).....	3
PS010 (new).....	4
PS011 (new).....	4
PS014 (new).....	4
PS016 (new).....	4
PS020 (new).....	4
Cobol Programs	4
New and Modified Programs	4
PPDELTK (new).....	4
PPFCTSPR (new)	4
PPHMESPR (new).....	5
PPMSGSPR (new).....	5
PPSPBSPR (new)	5
PPWPLT (new).....	5
PPWWUUSE (new).....	5
PS001 (new).....	6
PS003 (new).....	6
PS004 (new).....	7
PS004D (new).....	8
PS005 (new).....	8
PS005D (new).....	8
PS010 (new).....	8
PS010D (new).....	9
PS011 (new).....	9
PS014 (new).....	9
PS014D (new).....	9
PS016 (new).....	9
PS016D (new).....	10
PS020 (new).....	10
UCFNAUTH.....	10
UCROUTER.....	11
UCSSIMNT (new).....	11
UCWMMNU	11

Copy Members	11
PS001 (new).....	11
PS001I01 (new)	11
PS001O01 (new).....	11
PS003 (new).....	12
PS003I01 (new)	12
PS003O01 (new).....	12
PS004 (new).....	12
PS005 (new).....	12
PS010 (new).....	12
PS011 (new).....	12
PS011I01 (new)	12
PS011O01 (new).....	12
PS014 (new).....	12
PS016 (new).....	13
PS020 (new).....	13
PS020I01 (new)	13
PS020O01 (new).....	13
UCCOMMON	13
UCPIAUTH.....	13
DDL Members	14
SPFCTSPR (new)	14
SPHMESPR (new).....	14
SPMSGSPR (new).....	14
SPSPBSPR (new)	14
SPUSE00C (new)	14
Bind Members	14
New and Modified Binds.....	14
PPFCTSPR (new)	14
PPHMESPR (new).....	14
PPMSGSPR (new).....	14
PPSPBSPR (new)	14
PPWUUSE (new).....	14
PS001 (new).....	14
PS003 (new).....	15
PS005 (new).....	15
PS011 (new).....	15
PS014 (new).....	15
UCSSIMNT (new).....	15
Web Service Definitions (wsdl)	15
PS001.wsdl (new)	15
PS003.wsdl (new)	15
PS004.wsdl (new).....	15
PS005.wsdl (new).....	15
PS010.wsdl (new).....	15
PS011.wsdl (new).....	15
PS014.wsdl (new).....	16
PS016.wsdl (new).....	16
PS020.wsdl (new).....	16
Java Changes	16
Web Merit.....	16
login.properties	16
Web PAN.....	16
ApplicationResources.properties	16

PPSWebServices Application	16
Properties Files	16
log4j.properties	16
Java Classes	17
EDBEmployeeAdapter.java	17
EDBEmployeeAdapterImpl.java	17
IOccurrenceValuePair.java	17
XmlTypeConverter.java	17
PS010Client.java	17
UserNotAuthorizedException.java	17
PPSWSEException.java	17
EmpDataService.java	17
EmployeeDataElement.java	17
ObjectFactory.java	17
PS001Client.java	18
PS001Input.java	18
PS001Output.java	18
PS001Service.java	18
BottomLine.java	18
BrowseLine.java	18
BrowseRequest.java	18
EmployeeBrowser.java	18
EmployeeData.java	18
ObjectFactory.java	19
PS003Input.java	19
PS003InputWrapper.java	19
PS003Output.java	19
PS003OutputWrapper.java	19
PS003Service.java	19
TopLine.java	19
LogoffService.java	19
ObjectFactory.java	19
PS004Input.java	19
PS004InputWrapper.java	20
PS004Output.java	20
PS004Service.java	20
MatchData.java	20
ObjectFactory.java	20
PS005Input.java	20
PS005InputWrapper.java	20
PS005Output.java	20
PS005OutputWrapper.java	20
PS005Service.java	21
UniqueMatchChecker.java	21
UniqueMatchRequest.java	21
InputWrapper.java	21
ObjectFactory.java	21
OutputWrapper.java	21
PS010Input.java	21
PS010InputWrapper.java	21
PS010Output.java	21
PS010OutputWrapper.java	21
PS010PortType.java	22
PS010Service.java	22
ReturnCodeEnum.java	22
UserTokenEnum.java	22

UserTokenService.java	22
Adapter1.java	22
GrossToNetItem.java	22
GrossToNetTranslator.java	22
GrossToNetValues.java	22
ObjectFactory.java	22
PS011Input.java	23
PS011InputWrapper.java	23
PS011OUTPUT.java	23
PS011OutputWrapper.java	23
PS011Service.java	23
Adapter1.java	23
EmployeeIDMatcher.java	23
EmployeeMatch.java	23
MatchingEmployeesListing.java	23
ObjectFactory.java	23
PS012Input.java	24
PS012InputElem.java	24
PS012Output.java	24
PS012OutputElem.java	24
PS012Service.java	24
Adapter1.java	24
DataElementTranslation.java	24
ObjectFactory.java	24
PS013Input.java	24
PS013InputElem.java	24
PS013Output.java	25
PS013OutputElem.java	25
PS013Service.java	25
RelatedDataElements.java	25
RequestedDataElement.java	25
RequestedDataElements.java	25
Translations.java	25
ObjectFactory.java	25
PS014Input.java	25
PS014InputElem.java	25
PS014InputWrapper.java	26
PS014Output.java	26
PS014OutputElem.java	26
PS014OutputWrapper.java	26
PS014Service.java	26
SystemControlRecord.java	26
SystemControlRecordRetriever.java	26
ObjectFactory.java	26
PS020Input.java	26
PS020InputWrapper.java	26
PS020Output.java	27
PS020OutputWrapper.java	27
PS020Service.java	27
RequestProfile.java	27
RequestProfileItem.java	27
ResponseProfile.java	27
ResponseProfileItem.java	27
ScreenLevelAuthorizer.java	27
AppointmentImpl.java	27
AppointmentRowOutput.java	27

BenefitsRowOutput.java.....	28
DistributionImpl.java.....	28
DistributionRowOutput.java.....	28
DuesAgencyFeeRow.java.....	28
EDBEmployee.java.....	28
FamilyBenefitsRowOutput.java.....	28
FinancialAidImpl.java.....	28
FinancialAidRowOutput.java.....	28
FutureBenefitsRowOutput.java.....	28
HourBalancesDisplay.java.....	28
IAppointment.java.....	29
IDistribution.java.....	29
IFinancialAid.java.....	29
ILeaveAccrualPeriodSet.java.....	29
InsuranceRowOutput.java.....	29
IWorkStudy.java.....	29
LaborRelationsApptRow.java.....	29
LeaveAccrualHistoryRowOutput.java.....	29
LeaveAccrualPeriodRowOutput.java.....	29
LeaveAccrualPeriodSetImpl.java.....	29
LicenseRowOutput.java.....	30
MapBackedEDBEmpImpl.java.....	30
RetirementGTNRowOutput.java.....	30
WorkStudyImpl.java.....	30
WorkStudyRowOutput.java.....	30
EDBEmployeeDAO.java.....	30
EmployeeDAOImpl.java.....	30
EmployeeNotFoundException.java.....	30
MatchingEmp.java.....	30
IdentitySystemFacade.java.....	30
IIDFacadeImpl.java.....	31
PPSWSTranslationServiceFacadeImpl.java.....	31
TokenServiceFacade.java.....	31
TokenServiceFacadeImpl.java.....	31
TranslationServiceFacade.java.....	31
DataElementMapEmitter.java.....	31
EDBDataElement.java.....	31
EDBDataElementMapper.java.....	31
ModifyServiceClassWsdUrl.java.....	31
Other Files.....	32
README.....	32
PS001-jaxb-bindings.xml.....	32
PS001-jaxws-bindings.xml.....	32
PS003-jaxb-bindings.xml.....	32
PS003-jaxws-bindings.xml.....	32
PS004-jaxb-bindings.xml.....	32
PS004-jaxws-bindings.xml.....	32
PS005-jaxb-bindings.xml.....	32
PS005-jaxws-bindings.xml.....	32
PS010-jaxb-bindings.xml.....	32
PS010-jaxws-bindings.xml.....	32
PS011-jaxb-bindings.xml.....	33
PS011-jaxws-bindings.xml.....	33
PS012.wsdl.....	33
PS012-jaxb-bindings.xml.....	33
PS012-jaxws-bindings.xml.....	33

PS013.wsdl	33
PS013-jaxb-bindings.xml	33
PS013-jaxws-bindings.xml	33
PS014-jaxb-bindings.xml	33
PS014-jaxws-bindings.xml	33
PS020-jaxb-bindings.xml	33
PS020-jaxws-bindings.xml	34
junit-3.8.1.jar	34
webservices-api.jar	34
webservices-extra.jar	34
webservices-extra-api.jar	34
webservices-rt.jar	34
webservices-tools.jar	34
wstx-services.war	34
PPSWeb Java Application	35
Properties Files	35
ApplicationResources.properties	35
PPSWeb.properties	35
log4j.properties	35
Java Classes	35
AccessEDBAction.java	35
ChangePasswordAction.java	35
LogoffAction.java	36
LogonAction.java	36
MainMenuAction.java	36
PPSWebBusinessDelegate.java	36
StoredProcedureCalls.java	36
BaseDAO.java	36
CallspDAO.java	36
JDBCDataAccessor.java	36
DataAccessException.java	36
AuthenticationException.java	36
ChangePasswordException.java	37
EDBAccessException.java	37
EDBBaseException.java	37
EDBBusinessException.java	37
InvalidSessionException.java	37
AuthenticationFacade.java	37
ChangeUserFacade.java	37
EDBAvailibility.java	37
LogUserOffFacade.java	37
PPSWebAppProperties.java	38
ApplicationPropertiesLoader.java	38
MiscUtil.java	38
PPSWebValidator.java	38
ScreenLevelSecurity.java	38
SessionCheck.java	38
SessionFilter.java	38
JSPs	38
changepassword.jsp	38
checksession.jsp	38
login.jsp	39
logoff.jsp	39
mainmenu.jsp	39
sessioninvalid.jsp	39
staticJavascript.jsp	39

timeout.jsp	39
Javascript	39
ppsweb.js	39
Image Files	39
footerline.gif	39
login_background.gif	39
uc.gif	39
Other Files	39
pps.css	39
ibm-web-bnd.xmi	40
ibm-web-ext.xmi	40
struts-config.xml	40
validation.xml	40
validator-rules.xml	40
web.xml	40
antlr-2.7.2.jar	40
commons-beanutils-1.7.0.jar	40
commons-chain-1.1.jar	40
commons-digester-1.8.jar	40
commons-discovery-0.2.jar	40
commons-io-1.1.jar	41
commons-logging-1.0.4.jar	41
commons-validator-1.3.1.jar	41
jstl-1.0.2.jar	41
log4j-1.2.15.jar	41
oro-2.0.8.jar	41
standard-1.0.2.jar	41
struts-core-1.3.8.jar	41
struts-el-1.3.8.jar	41
struts-extras-1.3.8.jar	41
struts-faces-1.3.8.jar	41
struts-scripting-1.3.8.jar	41
struts-taglib-1.3.8.jar	42
EDBInquiry Java Application	43
Detail Screen Information	43
Properties Files	43
ApplicationResources.properties	43
EDBInquiry.properties	43
log4j.properties	44
Java Classes	44
AddressAction.java	44
AppointmentsAction.java	44
BenefitsAction.java	44
BrowseAction.java	44
CitizenshipAction.java	44
EDBCoreAction.java	45
EmploymentAction.java	45
FamilyBenefitsAction.java	45
FinancialAidAction.java	45
FooterAction.java	45
FutureBenefitsAction.java	45
HeaderAction.java	45
HourBalancesAction.java	45
LaborRelationsAction.java	46
LeaveAccrualAction.java	46
LicensesAction.java	46

LogoffAction.java.....	46
MainMenuAction.java.....	46
PayDispositionAction.java.....	46
PersonalAction.java.....	46
RetirementAction.java.....	47
StartEDBAction.java.....	47
TaxesAction.java.....	47
WorkstudyAction.java.....	47
EDDBusinessDelegate.java.....	47
EDDBusinessDelegate.java.....	47
EDBInquiryConstants.java.....	47
StoredProcedureCalls.java.....	48
BaseDAO.java.....	48
CallspDAO.java.....	48
JDBCDataAccessor.java.....	48
BrowseOutput.java.....	48
BrowseParams.java.....	48
DataAccessException.java.....	48
EDDBaseException.java.....	48
EDDBusinessException.java.....	48
EmployeeMatchException.java.....	48
BrowseFacade.java.....	49
EmployeeInfoFacade.java.....	49
GlobalValuesFacade.java.....	49
UniqueMatchFacade.java.....	49
BrowseForm.java.....	49
HeaderForm.java.....	49
EDBAppProperties.java.....	49
ApplicationPropertiesLoader.java.....	49
EDBValidator.java.....	49
Misc.java.....	50
ScreenLevelSecurity.java.....	50
SessionFilter.java.....	50
Translator.java.....	50
JSPs.....	50
address_ppwwiadr.jsp.....	50
appointments_ppwwiapp.jsp.....	50
benefits_ppwwiben.jsp.....	50
blank.jsp.....	50
browse_ppwwibrs.jsp.....	50
citizenship_ppwwicit.jsp.....	50
detail_print.jsp.....	51
detail_screen.jsp.....	51
edb.jsp.....	51
employeeselection.jsp.....	51
employment_ppwwiemp.jsp.....	51
error.jsp.....	51
familybenefits_ppwwifam.jsp.....	51
financialaid_ppwwifin.jsp.....	51
footer.jsp.....	51
futurebenefits_ppwwifut.jsp.....	51
header.jsp.....	51
hourbalances_ppwwipay.jsp.....	51
laborrelations_ppwwilab.jsp.....	52
leaveaccrual_ppwwilva.jsp.....	52
licenses_ppwwilic.jsp.....	52

navigation.jsp	52
notauthorized.jsp	52
paydisposition_ppwwidsp.jsp	52
personal_ppwwiper.jsp	52
retirement_ppwwiret.jsp	52
taxes_ppwwitax.jsp	52
workstudy_ppwwiwrk.jsp	52
Javascript	53
EDBInquiry.js	53
Image Files	53
address.gif	53
appointments.gif	53
benefits.gif	53
browse.gif	53
citizenship.gif	53
employment.gif	53
financialaid.gif	53
footerline.gif	53
laborrelations.gif	53
leaveaccrual.gif	53
licenses.gif	53
login_background.gif	54
paydisposition.gif	54
payhours.gif	54
PayPerSys3A.jpg	54
personal.gif	54
printerFriendly.gif	54
retirement.gif	54
taxes.gif	54
uc.gif	54
workstudy.gif	54
Other Files	54
edbinquiry.css	54
ibm-web-bnd.xmi	54
ibm-web-ext.xmi	54
struts-config.xml	55
validation.xml	55
validator-rules.xml	55
web.xml	55
antlr-2.7.2.jar	55
commons-beanutils-1.7.0.jar	55
commons-chain-1.1.jar	55
commons-digester-1.8.jar	55
commons-discovery-0.2.jar	55
commons-io-1.1.jar	55
commons-logging-1.0.4.jar	55
commons-validator-1.3.1.jar	56
jstl-1.0.2.jar	56
log4j-1.2.15.jar	56
oro-2.0.8.jar	56
standard-1.0.2.jar	56
struts-core-1.3.8.jar	56
struts-el-1.3.8.jar	56
struts-extras-1.3.8.jar	56
struts-faces-1.3.8.jar	56
struts-scripting-1.3.8.jar	56

struts-taglib-1.3.8.jar	56
Forms	57
UPAY930 (new)	57
Control Table Updates	57
System Messages Table	57
Obsolete Objects	57
Appendix A.....	58
Technical Note on web services with unbounded arrays	58
Appendix B.....	59
Steps for Customizing the Applications.....	59
Appendix C.....	60
How to Add Data Elements to an EDB Inquiry Screen	60
Appendix D.....	62
How to Create Alternate Versions of an EDB Inquiry Screen.....	62
Appendix E: UPAY930 - UCSSIMNT Program Run Specification	63

Introduction

Service Request 82355

Service request 82355 asks that the Web-based EDB Inquiry Application be re-written using Java, replacing the existing application written in the un-supported Net.Data language. In addition to web EDB inquiry, the login, change password and main menu processes have also been re-written in java.

In addition, the Service Request asks that the social security number be removed from all EDB inquiry detail screens. The functionality that allows users to search for an employee by Social Security Number remains unchanged.

Service Request 82355 provides the following background information:

The Web EDB Inquiry application was developed in an effort to provide a more user-friendly alternative to the CICS screens accessed by campus departmental users (see R1420, April, 2002). The application was initially written in the Net.Data programming language.

A plan was already underway to re-write the application, because Net.Data is no longer a supported technology. The timeline of the re-write has now been accelerated due to security concerns related to the technology.

Overview of PPS Modifications

PPS CICS Regions and New Batch Process

Historically, campus PPS CICS regions are brought down at the end of the business day to ensure that no database updates take place during the batch processing window. Now that the web login, main menu and EDB inquiry processes will depend on CICS web services, the regions must be kept running whenever access to these web applications is required. To prevent any CICS screen activity during the batch processing window, a new batch program, UCSSIMNT is being released which can selectively disable and enable PPS CICS subsystems while leaving the web services available at all times. New form UPAY930 is being released to describe the run specification for this program.

CICS Web Services Programs

New CICS web service programs have been written to handle the SOAP messages from the web clients.

Web Artifacts

The web service wsdl's have been generated by RADz as part of the development process. Other web service related items have been written to support the PPS services.

Web Merit Changes

Minor changes have been made to the Web Merit properties file to allow for returning to the java main menu.

Web PAN Changes

Minor changes have been made to the Web PAN properties to allow for returning to the java main menu and to EDB inquiry.

PPSWeb Application (new)

The current login, main menu and change password features, currently written in Net.Data have been rewritten utilizing Java based technologies. These Java based technologies include Struts, JSP and CICS Web Services. This is being deployed as a new web application called PPSWeb.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

EDBInquiry Application (new)

The current Employee Database Inquiry web application, written in net.data, has been rewritten utilizing Java based technologies. These Java based technologies include Struts, JSP and CICS Web Services. This is being deployed as a new web application called EDBInquiry. Removing the social security number from the detail screens will be done as part of this rewrite.

PPSWebServices Application (new)

The code to invoke CICS web services is contained in a shared library called PPSWebServices. This code is used by both PPSWeb and EDBInquiry.

Differences Between Old and New Application

The intent of this project is to provide the same functionality, look and feel as the existing net.data web applications, but using current technology. In order to provide the best possible performance using the new technology, a few minor discrepancies may be noticed. These are documented here.

Data Refresh

In the existing system, if the EDB is updated via CICS, that change is reflected when the user navigates to a different screen for the same employee within EDB inquiry and then back again.

In the new system, all of the data for the currently selected employee is cached until the user navigates to a different employee or back to the main menu. To see database changes reflected, it is necessary to either switch to another employee and back again, or to navigate to the main menu.

Employment Status Screen

In the existing system, there is a bug on the Employment Status Screen. When there is no data for the SLCG Effective Date, it is being displayed as "01/01/0001". This has been corrected in the java version and the field will display as blanks.

Employee Browse Screen

In the existing system, when the user enters data in the name field which does not match a name on the EDB, the browse screen is displayed starting with the name prior to the entered text. The new EDB inquiry application displays the names starting with the name after the entered text, which is consistent with the CICS WHO screen.

Links on "Print Friendly" Screens

In the existing net.data system, links on the "print friendly" screens are enabled, but clicking on the links causes an error. It is inappropriate to have enabled links on a printable screen, so these links will be removed from the print friendly screens in the new version.

Messages

It is considered good practice to avoid giving "too much information" to potential hackers in messages related to system security. For example, if a user enters a valid user ID and incorrect password, it is preferable to have a generic message saying "not authorized" rather than "invalid password". The rewritten application follows this recommendation. For example, message WA005 which currently reads "Invalid password" has been changed to "Login or password incorrect".

Security and Session Control in the Java PPS Applications

Certificates

Each web server on which the new applications are installed must have a client certificate authorizing it to access the web services in the CICS region. Each time a web service is invoked, program PPWEBSEC will validate the certificate.

Tokens

In release 1862, program UC0\$WBST was introduced to generate "tokens". In CICS, a token can be used to track the state of an operation which is not related to a specific terminal. A token is a randomly generated number used to identify an active session. Session-related data (eg. user ID) can be saved in main storage with the token as the key.

When the login web service, PS010, is invoked from the login screen, a token is generated and stored in CICS along with the user ID. The token is returned to the web application, where it is stored in a secure browser cookie. After that, calls to any PPS web service must supply a valid token as input, and program PPWEBSEC will cause the associated CICS task to run under the user ID associated with that token.

Each time PPWEBSEC validates a token, it will update the timestamp associated with the token. Program PPDELTOK will run each hour to delete any tokens which haven't been accessed for an hour.

Session Timeout

As with the old application, the rewritten applications will support session timeout based on an external property setting. If the timeout interval is set sufficiently low, this can prevent an unauthorized user from sitting down at an unoccupied workstation and taking advantage of the previous user's session.

CICS Web Services

PS001 (new)

This service provides all EDB data for a single employee. It checks ARSM to determine whether the logged in user is authorized to view data for that employee.

See Cobol program PS001 for more detail.

PS003 (new)

This service provides EDB data for the browse screen.

See Cobol program PS003 for more detail.

PS004 (new)

This service destroys a user token. It also calls PPDELTS to delete all possible temp storage queues associated with the token.

See Cobol program PS004 for more detail.

PS005 (new)

This service returns an employee ID for a name or SSN, but only if there's a unique match.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

See Cobol program PS005 for more detail.

PS010 (new)

This service authenticates a user and establishes a token in CICS which may be used for session control in other services. If a new password is sent, the service first changes the password, and then does the authentication and returns the token.

See Cobol program PS010 for more detail.

PS011 (new)

This service returns GTN data for GTNs which meet one of specified criteria.

See Cobol program PS011 for more detail.

PS014 (new)

This service returns SCR data.

See Cobol program PS014 for more detail.

PS016 (new)

This service returns the user ID associated with the token passed.

See Cobol program PS016 for more detail.

PS020 (new)

This service checks mainframe security to determine which screens the logged in user is authorized to use. The requestor sends an array of screen function codes and the service returns an array of codes indicating whether or not the user is authorized for each and if so, the authorized version.

See Cobol program PS020 for more detail.

Cobol Programs

New and Modified Programs

PPDELTK (new)

PPDELTK is a new program used to delete any token more than one hour old. It runs under transaction T010 which is started by the new PLT program PPWPLT. PPDELTK starts transaction CWBT which deletes any token which is more than one hour old. It then schedules itself to be restarted in one hour.

PPFCTSPR (new)

This stored procedure returns the foreign country name from the foreign country (FCT) table.

Input

- Foreign country code

Output

- Foreign country name

- Return code - success, failure

PPHMESPR (new)

This stored procedure returns the home department name from the home department table.

Input

- Home Department Number

Output

- Home Department Name
- Home Department Name - Abbreviated
- Return code - success, failure

PPMSGSPR (new)

This stored procedure returns message text from the system messages table.

Input

- Message number

Output

- Message text
- Batch Severity
- Online severity
- Return code - success, failure

PPSPBSPR (new)

This stored procedure returns the bank name and transit routing number from the bank table.

Input

- Bank Number

Output

- Bank Name
- Transit routing number
- Return code - success, failure

PPWPLT (new)

PPWPLT is a new program to be run at CICS startup by including it in the PLT. It starts transaction T010 which causes program PPDELTK to be run.

PPWWUUSE (new)

Description

This stored procedure updates data in the session table, UC0SES. If authorization text is included, then it will be used to update the record. If no authorization text is included, then only the timestamp field will be updated. It is called from the authentication process in the PPSWeb application. Web PAN uses the UC0SES data to determine authorization for each function.

Input

- Session ID
- Authorization text

Output

- Return code - success, failure

PS001 (new)

PS001 is the driver program for the EDB Data element fetch Web Service. The shell for this program was initially generated by RADz from PS001.wSDL (see Appendix A). Then the shell program was modified for the web service.

Input

- User Token
- Employee ID

Output

- Return code - employee found, employee not found, ARSM permission denied
- Array of data element numbers and values for all non-initialized elements for the requested employee

Processing is as follows:

- Call UCPPPEDB to determine if signed on user has ARSM authorization to view the selected employee data
- Call PPEDBFET with non-initial-request for all EDB data elements that have a non-initial value
- Return to requester
 - Return code
 - 0000 employee found
 - 0100 employee not found
 - 0200 ARSM permission denied
 - Array of data elements
 - ps001XreturnXdataXelexno PIC X(4).
 - ps001XreturnXoccurXkey PIC X(30).
 - ps001XreturnXvalue PIC X(30).
 - ps001XreturnXdataXtype PIC X(1).
 - ps001XreturnXdeXnumXsw PIC X(1).

PS003 (new)

PS003 is the driver program for the PS003 Web Service, which returns EDB data for the browse screen.

Input

- User Token
- Sequence - employee ID, name or home department
- Action - next or previous, new browse, start at name
- Return count (1 to 99)
- Top line
- Bottom line

Output

- Return code - success, failure
- Array of data for up to 99 employees.

The logic is as follows:

- Determine type of request
 - Sequence
 - Action
 - Number of employees requested
- Open one of six cursors depending on type of request
 - order by **hme_abrv_dept_name, employee_id**
 - order by **hme_abrv_dept_name descending, employee_id descending**
 - order by **emp_name, employee_id**
 - order by **emp_name descending, employee_id descending**
 - order by **employee_id**
 - order by **employee_id descending**
- Fetch from cursor up to number of employees requested saving returned row in a work array
- Sort the work array to return entries in the correct order when the requested action was browse backwards
- Return to requester
 - Return code
 - 0000 success
 - 0100 none returned
 - Array of data
 - ps003XreturnXempXid PIC X(9).
 - ps003XreturnXempXname PIC X(26).
 - ps003XreturnXhmeXdepXab PIC X(15).

PS004 (new)

PS004 is the main processing program for the PS004 web service, which deletes resources associated with a token.

Input

- User Token

Output

- Return code - success, failure

Processing is as follows:

- Link to program PPDELTS to delete all temporary storage queues associated with the token passed with the web service request.
- Perform DESTROY-TOKEN which calls program DFH\$WBST with a request to delete the token resources.
- A return code is returned with the web service response.

PS004D (new)

PS004D is the RAD-generated Converter Driver program which is invoked by the web service. It parses the XML passed from the web application and then calls PS004 to process the data. On returning from PS004, the request output is translated back to XML for return to the web application.

PS005 (new)

PS005 is the main processing program for the PS005 web service, which returns an employee ID for a name or SSN passed.

Input

- User Token
- SSN
- Name

Output

- Return code with the following values:
 - unique match found
 - non-unique match found
 - no match found
 - multiple keys passed
 - error
- Employee ID if a unique match was found.

Processing is as follows:

- If both a name and SSN were passed, return an error.
- If a name was passed, attempt to find a match on the PER table. The "LIKE" predicate is used to match partial names.
 - If a unique match found, pass back the employee ID and set the return code to "unique match found".
 - If multiple matches are found, set the return code to "non-unique match found".
 - If no matches were found, set the return code to "no match found".
- If an SSN was passed, attempt to find a match on the PAY table.
 - If a unique match found, pass back the employee ID and set the return code to "unique match found".
 - If multiple matches are found, set the return code to "non-unique match found".
 - If no matches were found, set the return code to "no match found".

PS005D (new)

PS005D is the RAD-generated Converter Driver program which is invoked by the web service. It parses the XML passed from the web application and then calls PS005 to process the data. On returning from PS005, the request output is translated back to XML for return to the web application.

PS010 (new)

PS010 is the main processing program for the login web service, PS010. This program verifies a user ID and password, or process a change password request.

Input

- User ID
- Password
- New password

Output

- Return code - success, failure, password expired, etc.
- Token

Processing is as follows:

- If the new password field contains data, then a change password request is assumed. An EXEC CICS CHANGE PASSWORD command is issued using the user ID, password and new password fields passed in the web service request.
- If the new password field is spaces, then it's a request to verify the password. An EXEC CICS VERIFY PASSWORD command is issued.
- If the password is verified, GENERATE-SESSION-TOKEN is performed, which calls program DFH\$WBST with a request to create a new token.
- A return code and the generated token value are returned with the web service response.

PS010D (new)

PS010D is the RAD-generated Converter Driver program which is invoked by the web service. It parses the XML passed from the web application and then calls PS010 to process the data. On returning from PS010, the request output is translated back to XML for return to the web application.

PS011 (new)

PS011 is the main processing program for web service PS011. This program returns an array of GTN data which meets selected criteria. The requestor can indicate a request code of "L" for Labor or "R" for Retirement.

- If the code is "L", GTNs must meet the following criteria to be included in the result set:
GTN_DISPLAY_IDED = 'Y' and (either GTN_GROUP = 'U' or GTN_SET_INDICATOR = '3', 'C', or 'P')
- If the code is "R", GTNs must have GTN_GROUP = 'R' to be included in the result set:
GTN_DISPLAY_IDED = 'Y' and (either GTN_GROUP = 'U' or GTN_SET_INDICATOR = '3', 'C', or 'P')

PS014 (new)

PS014 is the main processing program for web service, PS014. This program returns SCR data. The program reads data from the SCR table and returns it to the requestor.

PS014D (new)

PS014D is the RAD-generated Converter Driver program which is invoked by the web service. It parses the XML passed from the web application and then calls PS014 to process the data. On returning from PS014, the request output is translated back to XML for return to the web application.

PS016 (new)

PS016 is the main processing program for web service, PS016. This program returns the user ID for a token. The program calls UC0\$WBST with the token, which returns all token-related data. The user ID is passed back to the requestor.

Input

- Previously received token

Output

- User ID
- Return Code - success or failure

PS016D (new)

PS016D is the RAD-generated Converter Driver program which is invoked by the web service. It parses the XML passed from the web application and then calls PS016 to process the data. On returning from PS016, the request output is translated back to XML for return to the web application.

PS020 (new)

PS020 is the driver program for the PS020 Web Service, which checks function-level RACF security. The shell for this program was initially generated by RADz from PS020.wsd1 (see Appendix A). Then the shell program was modified for the web service.

Input

- Previously received token
- Maximum number of suffixed screen names to check
- Unbounded array:
 - Security prefix
 - Profile name of screen requested (un-suffixed version name)

Output

- Overall return code - success, failure
- Unbounded array:
 - Security prefix
 - Profile name of screen requested (un-suffixed version name)
 - Version suffix of *first authorized* screen version found – 1 character
 - Return code - authorized, not authorized or another error.

Processing is as follows:

Each requested screen function code is checked as well as suffixed screen version up to the “Maximum number of suffixed screen names to check”; suffixes are checked in this order space, 1, 2,..., 9, a, b, c,..., y. (E.g. if max is 1 only the base version is checked (the base version has space as a suffix). If the max is 2 the base version is checked (suffix is space) and the version with suffix 1 is checked. A parameter in EDBInquiry.properties contains the maximum number of versions (1 in base version of Web EBD Inquiry).

- The web service request contains an array of \$PPSFUNC RACF security prefixes and function codes. For each, PS020 calls program UCFNAUTH to determine whether the logged in user is authorized to access that function.
- An array of return codes is returned with the web service response.

UCFNAUTH

UCFNAUTH is used to check RACF security for a CICS screen function. This validates a user's access to the screen function requested. It has been modified as follows:

- Currently, UCFNAUTH uses the CICS region ID as the security prefix for the call to RACF. It has been changed so that for UCPIAUTH-FUNCTION-TYPE="W", it uses the security prefix passed from the caller.

UCROUTER

UCROUTER is the main driver for all PPS CICS processes. It has been modified as follows:

- For every iteration, read the "UC" row of the SSI table to see if subsystem access has been disabled at the global level. If it has, set a switch so that all subsystems will be disabled for CICS access.
- Whenever checking SSI for subsystem availability, also check the global switch and don't allow access if it is set.

UCSSIMNT (new)

UCSSIMNT is a new batch program to set subsystem availability. Spec card input indicates which subsystems should be enabled/disabled. After the initial spec card defining the run, there may be an additional card specifying subsystems to be enabled or disabled.

The initial spec card gives the program ID plus the "overall action" code for the current run of the program. Values for the overall action are:

- D - Disable all Subsystems
- E - Enable All Subsystems
- S - Selectively enable/disable subsystems

Additional spec cards may be included which specify subsystems which should be enabled or disabled selectively.

UCWMMNU

UCWMMNU is the main menu program for the PPS CICS applications. It has been modified as follows:

- Check the global subsystem availability switch (set by UCROUTER). If subsystem access has been disabled at the global level, flag all subsystems as unavailable.

Copy Members

CPWSTOKN

Changed definition of USER-TOKEN-BINARY from BINARY to COMP-5 to avoid truncation when computing using the token value.

PS001 (new)

PS001 defines the input and output structure for the PS001 web service (EDB Data element fetch). It is used only to generate the wsdl and is not copied into any Cobol program. Any changes to the structure require that the wsdl be re-generated. (see Appendix A)

PS001I01 (new)

PS001I01 is the input data structure for web service PS001 (EDB Data element fetch Web Service). It was generated by RADz from PS001.wsdl at the same time as the PS001 shell program was generated. (see Appendix A)

PS001O01 (new)

PS001O01 is the output data structure for web service PS001 (EDB Data element fetch Web Service). It was generated by RADz from PS001.wsdl at the same time as the PS001 shell program was generated. (see Appendix A)

PS003 (new)

PS003 will define the input and output structure for the PS003 web service (EDB data for browse). It is used only to generate the wsdl and is not copied into any Cobol program. Any changes to the structure require that the wsdl be re-generated. (see Appendix A)

PS003I01 (new)

PS003I01 is the input data structure for web service PS003 (EDB data for browse). It was generated by RADz from PS003.wsdl at the same time as the PS003 shell program was generated. (see Appendix A)

PS003O01 (new)

PS003O01 is the output data structure for web service PS003 (EDB data for browse). It was generated by RADz from PS003.wsdl at the same time as the PS003 shell program was generated. (see Appendix A)

PS004 (new)

PS004 is the data structure for the PS004 web service (delete token resources). It is used by RADz to generate the WSDL for the web service. Therefore, any changes to the data structure will require that the WSDL be re-generated. It will also be used to pass parsed XML between the web service and program PS004.

PS005 (new)

PS005 is the data structure for the PS005 web service (return employee ID for name or SSN passed). It is used by RADz to generate the WSDL for the web service. Therefore, any changes to the data structure will require that the WSDL be re-generated. It will also be used to pass parsed XML between the web service and program PS005.

PS010 (new)

PS010 is the data structure for the PS010 web service (login and change password). It is used by RADz to generate the WSDL for the web service. Therefore, any changes to the data structure will require that the WSDL be re-generated. It will also be used to pass parsed XML between the web service and program PS010.

PS011 (new)

PS011 will define the input and output structure for the PS011 web service (GTN Selection). It is used only to generate the wsdl and is not copied into any Cobol program. Any changes to the structure require that the wsdl be re-generated. (see Appendix A)

PS011I01 (new)

PS011I01 is the input data structure for web service PS011 (GTN Selection). It was generated by RADz from PS011.wsdl at the same time as the PS011 shell program was generated. (see Appendix A)

PS011O01 (new)

PS011O01 is the output data structure for web service PS011 (GTN Selection). It was generated by RADz from PS011.wsdl at the same time as the PS011 shell program was generated. (see Appendix A)

PS014 (new)

PS014 is the data structure for the PS014 web service (SCR selection). It is used by RADz to generate the WSDL for the web service. Therefore, any changes to the data structure will require that the WSDL be re-generated. It will also be used to pass parsed XML between the web service and program PS014.

PS016 (new)

PS016 is the data structure for the PS016 web service (Return user ID from token). It is used by RADz to generate the WSDL for the web service. Therefore, any changes to the data structure will require that the WSDL be re-generated. It will also be used to pass parsed XML between the web service and program PS016.

PS020 (new)

PS020 will define the input and output structure for the PS020 web service (function-level security). It is used only to generate the wsdl and is not copied into any Cobol program. Any changes to the structure require that the wsdl be re-generated. (see Appendix A)

PS020I01 (new)

PS020I01 is the input data structure for web service PS020 (function-level security). It was generated by RADz from PS020.wsdl at the same time as the PS020 shell program was generated. (see Appendix A)

PS020O01 (new)

PS020O01 is the output data structure for web service PS020 (function-level security). It was generated by RADz from PS020.wsdl at the same time as the PS020 shell program was generated. (see Appendix A)

UCCOMMON

UCCOMMON is the common external work area for all UCROUTER subsystems. A new field has been added to indicate global subsystem availability.

UCPIAUTH

UCPIAUTH defines an EXTERNAL area for UCNFAUTH. Two fields have been added to contain the web security prefix and profile names for web services.

DDL Members

SPFCTSPR (new)

SPFCTSPR is the DDL containing create statements for new stored procedure PPFCTSPR.

SPHMESPR (new)

SPHMESPR is the DDL containing create statements for new stored procedure PPHMESPR.

SPMSGSPR (new)

SPMSGSPR is the DDL containing create statements for new stored procedure PPMMSGSPR.

SPSPBSPR (new)

SPSPBSPR is the DDL containing create statements for new stored procedure PPSPBSPR.

SPUSE00C (new)

SPUSE00C is the DDL containing create statements for new stored procedure PPWWUUSE.

Bind Members

New and Modified Binds

PPFCTSPR (new)

PPFCTSPR contains the package bind statements for stored procedure PPFCTSPR.

PPHMESPR (new)

PPHMESPR contains the package bind statements for stored procedure PPHMESPR.

PPMSGSPR (new)

PPMSGSPR contains the package bind statements for stored procedure PPMMSGSPR.

PPSPBSPR (new)

PPSPBSPR contains the package bind statements for stored procedure PPSPBSPR.

PPWWUUSE (new)

PPWWUUSE contains the package bind statements for stored procedure PPWWUUSE.

PS001 (new)

PS001 contains the package bind statements for web service program PS001.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS003 (new)

PS003 contains the package bind statements for web service program PS003.

PS005 (new)

PS005 contains the package bind statements for web service program PS005.

PS011 (new)

PS011 contains the package bind statements for web service program PS011.

PS014 (new)

PS014 contains the package bind statements for web service program PS014.

UCSSIMNT (new)

UCSSIMNT contains the plan bind statements for batch program UCSSIMNT.

Web Service Definitions (wsdl)

PS001.wsdl (new)

PS001.wsdl defines the PS001 web service interface for calling web-service PS001 to get the EDB data elements. PS001.wsdl was generated in RADz.

PS003.wsdl (new)

PS003.wsdl defines the PS003 web service interface for calling web-service PS003 to get the EDB employee data based on employee ID, employee name and home department. PS003.wsdl was generated in RADz.

PS004.wsdl (new)

PS004.wsdl defines the PS004 web service interface for calling web-service PS004 to log off from the PPS mainframe system. PS004.wsdl was generated in RADz.

PS005.wsdl (new)

PS005.wsdl defines the PS005 web service interface for calling web-service PS004 to get employee ID based on employee name or SSN.. PS005.wsdl was generated in RADz.

PS010.wsdl (new)

PS010.wsdl defines the PS010 web service interface for logging in or change password services. PS010.wsdl was generated in RADz.

PS011.wsdl (new)

PS011.wsdl defines the PS011 web service interface. PS011.wsdl was generated in RADz.

PS014.wsdl (new)

PS014.wsdl defines the PS014 web service interface for calling web-service to obtain SCR data. PS014.wsdl was generated in RADz.

PS016.wsdl (new)

PS016.wsdl defines the PS016 web service interface. PS016.wsdl was generated in RADz.

PS020.wsdl (new)

PS020.wsdl defines the PS020 web service interface for calling web-service to return RACF permissions. PS020.wsdl was generated in RADz.

Java Changes

Web Merit

login.properties

appl.menuurl & appl.logouturl must be modified with the appropriate values on the new server.

- appl.menuurl=***servername***/PPSWeb/mainmenu.jsp
- appl.logouturl=***servername***/PPSWeb/logoff.do

Web PAN

ApplicationResources.properties

- net.data.base = ***servername***/PPSWeb
- net.data.edb.link = accessEDB.do?headeraction=locateid&browseLink=true
- net.data.mainmenu.link = mainmenu.jsp
- net.data.logout.link = logoff.do

PPSWebServices Application

PPSWebServices is a new java shared library which contains the code to invoke the CICS web services. It is used by both the PPSWeb and EDBInquiry applications.

This library includes interfaces for web services which are not being released at this time.

Properties Files

log4j.properties

log4j.properties contains the configurable properties for Apache. **log4j** is a Java-based logging utility. The log file location is defined by property log4j.appender.LOGFILE.File. The web application server administrator can control the location of the log file by defining a JVM custom property named APP_LOG_ROOT with the value of the directory path.

Java Classes

EDBEmployeeAdapter.java

Package: edu.ucop.ppsws.adapter

Provides an adapter layer between the raw PS010 return data and an EDBEmployee type.

EDBEmployeeAdapterImpl.java

Package: edu.ucop.ppsws.adapter

Provides an adapter layer between the web service output and logical organization of fields in the Employee object.

IOccurrenceValuePair.java

Package: edu.ucop.ppsws.adapter

Maps an occurrence key to its corresponding value.

XmlTypeConverter.java

Package: edu.ucop.ppsws.adapter

Utility class converting a Y/N string into a boolean.

PS010Client.java

Package: edu.ucop.ppsws.client

Convenience class wrapping a user authentication service request.

UserNotAuthorizedException.java

Package: edu.ucop.ppsws.client

Raised when a user is not authorized.

PPSWSException.java

Package: edu.ucop.ppsws.client.exception

General exception type for exceptions encountered by PPS web service clients.

EmpDataService.java

Package: edu.ucop.ppsws.client.ps001

Employee data service client for PS001.

EmployeeDataElement.java

Package: edu.ucop.ppsws.client.ps001

A non-initial data element number and value for the given EID.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps001

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps001 package.

PS001Client.java

Package: edu.ucop.ppsws.client.ps001

Convenience class wrapping an employee load service call.

PS001Input.java

Package: edu.ucop.ppsws.client.ps001

Wrapper around input values to service.

PS001Output.java

Package: edu.ucop.ppsws.client.ps001

Wrapper around the services output data.

PS001Service.java

Package: edu.ucop.ppsws.client.ps001

This class was generated by the JAX-WS RI.

BottomLine.java

Package: edu.ucop.ppsws.client.ps003

Object describing the current bottom line.

BrowseLine.java

Package: edu.ucop.ppsws.client.ps003

A browse line, comprised of the fields describing an employee.

BrowseRequest.java

Package: edu.ucop.ppsws.client.ps003

Describes the browse request being made by this service request.

EmployeeBrowser.java

Package: edu.ucop.ppsws.client.ps003

The employee browser service.

EmployeeData.java

Package: edu.ucop.ppsws.client.ps003

Java class for Dfhcommarea_ps003__output complex type.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps003

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps003 package.

PS003Input.java

Package: edu.ucop.ppsws.client.ps003

Outer wrapper for input values into service request.

PS003InputWrapper.java

Package: edu.ucop.ppsws.client.ps003

PS003Output.java

Package: edu.ucop.ppsws.client.ps003

The payload returned by the service.

PS003OutputWrapper.java

Package: edu.ucop.ppsws.client.ps003

PS003Service.java

Package: edu.ucop.ppsws.client.ps003

This class was generated by the JAX-WS RI.

TopLine.java

Package: edu.ucop.ppsws.client.ps003

Object describing the current top line.

LogoffService.java

Package: edu.ucop.ppsws.client.ps004

Service that logs a user off the system, invalidating the current authorization token and any associated mainframe session data.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps004

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps004 package.

PS004Input.java

Package: edu.ucop.ppsws.client.ps004

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

The request input object, which contains fields for all required values.

PS004InputWrapper.java

Package: edu.ucop.ppsws.client.ps004

Input wrapper.

PS004Output.java

Package: edu.ucop.ppsws.client.ps004

The response output wrapper.

PS004Service.java

Package: edu.ucop.ppsws.client.ps004

This class was generated by the JAX-WS RI.

MatchData.java

Package: edu.ucop.ppsws.client.ps005

Encapsulates the match response: either an employee ID or a failure error code.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps005

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps005 package.

PS005Input.java

Package: edu.ucop.ppsws.client.ps005

Input wrapper class.

PS005InputWrapper.java

Package: edu.ucop.ppsws.client.ps005

PS005Output.java

Package: edu.ucop.ppsws.client.ps005

The response data wrapper.

PS005OutputWrapper.java

Package: edu.ucop.ppsws.client.ps005

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS005Service.java

Package: edu.ucop.ppsws.client.ps005

This class was generated by the JAX-WS RI.

UniqueMatchChecker.java

Package: edu.ucop.ppsws.client.ps005

Service to check for the existence in the EDB of an employee matching an SSN or employee name.

UniqueMatchRequest.java

Package: edu.ucop.ppsws.client.ps005

Defines the attributes of a unique match request to the service.

InputWrapper.java

Package: edu.ucop.ppsws.client.ps010

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps010

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps010 package.

OutputWrapper.java

Package: edu.ucop.ppsws.client.ps010

PS010Input.java

Package: edu.ucop.ppsws.client.ps010

Data transfer object holding the request data fields.

PS010InputWrapper.java

Package: edu.ucop.ppsws.client.ps010

Wrapper class around the input data transfer object that actually hold the request data.

PS010Output.java

Package: edu.ucop.ppsws.client.ps010

The data transfer object holding the server's response data.

PS010OutputWrapper.java

Package: edu.ucop.ppsws.client.ps010

Wrapper class around the output data transfer object that actually hold the request data

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS010PortType.java

Package: edu.ucop.ppsws.client.ps010

This class was generated by the JAX-WS RI.

PS010Service.java

Package: edu.ucop.ppsws.client.ps010

This class was generated by the JAX-WS RI.

ReturnCodeEnum.java

Package: edu.ucop.ppsws.client.ps010

Java class for returnCodeEnum complex type.

UserTokenEnum.java

Package: edu.ucop.ppsws.client.ps010

Java class for userTokenEnum complex type.

UserTokenService.java

Package: edu.ucop.ppsws.client.ps010

Authentication service for the PPS web services.

Adapter1.java

Package: edu.ucop.ppsws.client.ps011

GrossToNetItem.java

Package: edu.ucop.ppsws.client.ps011

A gross-to-net translation.

GrossToNetTranslator.java

Package: edu.ucop.ppsws.client.ps011

Service to return a list of gross-to-net translations for labor or retirement.

GrossToNetValues.java

Package: edu.ucop.ppsws.client.ps011

List of GrossToNetItems that make of the set of translations for the requested type.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps011

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps011 package.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS011Input.java

Package: edu.ucop.ppsws.client.ps011

The data transfer object containing the request data.

PS011InputWrapper.java

Package: edu.ucop.ppsws.client.ps011

PS011OUTPUT.java

Package: edu.ucop.ppsws.client.ps011

Data transfer object contain the server's response data.

PS011OutputWrapper.java

Package: edu.ucop.ppsws.client.ps011

PS011Service.java

Package: edu.ucop.ppsws.client.ps011

This class was generated by the JAX-WS RI.

Adapter1.java

Package: edu.ucop.ppsws.client.ps012

EmployeeIDMatcher.java

Package: edu.ucop.ppsws.client.ps012

The IID employee ID match check and assignment service.

EmployeeMatch.java

Package: edu.ucop.ppsws.client.ps012

Data that describes a matching employee.

MatchingEmployeesListing.java

Package: edu.ucop.ppsws.client.ps012

Wrapper for list of matching employees found by the service for the submitted employee identification data.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps012

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps012 package.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS012Input.java

Package: edu.ucop.ppsws.client.ps012
Java class for PS012INPUT complex type.

PS012InputElem.java

Package: edu.ucop.ppsws.client.ps012

PS012Output.java

Package: edu.ucop.ppsws.client.ps012
The payload returned by the service.

PS012OutputElem.java

Package: edu.ucop.ppsws.client.ps012

PS012Service.java

Package: edu.ucop.ppsws.client.ps012
The employee ID matching/assignment service.

Adapter1.java

Package: edu.ucop.ppsws.client.ps013

DataElementTranslation.java

Package: edu.ucop.ppsws.client.ps013
A data element translation.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps013
This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps013 package.

PS013Input.java

Package: edu.ucop.ppsws.client.ps013
Contains the request input values.

PS013InputElem.java

Package: edu.ucop.ppsws.client.ps013

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS013Output.java

Package: edu.ucop.ppsws.client.ps013

The payload returned by the service.

PS013OutputElem.java

Package: edu.ucop.ppsws.client.ps013

PS013Service.java

Package: edu.ucop.ppsws.client.ps013

The PPS code translation service.

RelatedDataElements.java

Package: edu.ucop.ppsws.client.ps013

The list of data element numbers related to this translation set.

RequestedDataElement.java

Package: edu.ucop.ppsws.client.ps013

Stores a data element number when using the RETURN_REQUESTED_DENS.

RequestedDataElements.java

Package: edu.ucop.ppsws.client.ps013

Stores data element list when using the RETURN_REQUESTED_DENS.

Translations.java

Package: edu.ucop.ppsws.client.ps013

The set of code translations.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps014

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps014 package.

PS014Input.java

Package: edu.ucop.ppsws.client.ps014

The data transfer object containing the request data

PS014InputElem.java

Package: edu.ucop.ppsws.client.ps014

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS014InputWrapper.java

Package: edu.ucop.ppsws.client.ps014

Wrapper around the data transfer object containing the request data.

PS014Output.java

Package: edu.ucop.ppsws.client.ps014

Contains the service's response data.

PS014OutputElem.java

Package: edu.ucop.ppsws.client.ps014

PS014OutputWrapper.java

Package: edu.ucop.ppsws.client.ps014

Wrapper around the data transfer object containing the service's response.

PS014Service.java

Package: edu.ucop.ppsws.client.ps014

This class was generated by the JAX-WS RI.

SystemControlRecord.java

Package: edu.ucop.ppsws.client.ps014

The system control record.

SystemControlRecordRetriever.java

Package: edu.ucop.ppsws.client.ps014

Service to get the current system control record.

ObjectFactory.java

Package: edu.ucop.ppsws.client.ps020

This object contains factory methods for each Java content interface and Java element interface generated in the edu.ucop.ppsws.client.ps020 package.

PS020Input.java

Package: edu.ucop.ppsws.client.ps020

The data transfer object holding the request data.

PS020InputWrapper.java

Package: edu.ucop.ppsws.client.ps020

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS020Output.java

Package: edu.ucop.ppsws.client.ps020

The data transfer object containing the service's response.

PS020OutputWrapper.java

Package: edu.ucop.ppsws.client.ps020

PS020Service.java

Package: edu.ucop.ppsws.client.ps020

This class was generated by the JAX-WS RI.

RequestProfile.java

Package: edu.ucop.ppsws.client.ps020

The set of profiles for which authorization is being requested.

RequestProfileItem.java

Package: edu.ucop.ppsws.client.ps020

The data describing a profile.

ResponseProfile.java

Package: edu.ucop.ppsws.client.ps020

The set of profiles along with the service's indication of the authorization decision for each.

ResponseProfileItem.java

Package: edu.ucop.ppsws.client.ps020

A profile and the authorization decision on it for the given user.

ScreenLevelAuthorizer.java

Package: edu.ucop.ppsws.client.ps020

Service that checks for a user's authorization to access various screens (aka profiles) in an application.

AppointmentImpl.java

Package: edu.ucop.ppsws.domain

AppointmentImpl.java implements IAppointment.java interface class for appointment related services for an employee.

AppointmentRowOutput.java

Package: edu.ucop.ppsws.domain

AppointmentRowOutput.java is the java bean associated with the "Appointments" screen.

BenefitsRowOutput.java

Package: edu.ucop.ppsws.domain

BenefitsRowOutput.java is the java bean associated with the "Benefits" screen.

DistributionImpl.java

Package: edu.ucop.ppsws.domain

DistributionImpl.java implements IDistribution.java interface class for distribution related services for an employee.

DistributionRowOutput.java

Package: edu.ucop.ppsws.domain

DistributionRowOutput.java is the java bean associated with the "Distributions" screen.

DuesAgencyFeeRow.java

Package: edu.ucop.ppsws.domain

DuesAgencyFeeRow.java is the java bean associated with the "Dues Agency Fee" screen.

EDBEmployee.java

Package: edu.ucop.ppsws.domain

EDBEmployee.java is the interface class for all of the employee related properties.

FamilyBenefitsRowOutput.java

Package: edu.ucop.ppsws.domain

FamilyBenefitsRowOutput.java is the java bean associated with the "Benefits - Family Member Enrollments" screen.

FinancialAidImpl.java

Package: edu.ucop.ppsws.domain

FinancialAidImpl.java implements IFinancialAid.java interface class for financial aid related services for an employee.

FinancialAidRowOutput.java

Package: edu.ucop.ppsws.domain

FinancialAidRowOutput.java is the java bean associated with the "Financial Aid" screen.

FutureBenefitsRowOutput.java

Package: edu.ucop.ppsws.domain

FutureBenefitsRowOutput.java is the java bean associated with the "Benefits - Future Benefits" screen.

HourBalancesDisplay.java

Package: edu.ucop.ppsws.domain

HourBalancesDisplay.java is the java bean associated with the "Hours Balances" screen.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

IAppointment.java

Package: edu.ucop.ppsws.domain

IAppointment.java is the interface class for appointment related properties for an employee.

IDistribution.java

Package: edu.ucop.ppsws.domain

IDistribution.java is the interface class for distributions related properties for an employee.

IFinancialAid.java

Package: edu.ucop.ppsws.domain

IFinancialAid.java is the interface class for Financial Aid related properties for an employee.

ILeaveAccrualPeriodSet.java

Package: edu.ucop.ppsws.domain

ILeaveAccrualPeriodSet.java is the interface class for leave accrual related properties for an employee.

InsuranceRowOutput.java

Package: edu.ucop.ppsws.domain

InsuranceRowOutput.java is the java bean associated with the "Insurance" screen.

IWorkStudy.java

Package: edu.ucop.ppsws.domain

IWorkStudy.java is the interface class for work study related properties for an employee.

LaborRelationsApptRow.java

Package: edu.ucop.ppsws.domain

LaborRelationsApptRow.java is the java bean associated with the "Labor Relations" screen.

LeaveAccrualHistoryRowOutput.java

Package: edu.ucop.ppsws.domain

LeaveAccrualHistoryRowOutput.java is the java bean associated with the "Leave Accrual" screen.

LeaveAccrualPeriodRowOutput.java

Package: edu.ucop.ppsws.domain

LeaveAccrualPeriodRowOutput.java is the java bean associated with the "Leave Accrual" screen.

LeaveAccrualPeriodSetImpl.java

Package: edu.ucop.ppsws.domain

LeaveAccrualPeriodSetImpl.java implements ILeaveAccrualPeriodSet.java interface class for leave accrual related services for an employee.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

LicenseRowOutput.java

Package: edu.ucop.ppsws.domain

LicenseRowOutput.java is the java bean associated with the "Licenses" screen.

MapBackedEDBEmpImpl.java

Package: edu.ucop.ppsws.domain

The implementation of EDBEmployee used in the PPS web applications.

RetirementGTNRowOutput.java

Package: edu.ucop.ppsws.domain

RetirementGTNRowOutput.java is the java bean associated with the "Retirements" screen.

WorkStudyImpl.java

Package: edu.ucop.ppsws.domain

WorkStudyImpl.java implements IWorkStudy.java interface class for work study related services for an employee.

WorkStudyRowOutput.java

Package: edu.ucop.ppsws.domain

WorkStudyRowOutput.java is the java bean associated with the "Work Study" screen.

EDBEmployeeDAO.java

Package: edu.ucop.ppsws.edb.dao

Defines the data access object for an EDB employee.

EmployeeDAOImpl.java

Package: edu.ucop.ppsws.edb.dao

Implements the EDBEmployeeDAO interface using the PS001 service, using an implementation of the EDBEmployeeAdapter interface.

EmployeeNotFoundException.java

Package: edu.ucop.ppsws.edb.dao

Raised when a given employee ID can't be found in the EDB.

MatchingEmp.java

Package: edu.ucop.ppsws.edb.dao

Interface defining the fields that are used by the IID service to describe an employee match.

IdentitySystemFacade.java

Package: edu.ucop.ppsws.edb.facade

Interface that exposes the web service support of the IID system to clients.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

IIDFacadeImpl.java

Package: edu.ucop.ppsws.edb.facade

Implementation of the identity system facade using PPS web services.

PPSWSTranslationServiceFacadeImpl.java

Package: edu.ucop.ppsws.edb.facade

Translation service facade implementation using PPS web services.

TokenServiceFacade.java

Package: edu.ucop.ppsws.edb.facade

Interface that exposes the authentication web service to clients.

TokenServiceFacadeImpl.java

Package: edu.ucop.ppsws.edb.facade

Implementation of the token service facade using PPS web services.

TranslationServiceFacade.java

Package: edu.ucop.ppsws.edb.facade

Interface that exposes the data element code translation web service to clients.

DataElementMapEmitter.java

Package: edu.ucop.ppsws.edb.util

Development tool that emits a java class that contains a method for each data element supplied in an input file.

EDBDataElement.java

Package: edu.ucop.ppsws.edb.util

Describes a PPS data element, including its data element number, its description, and whether or not it is scalar.

EDBDataElementMapper.java

Package: edu.ucop.ppsws.edb.util

Development tool that helps with generating strings that can later be used to define an EDBEmployee interface, and can help when the user needs to know if a certain data element is scalar or not.

ModifyServiceClassWsdUrl.java

Package: edu.ucop.ppsws.util

Modifies the Service class of a generated client to use a classpath resolution for the wsdl instead of the autogenerated url.

Other Files

README

Steps for generating a client for a PPS web service

PS001-jaxb-bindings.xml

JAXB binding file for PS001 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS001-jaxws-bindings.xml

JAXWS binding file for PS001 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS003-jaxb-bindings.xml

JAXB binding file for PS003 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS003-jaxws-bindings.xml

JAXWS binding file for PS003 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS004-jaxb-bindings.xml

JAXB binding file for PS004 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS004-jaxws-bindings.xml

JAXWS binding file for PS004 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS005-jaxb-bindings.xml

JAXB binding file for PS005 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS005-jaxws-bindings.xml

JAXWS binding file for PS005 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS010-jaxb-bindings.xml

JAXB binding file for PS010 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS010-jaxws-bindings.xml

JAXWS binding file for PS010 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS011-jaxb-bindings.xml

JAXB binding file for PS011 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS011-jaxws-bindings.xml

JAXWS binding file for PS011 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS012.wsdl

Web Services Description Language (WSDL) file. This WSDL defines the interface for calling web service PS012. (for future projects)

PS012-jaxb-bindings.xml

JAXB binding file for PS012 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa. (for future projects)

PS012-jaxws-bindings.xml

JAXWS binding file for PS012 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features. (for future projects)

PS013.wsdl

Web Services Description Language (WSDL) file. This WSDL defines the interface for calling web-service to obtain code translation values. (for future projects)

PS013-jaxb-bindings.xml

JAXB binding file for PS013 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa. (for future projects)

PS013-jaxws-bindings.xml

JAXWS binding file for PS013 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features. (for future projects)

PS014-jaxb-bindings.xml

JAXB binding file for PS014 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

PS014-jaxws-bindings.xml

JAXWS binding file for PS014 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

PS020-jaxb-bindings.xml

JAXB binding file for PS020 web service. Java Architecture for XML Binding (JAXB) is a specification that automates the mapping between XML documents and Java objects and vice versa.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PS020-jaxws-bindings.xml

JAXWS binding file for PS020 web service. The JAX-WS 2.0 specification defines standard XML-based customization for WSDL to Java mapping and to control certain features.

junit-3.8.1.jar

JUnit jar - JUnit is a simple java framework to write repeatable tests.

webservices-api.jar

webservices-api jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

webservices-extra.jar

webservices-extra jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

webservices-extra-api.jar

webservices-extra-api jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

webservices-rt.jar

webservices-rt jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

webservices-tools.jar

webservices-tools jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

wstx-services.war

Webservices jar file is part Java's Metro high-performance, extensible, easy-to-use web service stack

PPSWeb Java Application

PPSWeb is a new java web application which is comprised of the login, main menu and change password functions.

Properties Files

ApplicationResources.properties

PPSWeb application resources property file used by the Struts framework. It contains error message key to error message text mapping.

PPSWeb.properties

PPSWeb.properties contains the configurable variables for the PPSWeb application. The source copy must be modified with appropriate local values and copied to the web server in the directory identified by the PPSWEB_PROPERTY_HOME custom property in the jvm. The customizable properties are:

- database.context.name - The JNDI name of the data source specified in the application server.
- database.authid - The auth ID of the database.
- session_timeout - Time in minutes after which a session expires
- securityprefix - The prefix used for the RACF security profile for web screens.
- maxnumversions - The maximum number of EDB inquiry screen versions.
- edbinquiryserver - The url of the server hosting the EDB inquiry application.
- edbinquirycontextroot - The context root of the instance of EDB inquiry to link to.
- webpanurl - The url of the Web PAN application.
- webmeriturl - The url of the Web Merit application.

log4j.properties

log4j.properties contains the configurable properties for Apache. **log4j** is a Java-based logging utility. The log file location is defined by property log4j.appender.LOGFILE.File. Web application server administrator can control the location of the log file by defining a JVM custom property named APP_LOG_ROOT with the value of the directory path.

Java Classes

AccessEDBAction.java

Package: edu.ucop.ppsweb.actions

AccessEDBAction.java action class is executed whenever the user transfers to the EDB Inquiry system. It checks if the EDB Inquiry system is available.

ChangePasswordAction.java

Package: edu.ucop.ppsweb.actions

ChangePasswordAction.java is the action class associated with the change password screen.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

LogoffAction.java

Package: edu.ucop.ppsweb.actions

LogoffAction.java action class is executed whenever the user clicks on the logoff link on the PPSWeb menu. It logs off the user from the application and invalidates the session.

LogonAction.java

Package: edu.ucop.ppsweb.actions

LogonAction.java is the action class associated with the logon screen. It authorizes user id and password before displaying PPSWeb main menu.

MainMenuAction.java

Package: edu.ucop.ppsweb.actions

MainMenuAction.java action class is executed whenever the user clicks on the Main Menu link in PANWeb. It updates the session create timestamp.

PPSWebBusinessDelegate.java

Package: edu.ucop.ppsweb.business

PPSWebBusinessDelegate.java is the business delegate for providing various services.

StoredProcedureCalls.java

Package: edu.ucop.ppsweb.business.constants

StoredProcedureCalls.java defines the constants for stored procedure calls.

BaseDAO.java

Package: edu.ucop.ppsweb.dao

BaseDAO.java is the Base Data Access Object that defines common SQL and other methods used by extended classes.

CallspDAO.java

Package: edu.ucop.ppsweb.dao

CallspDAO.java contains call to various stored procedures. This class extends BaseDAO.java.

JDBCDataAccessor.java

Package: edu.ucop.ppsweb.dao

JDBCDataAccessor.java is used to set up the JDBC datasource and get a connection to the database.

DataAccessException.java

Package: edu.ucop.ppsweb.exception

DataAccessException.java throws DataAccessException when errors are encountered during database access.

AuthenticationException.java

Package: edu.ucop.ppsweb.exception

AuthenticationException.java throws AuthenticationException when login errors are encountered.

ChangePasswordException.java

Package: edu.ucop.ppsweb.exception

ChangePasswordException.java throws ChangePasswordException when errors are encountered during password change.

EDBAccessException.java

Package: edu.ucop.ppsweb.exception

EDBAccessException.java throws EDBAccessException when errors are encountered during login process.

EDDBaseException.java

Package: edu.ucop.ppsweb.exception

EDDBaseException.java is the base EDB exceptions class.

EDDBusinessException.java

Package: edu.ucop.ppsweb.exception

EDDBusinessException.java is a base class for EDB business exceptions. Business exceptions occur when a transaction cannot proceed due to a condition related to the data input by the user. They do not indicate there is a problem with the system. This class of exceptions is distinguished from system exceptions, which are caused by a system failure.

InvalidSessionException.java

Package: edu.ucop.ppsweb.exception

InvalidSessionException.java throws InvalidSessionException when user session is not valid.

AuthenticationFacade.java

Package: edu.ucop.ppsweb.facade

AuthenticationFacade.java encapsulate all of the services related to user authentication.

ChangeUserFacade.java

Package: edu.ucop.ppsweb.facade

ChangeUserFacade.java encapsulate all of the services related to user password change.

EDBAvailibility.java

Package: edu.ucop.ppsweb.facade

EDBAvailibility.java checks for the availability of the EDB Inquiry application.

LogUserOffFacade.java

Package: edu.ucop.ppsweb.facade

LogUserOffFacade.java encapsulates all of the services related to logging the user off the application.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

PPSWebAppProperties.java

Package: edu.ucop.ppsweb.resources

PPSWebAppProperties.java provides a global point of access for all of the PPS application properties. This class caches property values in the memory. It follows the Singleton design pattern, which ensures a class only has one instance.

ApplicationPropertiesLoader.java

Package: edu.ucop.ppsweb.util

ApplicationPropertiesLoader.java reads and loads the PPS application property file. The property file may be internal (in the class path) or external as defined by the value of JVM customer property PPSWEB_PROPERTY_HOME.

MiscUtil.java

Package: edu.ucop.ppsweb.util

MiscUtil.java contains various utility methods used across the PPSWeb application.

PPSWebValidator.java

Package: edu.ucop.ppsweb.util

PPSWebValidator sanitizes user input for enhanced security.

ScreenLevelSecurity.java

Package: edu.ucop.ppsweb.util

ScreenLevelSecurity.java builds a profile list for the user to control screen level security.

SessionCheck.java

Package: edu.ucop.ppsweb.util

SessionCheck.java checks if the user session stored in the session context is a valid session.

SessionFilter.java

Package: edu.ucop.ppsweb.util

SessionFilter.java implements the interface javax.servlet.Filter interface. A filter provides functionality that can be “attached” to any kind of web resource via web.xml mapping.

JSPs

changepassword.jsp

changepassword.jsp is the html format for the change password screen.

checksession.jsp

checksession.jsp is the html format to display the message informing the user that his/her session has expired.

login.jsp

login.jsp is the html format for the login screen.

logoff.jsp

logoff.jsp is the html format for the logoff screen.

mainmenu.jsp

mainmenu.jsp is the html format for the main menu.

sessioninvalid.jsp

sessioninvalid.jsp is the html format displayed when a user's session is invalid.

staticJavascript.jsp

staticJavascript.jsp is the struts validation framework specific JSP file for caching client-side form validations.

timeout.jsp

timeout.jsp is the html format to display the message to the user that his/her session has timed out.

Javascript

ppsweb.js

ppsweb.js contains all of the javaScript functions used by various PPSWeb applications. These are utility functions applicable across various PPSWeb screens.

Image Files

footerline.gif

footerline.gif is an image of a line which is used to separate screen sections in several screens.

login_background.gif

login_background.gif is the background image for the login screen.

uc.gif

uc.gif is the UC header logo

Other Files

pps.css

pps.css is the html style sheet for PPSWeb.

ibm-web-bnd.xml

This is an IBM binding file specific to WAS implementation.

ibm-web-ext.xml

This is an IBM extension to web.xml binding file specific to WAS implementation.

struts-config.xml

This file contains all of the routing and configuration information for the Struts application. This mainly includes form bean definition, global forward definition, action mapping definition, controller configuration and application Resources definition.

validation.xml

The **validation.xml** configuration file defines which validation routines must be applied for which Form Beans.

validator-rules.xml

The **validator-rules.xml** declares the validation routines that must be plugged-in, and logical names are provided for the validations. It also contains the client-side JavaScript code for each validation routine. The validation routines are java methods plugged into the system to perform specific validations.

web.xml

The web.xml file provides configuration and deployment information for the Web components that comprise a Web application.

antlr-2.7.2.jar

ANTLR jar file - ANOther Tool for Language Recognition (ANTLR) is the name of a parser generator that uses LL(*) parsing.

commons-beanutils-1.7.0.jar

Commons bean utility jar file (Apache) - The *BeanUtils* component provides easy-to-use wrappers around Java language's *Reflection* and *Introspection* APIs

commons-chain-1.1.jar

Commons chain jar file (Apache) - The fundamental API contracts for the Chain of Responsibility Pattern.

commons-digester-1.8.jar

Commons digester jar file (Apache) - The Digester package lets you configure an XML->Java object mapping module which triggers certain actions called rules whenever a particular pattern of nested XML elements is recognized.

commons-discovery-0.2.jar

Commons discovery jar file (Apache) – This provides facilities for instantiating classes in general, and for lifecycle management of singleton (factory) classes.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

commons-io-1.1.jar

Commons IO jar file (Apache) - Commons IO is a library of utilities to assist with developing IO functionality.

commons-logging-1.0.4.jar

Common Logging jar file (Apache) - Commons Logging is a thin adapter allowing configurable bridging to other, well known logging systems.

commons-validator-1.3.1.jar

Common Validator jar file (Apache) – Commons Validator provides the building blocks for both client side validation and server side data validation. It may be used standalone or with a framework like Struts.

jstl-1.0.2.jar

Jstl jar file (Sun) - The JavaServer Pages Standard Tag Library (JSTL) encapsulates as simple tags the core functionality common to many Web applications.

log4j-1.2.15.jar

Log4j jar file (Apache) – This is used during run-time to log various debug, informational or error messages in the application's log file.

oro-2.0.8.jar

Oro jar file (Apache) - ORO Java classes are a set of text-processing Java classes that provide Perl5 compatible regular expressions, AWK-like regular expressions, glob expressions, and utility classes for performing substitutions, splits, filtering filenames, etc.

standard-1.0.2.jar

Standard jar file (Apache) – This contains implementation classes of Sun's JSTL API classes.

struts-core-1.3.8.jar

Struts core jar file – This contains the core java classes for the Apache's struts framework.

struts-el-1.3.8.jar

Struts EL jar file - Struts EL is an extension of the Struts Taglib component that uses JSTL (the JavaServer Pages Standard Tag Library) as a foundation technology.

struts-extras-1.3.8.jar

Struts extras jar file - Struts Extras provides several additional classes within the Struts framework.

struts-faces-1.3.8.jar

Struts faces jar file – This contains an add-on library that supports the use of JavaServer Faces (JSF) user interface technology in a Struts-based web application, in place of the Struts custom tag libraries.

struts-scripting-1.3.8.jar

This component allows Struts Actions to be written in the scripting language of one's choice rather than as Java classes.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

struts-taglib-1.3.8.jar

Struts taglib jar file – This provides a set of JSP custom tag libraries that help developers create interactive form-based applications. There are tags to help with everything from displaying error messages to dealing with nested ActionForm beans.

EDBInquiry Java Application

Detail Screen Information

Each EDB inquiry detail screen is represented by three objects

- java action class
- a screen JSP
- image file for the navigation bar.

Below is a table of these objects for the detail screens. Each of these will be described in the sections that follow.

Screen	Action class	Screen JSP	Navigation bar image
Address Information	AddressAction.java	address_ppwwiadr.jsp	address.gif
Appointments & Distributions	AppointmentsAction.java	appointments_ppwwiapp.jsp	appointments.gif
Benefits	BenefitsAction.java	benefits_ppwwiben.jsp	benefits.gif
Family Member Benefits	FamilyBenefitsAction.java	familybenefits_ppwwifam.jsp	n/a
Future Benefits	FutureBenefitsAction.java	futurebenefits_ppwwifut.jsp	n/a
Citizenship	CitizenshipAction.java	citizenship_ppwwicit.jsp	citizenship.gif
Employment	EmploymentAction.java	employment_ppwwiemp.jsp	employment.gif
Financial Aid	FinaicalAidAction.java	financialaid_ppwwifin.jsp	financialaid.gif
Hours Balances	HourBalancesAction.java	hourbalances_ppwwipay.jsp	payhours.gif
Labor Relations	LaborRelationsAction.java	laborrelations_ppwwilab.jsp	laborrelations.gif
Leave Accrual	LeaveAccrualAction.java	leaveaccrual_ppwwilva.jsp	leaveaccrual.gif
Licenses	LicensesAction.java	licenses_ppwwilic.jsp	licenses.gif
Pay Disposition	PayDispositionAction.java	paydisposition_ppwwidsp.jsp	paydisposition.gif
Personal	PersonalAction.java	personal_ppwwiper.jsp	personal.gif
Retirement	RetirementAction.java	retirement_ppwwiret.jsp	retirement.gif
Taxes	TaxesAction.java	taxes_ppwwitax.jsp	taxes.gif
Work Study	WorkstudyAction.java	workstudy_ppwwiwrk.jsp	workstudy.gif

Properties Files

ApplicationResources.properties

EDBInquiry application resources property file used by the Struts framework. It contains error message key to error message text mapping.

EDBInquiry.properties

EDBInquiry.properties contains the configurable variables for the EDBInquiry application. The source copy must be modified with appropriate local values and copied to the web server in the directory identified by the EDBINQUIRY_PROPERTY_HOME custom property in the jvm. The customizable properties are

- database.context.name - The JNDI name of the data source specified in the application server.
- database.authid - The auth ID of the database.
- webppsserver - The url for the server hosting the PPSWeb application.
- webppsscontextroot - The context root for the PPSWeb application

- header_logo - The name of the image file to use for the header.
- session_timeout - Time in minutes after which a session expires
- securityprefix - The prefix used for the RACF security profile for web screens.
- maxnumversions - The maximum number of EDB inquiry screen versions.

log4j.properties

log4j.properties contains the configurable properties for Apache. **log4j** is a Java-based logging utility. The log file location is defined by property log4j.appender.LOGFILE.File. Web application server administrator can control the location of the log file by defining a JVM custom property named APP_LOG_ROOT with the value of the directory path.

Java Classes

AddressAction.java

Package: edu.ucop.edbinquiry.actions

AddressAction.java is the action class associated with the Address Information screen, address_ppwwiadr.jsp. It formats address fields including campus release codes from the EDBEmployee object and stores the formatted data back in that object.

AppointmentsAction.java

Package:edu.ucop.edbinquiry.actions

AppointmentsAction.java is the action class associated with the Appointment and Distribution Information screen, appointments_ppwwiapp.jsp. It loops through the list of appointments built by the method parseAppointmentsAndDistributions in EDBEmployeeAdapterImpl.java. For each appointment, a display row record is formatted and saved in the AppointmentRowOutput list. Similar processing is done for distributions associated with each appointment. The screen loops through the output display rows.

BenefitsAction.java

Package:edu.ucop.edbinquiry.actions

BenefitsAction.java is the action class associated with the Benefits Enrollment Information screen, benefits_ppwwiben.jsp. This screen has several data sections, i.e., benefit plans, summary of family benefits, links to future and detailed family benefits, and insurance information. It formats fields from the EDBEmployee object and stores the formatted data back in that object. The benefits and insurance information is written to separate arrays of output display rows printed by the jsp.

BrowseAction.java

Package:edu.ucop.edbinquiry.actions

BrowseAction.java is the action class associated with the Employee Browse screen, browse_ppwwibrs.jsp.

CitizenshipAction.java

Package:edu.ucop.edbinquiry.actions

CitizenshipAction.java is the action class associated with the Citizenship/Visa Status Information screen, citizenship_ppwwicit.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

EDBCoreAction.java

Package:edu.ucop.edbinquiry.actions

EDBCoreAction.java extends the Struts Action class. Action class is an adapter between the contents of an incoming HTTP request and the corresponding business logic that should be executed to process this request. All EDB Inquiry action classes extend this class. The execute method in this class checks to see if the user is authorized to view the requested screen.

EmploymentAction.java

Package:edu.ucop.edbinquiry.actions

EmploymentAction.java is the action class associated with the Employment Status Information screen, employment_ppwwiemp.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

FamilyBenefitsAction.java

Package:edu.ucop.edbinquiry.actions

FamilyBenefitsAction.java is the action class associated with the Family Member Enrollments screen, familybenefits_ppwwifam.jsp (viewed from the Benefits Enrollment Information screen). FormatFamilyBenefits.java is called by this action class and by BenefitsAction.java to format family benefits information.

FinancialAidAction.java

Package:edu.ucop.edbinquiry.actions

FinancialAidAction.java is the action class associated with the Financial Aid Information screen, financialaid_ppwwifin.jsp. Financial aid fiscal years are stored in the database as integers, not as dates. The method parseFinancialAid in EDBEmployeeAdapterImpl.java builds the list of financial aid input records with the year data as dates so the records can be sorted appropriately. FinancialAidAction.java creates a list of formatted output display rows read by the jsp.

FooterAction.java

Package:edu.ucop.edbinquiry.actions

FooterAction.java is the action class associated with the footer screen, footer.jsp, displayed at the bottom of each screen. The footer is where an employee can be selected by name, SSN or ID.

FutureBenefitsAction.java

Package:edu.ucop.edbinquiry.actions

FutureBenefitsAction.java is the action class associated with the Future Benefits Enrollment Information screen, futurebenefits_ppwwifut.jsp (viewed from the Benefits Enrollment Information screen). It formats fields from the EDBEmployee object and creates an array of output display rows read by the jsp.

HeaderAction.java

Package:edu.ucop.edbinquiry.actions

HeaderAction.java is the action class associated with the header screen, header.jsp, displayed at the top of each screen. The header displays the name of the current selected employee and other thumbnail information.

HourBalancesAction.java

Package:edu.ucop.edbinquiry.actions

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

HourBalancesAction.java is the action class associated with the Hours Balances screen, hourbalances_ppwwipay.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

LaborRelationsAction.java

Package:edu.ucop.edbinquiry.actions

LaborRelationsAction.java is the action class associated with the Labor Relations Information screen, laborrelations_ppwwilab.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object. Data for the Dues/Agency Fee Information section is determined by testing the list of GTN's associated with the employee to a static list of GTN's that apply.

LeaveAccrualAction.java

Package:edu.ucop.edbinquiry.actions

LeaveAccrualAction.java is the action class associated with the Leave Accrual History screen, leaveaccrual_ppwwilva.jsp. The method parseLeaveAccrualPeriods in EDBEmployeeAdapterImpl.java first builds the list of leave accrual input records. LeaveAccrualAction.java creates a list of formatted output display rows read by the jsp. For each leave accrual period, four history rows are created and added to a list associated with each leave accrual period.

LicensesAction.java

Package:edu.ucop.edbinquiry.actions

LicensesAction.java is the action class associated with the Licenses and Certificates Information screen, licenses_ppwwilic.jsp. It loops through the input array list of license information created by the method OccurrenceValuePair in EDBEmployeeAdapterImpl.java and formats output display rows to be printed by the jsp.

LogoffAction.java

Package:edu.ucop.edbinquiry.actions

LogoffAction.java is the action class invoked when a user logs off.

MainMenuAction.java

Package:edu.ucop.edbinquiry.actions

MainMenuAction.java clears the EDB session data on the way back to the main menu.

PayDispositionAction.java

Package:edu.ucop.edbinquiry.actions

PayDispositionAction.java is the action class associated with the Pay Disposition Information screen, paydisposition_ppwwidsp.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

PersonalAction.java

Package:edu.ucop.edbinquiry.actions

PersonalAction.java is the action class associated with the Personal Information screen, personal_ppwwiper.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

RetirementAction.java

Package:edu.ucop.edbinquiry.actions

RetirementAction.java is the action class associated with the Retirement Information screen, retirement_ppwwiret.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object. Data for the GTN section is determined by testing the list of GTN's associated with the employee to a static list of GTN's that apply.

StartEDBAction.java

Package:edu.ucop.edbinquiry.actions

StartEDBAction.java is the action class for the entry into the EDB Inquiry System.

TaxesAction.java

Package:edu.ucop.edbinquiry.actions

TaxesAction.java is the action class associated with the Tax Information screen, taxes_ppwwitax.jsp. It formats fields from the EDBEmployee object and stores the formatted data back in that object.

WorkstudyAction.java

Package:edu.ucop.edbinquiry.actions

WorkstudyAction.java is the action class associated with the Address Information screen, workstudy_ppwwiwrk.jsp. Like FinancialAidAction.java, fiscal year data is stored in the database as integers, not as dates. The method parseWorkStudy in EDBEmployeeAdapterImpl.java builds the list of work study input records with the year data as dates so the records can be sorted appropriately. WorkStudyAction.java creates a list of formatted output display rows read by the jsp.

EDBBusinessDelegate.java

Package:edu.ucop.edbinquiry.business

EDBBusinessDelegate.java is the business delegate for providing various services.

FormatFamilyBenefits.java

Package:edu.ucop.edbinquiry.business

FormatFamilyBenefits.java formats data for the family benefits portion of the main benefits screen AND the family benefits enrollment screen.

EDBBusinessDelegate.java

Package:edu.ucop.edbinquiry.business

EDBBusinessDelegate.java is the business delegate for providing various services.

EDBInquiryConstants.java

Package:edu.ucop.edbinquiry.business.constants

EDBInquiryConstants.java defines the constants for EDB Inquiry.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

StoredProcedureCalls.java

Package:edu.ucop.edbinquiry.business.constants

StoredProcedureCalls.java defines the constants for stored procedure calls.

BaseDAO.java

edu.ucop.edbinquiry.dao

BaseDAO.java is the base data access object which defines common SQL and other methods used by extended classes.

CallspDAO.java

edu.ucop.edbinquiry.dao

CallspDAO.java contains call to various stored procedures.

JDBCDataAccessor.java

edu.ucop.edbinquiry.dao

JDBCDataAccessor.java is used to set up the JDBC datasource and get a connection to the database.

BrowseOutput.java

Package: edu.ucop.edbinquiry.dto

BrowseOutput.java is a bean pertaining to the browse screen output row.

BrowseParams.java

Package:edu.ucop.edbinquiry.dto

BrowseParams.java bean contains the parameters required to browse/navigate through employee browse list.

DataAccessException.java

Package:edu.ucop.edbinquiry.exception

DataAccessException.java is the general exception thrown by the data layer/ DAO methods.

EDBBaseException.java

Package:edu.ucop.edbinquiry.exception

EDBBaseException.java is the base EDB exceptions class.

EDBBusinessException.java

Package:edu.ucop.edbinquiry.exception

EDBBusinessException.java is a base class for EDB business exceptions. Business exceptions occur when a transaction cannot proceed due to a condition related to the data input by the user. These do not indicate there is a problem with the system. This class of exceptions is distinguished from system exceptions, which are caused by a system failure.

EmployeeMatchException.java

Package:edu.ucop.edbinquiry.exception

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

EmployeeMatchException.java throws EmployeeMatchException when a unique match is not found for entered name or SSN.

BrowseFacade.java

Package:edu.ucop.edbinquiry.facade

BrowseFacade.java encapsulates all of the services related to browsing employee list.

EmployeeInfoFacade.java

Package:edu.ucop.edbinquiry.facade

EmployeeInfoFacade.java encapsulates the services to get employee info based on employee id.

GlobalValuesFacade.java

Package:edu.ucop.edbinquiry.facade

GlobalValuesFacade.java gets the global data for the EDB application and stores in the session.

UniqueMatchFacade.java

Package:edu.ucop.edbinquiry.facade

UniqueMa5chFacade.java encapsulates all of the services related to employee search based on employee name or social security number.

BrowseForm.java

Package:edu.ucop.edbinquiry.forms

BrowseForm.java is form bean associated with the browse action.

HeaderForm.java

Package:edu.ucop.edbinquiry.forms

HeaderForm.java is form bean associated with the header action.

EDBAppProperties.java

Package:edu.ucop.edbinquiry.resources

EDBAppProperties.java provides a global point of access for all of the PPS application properties. This class caches property values in memory. It follows the Singleton design pattern, which ensures a class only has one instance.

ApplicationPropertiesLoader.java

Package:edu.ucop.edbinquiry.util

ApplicationPropertiesLoader.java loads the property values defined in the property file. It also gets the global data for the EDB application and stores in the session.

EDBValidator.java

Package: edu.ucop. edbinquiry.util

EDBValidator sanitizes user input for enhanced security.

Misc.java

Package:edu.ucop.edbinquiry.util

Misc.java contains various utility methods used across the EDBInquiry application.

ScreenLevelSecurity.java

Package:edu.ucop.edbinquiry.util

ScreenLevelSecurity.java builds a profile list for the user to control screen level security.

SessionFilter.java

Package:edu.ucop.edbinquiry.util

SessionFilter.java implements the interface javax.servlet.Filter interface. A filter provides functionality that can be “attached” to any kind of web resource via web.xml mapping.

Translator.java

Package:edu.ucop.edbinquiry.util

Translator.java contains utility classes to convert various data elements.

JSPs

address_ppwwiadr.jsp

address_ppwwiadr.jsp is the html format for the address information screen. Data is displayed from the employeeInfo object.

appointments_ppwwiapp.jsp

appointments_ppwwiapp.jsp is the html format for the appointment and distribution screen. Data is displayed from the employeeInfo object.

benefits_ppwwiben.jsp

benefits_ppwwiben.jsp is the html format for the benefits information screen. Data is displayed from the employeeInfo object.

blank.jsp

blank.jsp is the html format for subsidiary screens called by some detail screens, i.e., benefits_ppwwiben.jsp. The purpose of blank.jsp is to display the same background color on any such secondary screens.

browse_ppwwibrs.jsp

browse_ppwwibrs.jsp is the html format for the address information screen. Data is displayed from the employeeInfo object.

citizenship_ppwwicit.jsp

citizenship_ppwwicit.jsp is the html format for the citizenship information screen. Data is displayed from the employeeInfo object.

detail_print.jsp

detail_print.jsp is the "wrapper" html format used when printing any of the individual detail screens.

detail_screen.jsp

detail_screen.jsp is the "wrapper" html format for all the individual detail screens.

edb.jsp

edb.jsp is the html format for all 3 frames used for display.

employeeelection.jsp

employeeelection.jsp is the html format for essentially the home page of this application, or where the user enters the first selections.

employment_ppwwiemp.jsp

employment_ppwwiemp.jsp is the html format for the employment status information screen. Data is displayed from the employeeInfo object.

error.jsp

error.jsp is the html format for error screen. All the action classes forward to this page in case of some error.

familybenefits_ppwwifam.jsp

familybenefits_ppwwifam.jsp is the html format for the family member benefits screen. Data is displayed from the employeeInfo object. It displays data from the familyBenefitsDisplayRows array, if any.

financialaid_ppwwifin.jsp

financialaid_ppwwifin.jsp is the html format for the financial aid information screen. Data is displayed from the employeeInfo object. It displays data from the financialAidDisplayRows array, if any.

footer.jsp

footer.jsp is the html format for the footer displayed at the bottom of the screen.

futurebenefits_ppwwifut.jsp

futurebenefits_ppwwifut.jsp is the html format for the future benefits data screen. Data is displayed from the employeeInfo object. It displays data from the futureBenefitsDisplayRows array, if any.

header.jsp

header.jsp is the html format for the header information shown at the top of each page.

hourbalances_ppwwipay.jsp

hourbalances_ppwwipay.jsp is the html format for the hours balances screen. Data is displayed from the employeeInfo object.

laborrelations_ppwwilab.jsp

laborrelations_ppwwilab.jsp is the html format for the labor relations screen. Data is displayed from the employeeInfo object. It displays the items in the duesAgencyFeeRows array and the laborRelationsApptRows array, if any.

leaveaccrual_ppwwilva.jsp

leaveaccrual_ppwwilva.jsp is the html format for the leave accrual screen. Data is displayed from the employeeInfo object.

licenses_ppwwilic.jsp

licenses_ppwwilic.jsp is the html format for the licenses and certificates information screen. Data is displayed from the employeeInfo object.

navigation.jsp

navigation.jsp is the html format for the navigation action buttons displayed on the left side of each screen.

notauthorized.jsp

notauthorized.jsp is the html format for the screen which appears when a user not authorized for EDB inquiry first attempts to get into EDB inquiry.

paydisposition_ppwwidsp.jsp

paydisposition_ppwwidsp.jsp is the html format for the pay disposition screen. Data is displayed from the employeeInfo object.

This screen supports two versions - the bank account number is suppressed for version 1, otherwise it is shown. The version authorized for the logged in user is obtained from the screen_level_version_map contained in the session. Bank data is displayed only if there is non-null data in directDepositPrenoteInd0227. Check address data is displayed only if checkDispCode0201 is between 5 and 7.

personal_ppwwiper.jsp

personal_ppwwiper.jsp is the html format for the personal data screen. Data is displayed from the employeeInfo object.

retirement_ppwwiret.jsp

retirement_ppwwiret.jsp is the html format for the retirement information screen. Data is displayed from the employeeInfo object. It displays the items in the retirementGTNDisplayRows array, if any.

taxes_ppwwitax.jsp

taxes_ppwwitax.jsp is the html format for the tax information screen. Data is displayed from the employeeInfo object.

workstudy_ppwwiwrk.jsp

workstudy_ppwwiwrk.jsp is the html format for the work study screen. Data is displayed from the employeeInfo object.

Javascript

EDBInquiry.js

EDBInquiry.js contains all the javaScript functions used by various EDBInquiry screens. This also contains javaScript functions used for navigation within the EDBInquiry application.

Image Files

address.gif

address.gif is the menu bar image for Address

appointments.gif

appointments.gif is the menu bar image for Appointments

benefits.gif

benefits.gif is the menu bar image for Benefits

browse.gif

browse.gif is the menu bar image for Browse

citizenship.gif

citizenship.gif is the menu bar image for Citizenship

employment.gif

employment.gif is the menu bar image for Employment

financialaid.gif

financialaid.gif is the menu bar image for Financial Aid

footerline.gif

footerline.gif is the Line image

laborrelations.gif

laborrelations.gif is the menu bar image for Labor Relations

leaveaccrual.gif

leaveaccrual.gif is the menu bar image for Leave Accrual

licenses.gif

licenses.gif is the menu bar image for Licenses

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

login_background.gif

login_background.gif is the Background image for login screen

paydisposition.gif

paydisposition.gif is the menu bar image for Pay Disposition

payhours.gif

payhours.gif is the menu bar image for Hours Balances

PayPerSys3A.jpg

PayPerSys3A.jpg PPS logo

personal.gif

personal.gif is the menu bar image for Personal

printerFriendly.gif

printerFriendly.gif is the Printer-Friendly image

retirement.gif

retirement.gif is the menu bar image for Retirement

taxes.gif

taxes.gif is the menu bar image for Taxes

uc.gif

uc.gif is the UC Logo

workstudy.gif

workstudy.gif is the menu bar image for Work Study

Other Files

edbinquiry.css

edbinquiry.css is the html style sheet used for the EDBInquiry application.

ibm-web-bnd.xmi

This is an IBM binding file specific to WAS implementation.

ibm-web-ext.xmi

This is an IBM extension to web.xml binding file specific to WAS implementation.

struts-config.xml

This file contains all of the routing and configuration information for the Struts application. This mainly includes form bean definition, global forward definition, action mapping definition, controller configuration and application Resources definition.

validation.xml

The **validation.xml** configuration file defines which validation routines must be applied for which Form Beans.

validator-rules.xml

The **validator-rules.xml** declares the validation routines that must be plugged-in, and logical names are provided for the validations. It also contains the client-side javascript code for each validation routine. The validation routines are java methods plugged into the system to perform specific validations.

web.xml

The web.xml file provides configuration and deployment information for the Web components that comprise a Web application.

antlr-2.7.2.jar

ANTLR jar file - ANOther Tool for Language Recognition (ANTLR) is the name of a parser generator that uses LL(*) parsing.

commons-beanutils-1.7.0.jar

Commons bean utility jar file (Apache) - The *BeanUtils* component provides easy-to-use wrappers around Java language's *Reflection* and *Introspection* APIs.

commons-chain-1.1.jar

Commons chain jar file (Apache) - The fundamental API contracts for the Chain of Responsibility Pattern.

commons-digester-1.8.jar

Commons digester jar file (Apache) - The Digester package lets you configure an XML->Java object mapping module which triggers certain actions called rules whenever a particular pattern of nested XML elements is recognized.

commons-discovery-0.2.jar

Commons discovery jar file (Apache) – This provides facilities for instantiating classes in general, and for lifecycle management of singleton (factory) classes.

commons-io-1.1.jar

Commons IO jar file (Apache) - Commons IO is a library of utilities to assist with developing IO functionality.

commons-logging-1.0.4.jar

Common Logging jar file (Apache) - Commons Logging is a thin adapter allowing configurable bridging to other, well known logging systems.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

commons-validator-1.3.1.jar

Common Validator jar file (Apache) – Commons Validator provides the building blocks for both client side validation and server side data validation. It may be used standalone or with a framework like Struts.

jstl-1.0.2.jar

Jstl jar file (Sun) - The JavaServer Pages Standard Tag Library (JSTL) encapsulates as simple tags the core functionality common to many Web applications.

log4j-1.2.15.jar

Log4j jar file (Apache) – This is used during run-time to log various debug, informational or error messages in the application's log file.

oro-2.0.8.jar

Oro jar file (Apache) - ORO Java classes are a set of text-processing Java classes that provide Perl5 compatible regular expressions, AWK-like regular expressions, glob expressions, and utility classes for performing substitutions, splits, filtering filenames, etc.

standard-1.0.2.jar

Standard jar file (Apache) – This contains implementation classes of Sun's JSTL API classes.

struts-core-1.3.8.jar

Struts core jar file – This contains the core java classes for the Apache's struts framework.

struts-el-1.3.8.jar

Struts EL jar file - Struts EL is an extension of the Struts Taglib component that uses JSTL (the JavaServer Pages Standard Tag Library) as a foundation technology.

struts-extras-1.3.8.jar

Struts extras jar file - Struts Extras provides several additional classes within the Struts framework.

struts-faces-1.3.8.jar

Struts faces jar file – This contains an add-on library that supports the use of JavaServer Faces (JSF) user interface technology in a Struts-based web application, in place of the Struts custom tag libraries.

struts-scripting-1.3.8.jar

This component allows Struts Actions to be written in the scripting language of one's choice rather than as Java classes.

struts-taglib-1.3.8.jar

Struts taglib jar file – This provides a set of JSP custom tag libraries that help developers create interactive form-based applications. There are tags to help with everything from displaying error messages to dealing with nested ActionForm beans.

Forms

UPAY930 (new)

UPAY930 is a new form containing the run specification for program UCSSIMNT. See Appendix E.

Control Table Updates

System Messages Table

The following message text will be modified:

1. WA005
Old text - Invalid password
New text - Login or password incorrect

The following obsolete messages will be deleted:

2. WA004 Application error, notify programmer, code=
3. WA006 Password successfully changed
4. WA007 Installation error, notify programmer, code=
5. WA010 Unknown internal error, notify programmer, code=
6. WA012 Invalid old password
7. WA013 Userid has been revoked
8. WA021 Your user ID is not authorized for Web EDB Inquiry.
9. WA022 Your access to EDB Inquiry could not be verified due to error=

Obsolete Objects

All PPS net.data objects are now obsolete. This includes all members of the following directories:

- o WHTML
- o WIMAGES
- o WINCLUDE
- o WMACRO
- o WSOURCE

Appendix A

Technical Note on web services with unbounded arrays

Some of the web services require unbounded array to contain a large number of input and output elements. The RADz tools do not successfully implement unbounded array. Therefore, a multi-step approach will be used for generating the web service programs:

1. Copylib members PSxxx describe the input and output structures for the web services. These copylib members will not be copied into any programs. They will be exclusively used to generate the WSDLs for the services using RADz (bottom-up approach).
2. Once the WSDLs are generated, they must be manually modified so that the arrays are unbounded.
3. The WSDLs are then used in RADz to generate the web service shell programs PSxxx. This process will also generate the interface copylib members PSxxxI01 and PSxxxO01.

Important: Note that any changes to the input and output structure will cause copylib member PSxxx to change. This will require that the associated wsdl be re-generated and modified again as in steps 1 & 2 above. After that, step 3 must be repeated to re-generate the interface copylib members, but the shell program which is generated by this process should not be used. Instead, the PSxxx Cobol program should be maintained manually.

Appendix B

Steps for Customizing the Applications

1. Import the the source code into RAD. The source code is provided in release file PAYDIST.R1873.PPSZIP. This will cause four project files to be created: EDBInquiry, PPSWeb, PPSWebAppsEar and PPSWebServices.
2. Code and test changes.
3. In RAD, right-click on project PPSWebAppsEar and select "Prepare for Deployment".
4. Export PPSWebAppsEar to an .ear file and deploy to the web application server.

Appendix C

How to Add Data Elements to an EDB Inquiry Screen

To add data elements to an EDB Inquiry screen, it is necessary to modify components of both the PPSWebServices and EDBInquiry applications. Familiarity with RAD is assumed.

1. Import the the source code into RAD (see Appendix B).
2. In the PPSWebServices project you will find the EDBEmployee Interface in package “edu.ucop.ppsws.domain”.

- Add a get method for the new field, eg.

```
public String getColbargUnitcode0161();
```

Each field name should be in camel case, with individual words expanded to be readable, and ending with the data element number.

- If the data element requires a code translation or formatting (eg. dates), add get and set methods for the formatted field, eg.

```
public String getCheckDispCode0201Formatted();
```

```
public void setCheckDispCode0201Formatted(String checkDispCode0201Formatted);
```

3. The get & set methods added above have to be implemented in MapbackedEDBEmpImpl class located in the edu.ucop.ppsws.domain” package. Note that this is only for fields which require formatting or translating.

- If a formatted field was added above, add a private string to hold the formatted field, e.g.

```
private String checkDispCode0201Formatted;
```

- Add get and set methods for the raw and formatted fields.
- If the element is scalar (non-recurring), the get method is as follows:

```
public String getCampusAddrLine10331() {  
    return scalarDataElemValues.get( "0331" );  
}
```

- If the element is non-scalar (recurring), and only one specific occurrence value is needed the get method is as follows:

```
public String getYtdTaxWithHeld6012Y() {  
    String value = null;  
    Map<String,String> occurrences = repeatingDataElemsAsMaps.get( "6000"  
);  
    if( occurrences != null ){  
        value = occurrences.get( "012Y" );  
    }  
    return value;  
}
```

- If the element is non-scalar (recurring), and the entire list is needed the get method is as follows:

```
public List<IOccurrenceValuePair> getEmplActionOneD0001() {  
    return repeatingDataElemsAsLists.get( "0001" );  
}
```

```
public void setCheckDispCode0201Formatted(String checkDispCode0201Formatted);
```

4. If the data element requires formatting:
 - In EDBInquiry, locate the action class for the screen to which the data element will be added. This will be in package edu.ucop.edbinquiry.actions.
 - In method formatData, add code to format the element and save it in the EDBEmployee object.
5. Locate the jsp for the screen to which the data element will be added. This can be found in folder WebContent/jsp. Add html for either the raw or formatted element by referring to the field in the employee object, e.g. `${employeeInfo.countryOfResidency0143Formatted}`
6. Export PPSWebAppsEar to an .ear file for deployment (See Appendix B).

Appendix D

How to Create Alternate Versions of an EDB Inquiry Screen

The existing net.data EDB Inquiry application supports "screen versioning" to give campuses the ability to display the same screen, with more or less data, based upon RACF security rules as well as campus wide requirements. The java rewrite includes this feature as well.

The pay disposition screen is released with two versions - the bank account number is suppressed for version 1, otherwise it is shown. The same technique can be used for other campus-specific versions. The RACF requirements are unchanged.

The following code is from paydisposition_ppwwidsp.jsp showing how alternate versions are coded into a single jsp:

```
<%-- *****  
*** Don't show the bank account number for version 1  
*** Otherwise, show it  
***** --%>  
<c:if test="`${sessionScope.screen_level_version_map.PPWWIDSP lt '1'}">  
  <tr>  
    <td width="20%" align="RIGHT" class="ROWLABEL">Account Number:</td>  
    <td colspan=1 align="LEFT"  
      class="DATA">`${employeeInfo.directDepositBankAcctNumber0225}</td>  
  </tr>  
</c:if>
```

The version for which the logged in user is authorized will always be available in sessionScope.screen_level_version_map.NNNNNNNN, where NNNNNNNN is the screen key. This value can be interrogated in the jsp to control which fields will be displayed. The screen key is hard-coded in each screen's action class, in the method getScreenKey.

Release 1873
Web Inquiry Rewrite SR82355
Detailed Specification
July 21, 2009

Appendix E: UPAY930 - UCSSIMNT Program Run Specification



PAYROLL/PERSONNEL
UCSSIMNT RUN SPECIFICATION
 UPAY930 (R07/09) R1873

UCSSIMNT Run Specification - Required

PROGRAM ID	OVERALL ACTION
1 14	15
UCSSIMNT-SPEC	

OVERALL ACTION: One of the values below must be entered:
 D - Disable All Subsystems
 E - Enable All Subsystems
 S - Enable/Disable Selected Subsystems in Additional Cards Below

Additional cards for subsystem specification - required for Overall Action = S, optional for others.

ACTION	SUBSYSTEM
1	2 3

ACTION: One of the values below must be entered:
 D - Disable Subsystem
 E - Enable Subsystem

Subsystem - Any valid "UCROUTER" subsystem may be entered, eg.

- DP Departmental Inquiry
- DX Employee Documents
- EH EDB/HDB Inquiry
- EI EDB Inquiry
- ET Dept. Time Reporting
- EU EDB Entry/Update
- FC CSER Entry/Update
- HD History Inquiry
- HR History Reporting
- ID ID Numbers
- OR Special Processes
- PI PAR Inquiry
- PN PostAuth Notification
- SA System Administration
- TD Dept. Time Reporting
- TF THF Entry/Update
- TR Dept. Time Reporting

PREPARED BY:	DATE	APPROVED BY:	DATE
--------------	------	--------------	------

RETN: UNTIL ACTION TAKEN