University of California ~ Effort Reporting System Customization Guide

Contents

Introduction

The Effort Reporting System was designed to be configurable and customizable to support the specific needs of each installation. In preparing to customize the system, it is useful to review the types of configuration and customization that might be needed for a particular installation.

<u>Configuration</u>
Configuration means changing system behavior by modifying one or more options that are provided by the application configuration file. Configuration can be done by the installer or a programmer after installation. For ERS, configuration is accomplished by editing settings in the application configuration file ***ERSConfig.XML***

<u>Customization</u>
Customization means changing the system "look and feel" by modifying the fonts, colors or graphics used for the web application. This process is commonly known as "skinning". Customization may be done by a web designer or HTML programmer.   This task requires editing CSS files, possibly editing HTML or JSP files, creating new graphics and installing these modified objects in the appropriate application folders.

<u>Modification</u>
Modification means changing the system behavior by modifying source code. This type of customization will require Java programming skills and the changes are introduced by editing source code, recompiling and redeploying the application in WAR format. In order to modify the ERS source code, the installation must establish the required development environment with all of the products described in the "ERS Software Specification" document.

Configuration File: ERSConfig.XML

The ERS configuration file contains parameters which control the behavior of many aspects of the Effort Reporting System. It is a structured XML document that will allow the installer to set up the Effort Reporting System to meet installation specific needs.

The root element of this document is the "Configuration" element. The "root" element (Configuration) is only mentioned in the first entry.

Throughout this section, the element descriptions will use a dotted notation to refer to the sub-elements. For example, an entry in the XML file with the following structure:
`<GlobalItems><ApplicationBaseURL>XXX</ApplicationBaseUrl></GlobalItems>`

will be shown as:
`GlobalItems.ApplicationBaseURL` with a supplied value of: XXX

Configuration Element	Supplied Value
Description	
Configuration	url=http://www.ucop.edu
<string> the URL of the campus	
	campusName=UC Office of the President
<string> text name of the campus	
GlobalItems.ApplicationBaseURL	http://localhost:8080/EffortReporting
<string> URL of index.jsp for the application	
GlobalItems.TrainingBaseURL	http://localhost:8080/EffortReporting/Training
<string> URL of Training website for linking from online help	
GlobalItems.SupportEmail.Name	Effort Reporting System
<string> text Name of the Support Email User	
GlobalItems.SupportEmail.Address	ersadmin@ucop.edu
<string> Email Address of the Support Email User	
AccessControl.Authentication.internal-authentication	true
whether the authentication method used is internal (true) or external (false)	
AccessControl.Authentication.external.token-key-name	username
Name of the fields in the HTTP request object that contains the identity strings to pass to a custom UserContext builder. This element may be repeated for as many authorization tokens as are needed.	
AccessControl.Authentication.external.error-redirect-url	Test
a URL to which the user will be redirected in the event of an authentication validation failure in the custom UserContext builder class (only used in externally authenticated instances of ERS)	
AccessControl.Authentication.external.signout-redirect-url	Test
a URL to which the user will be redirected when clicking any "Sign Out" link in the application. (only used in externally authenticated instances of ERS)	
AccessControl.Authentication.external.timeout-redirect-url	Test
a URL to which the user will be redirected when session times out due to inactivity. (only used in externally authenticated instances of ERS)	
AccessControl.Authorization.internal-authorization	true

Configuration Element	Supplied Value
Description	
whether the authorization method used is internal (true) or external (false)	
AccessControl.Authorization.decentralized-administration	false
whether the decentralized security administration features are enabled (true) or disabled (false)	
AccessControl.Authorization.authorization-package	edu.ucop.ers.access
<string> package name containing authorization classes	
AccessControl.Authorization.authorization-class	UCOPAuthorizer
<string> class that will check requests to perform functions	
AccessControl.Authorization.authorizer-mgr-package	edu.ucop.ers.access
<string> package name containing the authorization classes	
AccessControl.Authorization.authorizer-mgr-class	UCOPAuthorizerManagerImpl
<string> name of the class that implements the IAuthorizer Interface to determine whether a logged-in user may perform a given function on any one of a list of specific resources	
AccessControl.Authorization.my-report-authorizer-package	edu.ucop.ers.access
<string> name of the package containing the authorization classes	
AccessControl.Authorization.my-report-authorizer-class	UCOPMyReportAuthorizerImpl
<string> name of the class of the AuthorizerManager that takes care of reports designated as My Reports	
AccessControl.Authorization.my-project-authorizer-package	edu.ucop.ers.access
<string> package name containing the authorization classes	
AccessControl.Authorization.my-project-authorizer-class	UCOPMyProjectAuthorizerImpl
<string> name of the class of the AuthorizerManager that takes care of reports designated My Projects.	
AccessControl.Authorization.my-roles-checker-package	edu.ucop.ers.access
<string> package name containing the authorization classes	
AccessControl.Authorization.my-roles-checker-class	UCOPMyRolesCheckerImpl
<string> name of the class of the AuthorizerManager that determines which role(s) a user possesses.	
AccessControl.Authorization.uc-builder-package	edu.ucop.ers.userinfo
<string> package name containing the authentication classes	
AccessControl.Authorization.uc-builder-class	UCOPUserContextBuilderImplementation
<string> name of the class that builds the User Context consistent with the IUserContext Interface. Whether you use the UCOP Authorization or not, your implementation must satisfy IUserContext. If you use UCOP authorization, your implementation must additionally satisfy IUCOPUserContext.	
AccessControl.Authorization.admin-authorizer-package	edu.ucop.ers.access
<string> package name containing the authorization classes for decentralized security administration	
AccessControl.Authorization.admin-authorizer-class	UCOPAdminAuthorizerImpl
<string> name of the class that implements the IAdminAuthorizer interface to check authorization of resources, permissions and roles as well as returned filtered lists of permissions and roles based on users security admin level.	
CostSharing.DrillDownActive	False
Controls whether cost sharing numbers on an effort report will (true) or will not (false) provide an HTML link to an external cost sharing system	
CostSharing.DrillDownURL	http://www.ucop.edu/costsharing
<string> the URL that will be used to construct a link to the external cost sharing system. This link will be passed two parameters, the employee ID for the effort report and the project ID that is	

Configuration Element	Supplied Value
Description	
selected via the HTML link.	
Notification.MailServer	smtp.ucop.edu
<string> The hostname of an SMTP server that will accept mail from the application server.	
Notification.MailAgentHeader	Windows Eudora Version 6.1.2.0
<string> An RFC 2821 Mail User Agent acceptable to the SMTP server from above.	
Notification.AdministratorName	Effort Reporting System
<string> The text Name portion of the email address from which notifications are sent (From:)	
Notification.AdministratorEmail	ersadmin@ucop.edu
<string> The actual email Address of the email User from which notifications are sent. Insure that intended recipients do not consider this a SPAM from address.	
Notification.Reminder.RDUE	15
<number> This parameter defines the number of days prior to the certification due date that a reminder notification will be sent to the ER coordinator. A value of −1 will prevent reminder notifications of this type from being sent.	
Notification.Reminder.CDUE	10
<number> This parameter defines the number of days prior to the certification due date that a reminder notification will be sent to the PI. A value of −1 will prevent reminder notifications of this type from being sent.	
Notification.Reminder.ODPI	5
<number> This parameter defines the number of days after the certification due date that a reminder notification will be sent to the PI. A value of −1 will prevent reminder notifications of this type from being sent.	
Notification.LDCT	true
"true"|"false" When false, suppresses sending of line decertified notifications.	
Notification.RNLP	true
"true"|"false" When false, suppresses sending of notifications for new effort reports generated due to late pay processing	
Interface.ProjectAltID.SearchName	
This option is used by SponsoredProject Search Processor Interface for inserting/updating the <ProjectAltSearchName> search for a user.	
DatabaseConfiguration.PropFiles.PropFile	hibernate.base.properties
<string> Common Hibernate configuration properties used by all databases.	
DatabaseConfiguration.PropFiles.PropFile	hibernate.tst01.properties
<string> Specific Hibernate configuration properties used for a particular database and database management system. Note that the PropFiles items are read hierarchically.	
DatabaseConfiguration.MapFile	mappings.xml
<string> A list (in xml format) of Hibernate class mapping files (HBM xml). Shared by all DBMS and databases. A copy may be used to test a new mapping.	
Utility.fau-formatter.package	edu.ucop.ers.utility
<string> name of the package that contains the FAU formatter class	
Utility.fau-formatter.class	FauFormatterAdapter
<string> name of the class used to format the FAU in a way that makes sense to a particular campus.	
Branding.BrandDelimiter	/campus/brand/
<string> The string that will be replaced in Tiles definitions with the CampusBrand below.	
Branding.CampusBrand	/campus/generic/
<string> The replacement values to be used for branding the web pages with the local campus page elements.	

Configuration Element	Supplied Value
Description	
Constants.Classname	edu.ucop.ers.ERSConstants
<string> The name of the class that will be used to load constants into the application context during application server initialization for the web application.	
Externalized.Strings.FileName	ExternalizedStrings.xml
<string> Identifies an XML file that contains customizable string literals for various parts of the web application.	
Options.allow-save-out-of-balance	false
Web application option that controls whether an out of balance effort report can (true) or cannot (false) be saved to the database.	
Options.search-fund-tab-visible	true
Web application option that controls whether the funding source search tab is displayed.	
Options.search-alt-tab-visible	true
Web application option that controls whether the alternate projects search tab is displayed.	
Options.search-primary-filter3-visible	false
Web application option that controls whether the third filter field is visible on the primary project search tab.	
Options.search-alternate-filter3-visible	true
Web application option that controls whether the third filter field is visible on the alternate project search tab.	
Options.co-pi-only-my-projects	false
Web application option that controls whether "My Projects" includes reports when a co-PI is present on the sponsored project. False means that all projects will appear on both the PI and co-PI "My Projects" list. True means that when a co-PI is present on the project, the report will not appear on the "My Projects" list for the PI – only for the co-PI.	
Options.show-perm-icons	true
Web application option that controls whether the authorization icons (edit and view) appear on the report list. Turning these off will result in considerable performance gains when loading report lists.	
Options.primary-sort-col	orig
Field to sort on for above-the-line items on the report edit/view screens.*	
Options.primary-sort-direction	desc
Sorting direction of the above. Possible values are "desc", "asc"	
Options.load-btl-rpt-rsrcs	
Controls whether to load (get/generate) below-the-line report resources.	
Options.load-org-flat	false
When true, cause Department interface to load the OrganizationFlat table, which is the signal that the application should allow hierarchy-based permissions and searches. A "true" value for AccessControl.Authorization.decentralized-administration implies "true" here. Set this flag to "true" when you wish to load organization hierarchy data, but are not using Decentralized Security Administration.	
Options.req-url-scheme	http
Web/Application Server option that controls the scheme (i.e. protocol) to be used for application constructed Request URLs such as for Compliance Reports.	
Options.req-url-port	8080
Web/Application Server option that controls the port number to be used for the specified scheme in application constructed Request URLs such as for Compliance Reports.	
Options.alert-payroll-transfer	true
When true, the system will display a pop-up window with a warning to indicate that the a payroll cost transfer is required whenever the original payroll efforts do not match the adjusted efforts.	

Options. show-prelim-rvw-flag	true
When true, the system will display the "Under preliminary review" column on the report list, and allow a user to set this flag on the Edit tab of the report screen. Setting this option to false hides the report list column, the check box and flag on the Edit tab, and the flag itself on the View and Comment tabs.	

*Options.primary-sort-col possible values include:

Configuration Value	Corresponding Field
parPercent	Original Payroll % (String)
orig	Original Payroll % (Numeric)
adjustedPercent	Adjusted Payroll % (String)
importedCostShare	Original Cost Share % (String)
adjustedCostShare	Adjusted Cost Share % (String)
originalTotal	Original line total
contractNbr	Sponsored Project Contract Number
projectID	Sponsored Project ID
projectTitle	Sponsored Project Title
projectIdAlt	Sponsored Project ID Alt
projectNameAlt	Sponsored Project Name Alt
sponsorName	Sponsor Name
piName	Sponsored Project PI Name

With Release 9.2, the new XSD for ERSConfig.xml can help with these values if you use an XSD-aware XML editor.

Customization / Branding / Skinning

Overview

The ERS application was designed so that the look of the application could quickly and easily match the established web style of each deploying campus.

There are basically four things that can be customized, campus headers and footers, CSS, graphics, and HTML.

Headers and Footers

Campus specific headers and footers are defined in files separately from the main content of the ERS application. This allows you to completely replace those files with our own files and have almost no impact on the structure of the application.

CSS

CSS is used to define the look, and to a certain extent the layout, of ERS. With a little ingenuity you could even design your own tabs and buttons.

The cascading style sheets are structured so that you can override the default styles for the application in your own CSS file and never touch the default styles. This makes it easy to maintain your campus styles and see what new styles you need to update when the application is updated.

Graphics

Graphics used in ERS are stored in campus specific folders. This will allow you to maintain your own folder and see what changes are made to the default (generic) folder when the application is updated.

Note: since dimensions are declared for graphics in the HTML source, if you diverge from the default size of an image, you will need to modify the HTML source to correctly render the image. Specifying image dimensions allows the browser to more quickly render an HTML page, although this improved performance may be negligible with fast processors and broadband network connections.

HTML

The actual HTML structure of the application should not need to be modified in order to brand the application for your campus. However, if you need to change the structure of the pages, for the most part you can. It is not encouraged for reasons that we will discuss later, along with giving you some strategies to do so effectively and safely.

Getting Started: Setting up a Campus folder

In order to make ERS easy to customize and easy to upgrade later, we have engineered the application to use a special "campus" folder that will contain your headers and footers, your style overrides, and your graphic files.

This allows you to easily switch back and forth between your instance of the application and the generic instance.   This is useful when debugging your campus customization and is also very useful when upgrading ERS with a new version, since you can use the changes to the generic folder as a template for the changes that you need to make to your own folder.

Note: There isn't an override scheme for applications HTML files. HTML changes are done right in the source. For that reason, changing HTML is much more difficult to maintain when a new version of the application comes along.

We've also provided you with a sample campus folder called *ucop*. The ucop folder was used during development to test our customization scheme and may not be completely up-to-date. However, we felt that it does provide and interesting example of a campus customization.

Setting up a new campus folder

Inside the /*campus* directory, you will find two folders, *generic*, and *ucop*.
Duplicate the *generic* directory.
Rename the duplicate directory to some useful name such as the name or initials for your campus.
Once you set up the new directory, any changes to headers, footers, CSS or graphics should be done in this folder.

Modify settings

Once you set up the new directory, you need to make one small modification to *ERSConfig.xml* for the application to start using your campus folder.

In *ERSConfig.xml* replace the word *generic* in the following XML with the name that you gave to your campus directory. (Do not modify the *BrandDelimiter* line.)

```
<Branding>
        <BrandDelimiter>/campus/brand/</BrandDelimiter>
        <CampusBrand>/campus/generic/</CampusBrand>
</Branding>
```

Restart your server and you're ready to develop your own customized ERS application.

A Customization Strategy

We suggest that you customize your instance of ERS in the following order: headers and footers, css, graphics, and finally, and only if absolutely necessary, HTML.

Note: While it may be tempting to just make changes to the CSS and .jspf files outside of your campus folder, doing so will make it more difficult to keep your installation up-to-date.

Headers and Footers

Whether you're making minor modifications to the header and footer provided or you're completely replacing the provided header and footer with your own, the process is pretty much the same. Simply edit or replace the content of the current header and footer files.

Self-contained headers and footers or wrapper

Your headers and footers can be self-contained or can act as a *wrapper* for the rest of the application. You must decide whether you want to have your headers and footers to be self-contained or have them wrap the main content.

Self-contained headers

With self-contained headers and footers, the tags for each section are complete; the header or footer could stand on its own and be valid XHTML. The header section, footer section, and main section each have their own styles defining their width and position on the page. If done correctly, they'll all line up and look right.

```
<div id="header">
</div>

<div id="main">


</div>

<div id="footer">
</div>
```

Wrapping headers and footers

Wrapping headers and footers give you slightly more control over the layout of the page by using the header section and the footer section to *wrap* the main section that contains the application.
The header initiates the wrapper div and the footer closes it.
Widths and positioning on the page is controlled by the wrapper – the enclosed elements are contained and constrained by the wrapper.

Note: According the Struts documentation, you're supposed to close all tags within a given file, so we're officially cheating when we do this. But if done correctly it doesn't break anything and it buys us a lot of control.

Liquid vs. Fixed Width Layout

While not strictly a header and footer issue, this is a good place to talk about liquid verses fixed-width layout. Liquid layouts resize as the user resizes the browser. Fixed-width layout, as the name implies, have a fixed width. Both styles have their advantages and ERS supports both. Which one you choose will depend mostly on preference or precedence.

```
<div id="wrapper">
    <div id="header">
    </div>

    <div id="main">


    </div>

    <div id="footer">
    </div>
</div><!-- end of wrapper -->
```

The generic layout is fixed width. To change your layout to liquid, you'll want to change the width attribute for the wrapper element to a percentage instead of a pixel width.

Exercise

If you're not sure what we're talking about here, just try this simple exercise:
1. Start up your server and open your instance of the ERS in a browser.

2. Sign-in to the Report List view.
3. In your editor, open the file styles.css in your campus folder, and change the line width: 740px; to width: 97%;, Save and refresh the page.
4. Now, try resizing the browser, and notice that the content of the page resizes with the browser.

Customizing the default header and footer files

You can modify the header and footer files pretty much anyway you like as long as all of your changes are self-contained to the header and footer files.

If you're doing a very light customization, you may want to just edit the header and footer files that are provided; change the colors, add your own logo, etc. If you would like to add links to the header file to other online campus resources, there are already some commented out links in the default header file.

Styles for the header and footer files

The styles for the header and footer provided can be found in the **styles.css** file in your *campus* folder. The styles in this file define only the header and footer and the wrapper div for the application.

To link to a different style sheet, such as a style sheet already in use at your campus, modify the stylesheet reference link in *campus/<yourcampus>/head_tags.jspf.*

```
<link href='<html:rewrite page="/${campusBrand}styles/campus_styles.css" />'
rel="stylesheet" type="text/css" media="all" />
```

CSS

This guide assumes that you already know a fair amount about CSS; how it works and why it good. If you need a refresher course on CSS, you might start with:
http://www.mako4css.com/Basics.htm, or http://www.cssbasics.com/.

ERS contains some very complex styling. We're not claiming that it's the "be all and end all" of CSS design—there were a lot of changes and modifications that crept into the styles that could have been more artfully done. However, we've attempted to follow a best practices approach to CSS while also making it easy for each campus to have a high-level of control when customizing the look of ERS.

The tools mentioned above, especially the Firefox *View Rendered Source* extension, will help you identify how the styles are applied to each element.

Note: As you make changes to the CSS, make sure to check your changes in the live application with both IE and Firefox.

CSS Structure: How to override our styles with your styles

In CSS, styles cascade from enclosing element to enclosed element. However, you can also think of styles as cascading through the various CSS documents and style declarations that are applied to the page. Just as you can declare a style to a parent element and then override that style with a style attached to a child element, you can also declare a style for an element in one style sheet and then override that style in another style sheet, on the actual page or element.

This is the hierarchy of the css files attached to every page in ERS.

`<your campus CSS file for your chrome>`

`ers_styles.css`   *:the bulk of the ERS CSS is here*

`campus_styles.css`   *:but you'll modify it here*

Editing CSS

Approach modifying the styles from the outside in, starting with enclosing elements (parents) and then modifying the styles attached to child elements, starting with the body and working your way inward to the smaller more localized elements on the page.

Another reason to start with enclosing objects is that an element on a page may have the same class or id as a similar element on a different page. However, because of styles applied to the enclosing element the styles applied any given element may look and behave quite differently from page to page.

The best way to learn is go in and make changes. Go ahead, be bold! Make a change and then quickly take a stroll through the app to see if your change has had the effect that you expected. You can always rollback to a previously saved version.

Tricks

Image references in the CSS files

In order to have image references in the CSS reference image files inside your campus folder, all image references are declared in the campus override CSS file in your campus folder, and not in the main CSS file. There is a comment calling this out wherever it happens.

IE Hacks

Not all CSS standards are implemented exactly the same way from browser to browser. Luckily, with most modern browsers, CSS Level 2 for XHTML documents is implemented pretty consistently. However, there are a couple of styles that are modified to support quirks in Microsoft's Internet Explorer.

There are a number of techniques to apply different styles for different browsers. Each technique has its advantages and disadvantages, and its supporters and detractors debating *ad nauseam* in the blogsphere.   We have chosen to hide styles from IE using a child selector, ">".

Example

```
/* Text Buttons */
span.textbutton {
        position: relative;
        top: -2px;
        border: black 1px solid;
        padding: 0;
        margin: 0;
}

/* selector hack to hide from IE */
html>body span.textbutton {
        padding: 1px 0px 2px;
        top: 0px;
        font-size: 12px;
}
```

The child selector is part of the [W3 CSS2 specification](#). Conveniently, IE does not support the child selector and thusly ignores any CSS statements defined in this way.

There is a danger that some day Microsoft will release a version of IE that will support this part of the W3 specification. However, it is hoped that they will clean up their CSS bugs at the same time.

A few other CSS files

There are a couple of other CSS files that you will need to know about.

Printing styles are defined in */styles/ers_print.css.* You can also override printing styles in your campus_styles.css file using the `@media print` declaration.

The styles for the tooltip style popups are declared in */styles/nicetitles.css*.

Graphics

Your graphics in your folder

All the graphics files for ERS are in the images folder inside the campus folder that you set up. You can easily change the look of your application by replacing or modifying these images.

Sizes

However, don't change the size of a particular graphic element unless you have to. Most of the images in the source are specified with height and width attributes to increase performance. If you change the size of an image, you will need to change it's size in the source.

HTML

Changing the HTML really should be done only in the last resort. Changing the HTML in the source files must be done with great care since it may change the behavior of the page. And, changes to source will need to be resolved each time the application is updated.

Hide don't delete fields

When changing HTML on pages that use forms, do not delete any fields, as these fields are referenced by the application both in the Java and, in many cases, by the JavaScripts on the page. Missing fields will result in errors. To remove a field, change the type of the field to hidden. In most cases this is done by changing the struts tag from `html:text` to `html:hidden`. You may also need to attach a default value to the hidden field as well in order to pass the right value to the application.

Comment, Comment, Comment

Don't forget to comment your changes to the source. Use a standard format and a key word in your comments so you can easily search for them when doing updates.

Example:
```
<%-- CHANGE 2/17/06 Eli: Changed case of field value to upper --%>
```

Test

I guess it goes without saying that any change that you make, you should test thoroughly. Make sure to test in all the browsers that you plan to support.

Campus Folder Management

The customized campus folder must be stored within the application directory structure so that the customized UI elements can be served by the application server or web server. The method of deploying the campus folder varies depending on the application server product in use. In general, it is important to note that the installation of new ERS releases in the form of WAR files will recreate the entire application directory structure, which will erase the campus folder. Campuses should develop a protocol for backing up this folder and restoring it after installation of a new ERS WAR file. The folder may be manually recopied after WAR file deployment or the folder could be placed in a standalone JAR file and deployed using the application server deployment tool after the ERS WAR file is redeployed.

It is also possible that a release of the Base ERS system could add new elements to the campus folder, or change existing elements. These changes will be noted in the release letter but will require analysis and comparison of the skin elements in the campus folder, to insure that new and changed elements are applied to a custom campus skin.

Authentication

Authentication in ERS is the process of making the decision whether a user can access the web application.

ERS supports two modes of authentication, Internal and External. As released, ERS is configured to use internal authentication.

<u>Internal Authentication</u>

Internal authentication uses an application login page, and compares the entered user ID and password against an application table, ERSUser. The user interface supports user-requested password resets via email, and the System Administration module also allows a password to be reset by a central administrator.

After the login page validates the user's password, it constructs an object known as a UserContext. This object contains information about the logged in user, such as their user ID, full name, email address and employee ID among other items. This object is central to the operation of the web application and is required to be present for all application users.

<u>External Authentication</u>

An external authentication mechanism may be used for ERS. The login process would be managed by another application which would have responsibility for user ID and password verification.

Using an external authentication mechanism requires that the authenticator fulfill several responsibilities prior to attempting to access ERS on behalf of their user. These are:

1. implement a campus-specific UserContext builder class

    The custom UserContextBuilder class must retrieve user attributes and populate the appropriate UserContext fields. This class is invoked from the ERS authentication servlet filter. The builder class must also make the determination whether the User is authorized to access ERS.

2. provide an "error URL" to which user will be redirected if authentication fails, and a timeout URL, to which the user will be redirected if their session times out.

    If the UserContext is null or otherwise invalid, ERS will not allow access to the application and will redirect to the provided error URL.

3. provide key name of the request parameter that contains an authorization string.

    The custom UserContextBuilder will likely need to get one or more tokens or strings that have been built by the campus authentication process to represent the user's identity. The builder class would perform validation of these strings and translation of these strings into user attributes that can be used by the application.

ERS expects that the authorization tokens or strings can be accessed through the HTTP request object. This approach supports the use of a form field or query string as the mechanism for passing the authorization strings into the application.

An ERS configuration parameter identifies the name of the HTTP request object fields that contains these values.   The ERS authentication servlet filter retrieves these values from the request object and passes them to the custom UserContextBuilder class when it is invoked.

To implement a custom UserContext builder, the UCOPUserContextBuilder can be used as a model.

Authorization

Authorization in ERS is the process of making decisions about a user performing an action on a specific application resource, and in some cases, obtaining email and full name information about the user.

The ERS authorization scheme consists of several interfaces that capture the types of decisions the system must make.

The interfaces are:

> IAuthorizer
> IMyProjectsAuthorizer
> IMyReportsAuthorizer
> IMyRolesChecker
> IEmployeeInfo
> IAdminAuthorizer

ERS provides internal implementations of all these interfaces. Campuses can choose to rely entirely upon the ERS internal system, they can completely rely on an external system, or they can choose to use a mix of both internal and external implementations.

There is support for implicit permissions, and exclusion lists from the implicit permissions. The implicit permissions are termed "My Projects" and "My Reports".

The "My Projects" implicit permission means a principal investigator has permission to view, edit, certify and view payroll data for any report that contains at least one of his/her projects. The "My Report" permission means any user with a report in the system can view, edit, certify, and view payroll data for their own reports.

Users who would otherwise have these permissions can be explicitly excluded from them. Furthermore, users can be excluded from certifying their own reports, while still retaining the other permissions. Internally this is done by maintain lists that are managed through the System Administration module.

Interfaces

IAuthorizer

The core interface is IAuthorizer. A campus that implements this interface can support all authorization requirements of the application.

Every ERS application instance must provide an implementation of this interface. The implementing class must be specified in ERSConfig.xml with the following items:

AccessControl.Authorization.authorization-package
AccessControl.Authorization.authorization-class

ERS provides a concrete implementation of this interface for internal authorization:

edu.ucop.ers.access.UCOPAuthorizerImpl

IMyProjectsAuthorizer

To enforce the "My Projects" business rule (i.e. any principal investigator can perform any intent on any report that contains at least one of his/her projects unless explicitly excluded otherwise) when performing an authorization check, ERS makes a call on this interface.

Every ERS installation will have an implementing class, either a custom-build class that connects to the campus-specific authorization application that can return a decision based on the provided parameters, or one of the two ERS-provided implementations. The implementing class must be specified in ERSConfig.xml with the following items:

AccessControl.Authorization.my-project-authorizer-package
AccessControl.Authorization.my-project-authorizer-class

Since this permission can be explicitly denied to specific users, the interface call relies on another interface, IMyRolesChecker, to enable that business rule.

ERS provides a trivial implementation of this interface,

edu.ucop.ers.access.SimpleMyProjectsAuthorizerAdapter

for external implementations that wish to use IAuthorizer to handle the self-permission role.

ERS also provide the class

edu.ucop.ers.access.UCOPMyProjectAuthorizerImpl

which is a full implementation of the My Projects interface, supported internally by a persisted list of excluded users. This list is populated via the System Administration module.

IMyReportAuthorizer

To enforce the "My Reports" business rule (i.e. any ERS user with a generated report can perform any intent on that report unless explicitly excluded otherwise) when performing an authorization check, ERS makes a call on this interface.

Every ERS installation will have an implementing class, either a custom-built class that connects to the campus-specific authorization application that can return a decision based on the provided parameters, or one of the two ERS-provided implementations. The implementing class must be specified in ERSConfig.xml with the following items:

    AccessControl.Authorization.my-reports-authorizer-package
    AccessControl.Authorization.my-reports-authorizer-class

Since this permission can be explicitly denied to specific users, the interface call relies on another interface, IMyRolesChecker, to enable that business rule.

ERS provides a trivial implementation of this interface,

    edu.ucop.ers.access.SimpleMyReportsAuthorizerAdapter

for external implementations that wish to bypass the implicit my reports permission and pass through to the IAuthorizer call.

ERS also provides the class

    edu.ucop.ers.access.UCOPMyReportAuthorizerImpl

which is a full implementation of the my report interface, supported internally by a persisted list of excluded users.

IMyRolesChecker

This interface works in conjunction with the IMyProjectsAuthorizer and IMyReportsAuthorizer interfaces to enforce the self-permission business rules described above.

As in the My Reports/My Projects interfaces, ERS provides trivial adapter implementations that allow campus customizations to just use the authorization call in the IAuthorizer.

The implementing class must be specified in ERSConfig.xml with the following items:

    AccessControl.Authorization.my-roles-checker-package
    AccessControl.Authorization.my-roles-checker-class

IEmployeeInfo

This is a simple struct-like type that associates an employee's list of email addresses with a formatted version of the employee's name. Since it is only ever used in the context of a return value from one of two method calls defined in IAuthorizer, there is no need for it to be described by the configuration scheme.

IAdminAuthorizer

This interface defines methods used to check authorization to resources, permissions and roles when the decentralized security administration feature is enabled. The interface also defines methods to return filtered lists of permissions and roles based on the organizational association of a security admin.

Levels of Customization

There is a true/false flag configuration item in ERSConfig.xml that indicates whether the application should be run under internal authentication or external authentication:

AccessControl.Authentication.internal-authentication

External

IAuthorizer/IEmployeeInfo

If a campus's authorization system can provide support for the all authorization decisions the system requires, including the "My Projects" and "My Reports" business rules, then its customization will consist solely of implementing the IAuthorizer and IEmployeeInfo interfaces. As part of this customization, the campus will specify in ERSConfig.xml that the system should use the trivial bypass implementations of the IMyReportsAuthorizer, IMyProjectsAuthorizer, and IMyRolesChecker interfaces:

Interface	Trivial Adapter
edu.ucop.ers.access.IMyProjectsAuthorizer	edu.ucop.ers.access.SimpleMyProjectsAuthorizerAdapter
edu.ucop.ers.access.IMyReportsAuthorizer	edu.ucop.ers.access.SimpleMyReportsAuthorizerAdapter
edu.ucop.ers.access.IMyRolesChecker	edu.ucop.ers.access.SimpleMyRolesCheckerAdapter

This approach will require the implementation of the IEmployeeInfo interface, but this is expected to be trivial.

Full External Implementation

A campus can also choose to use their external system to implement the IMyReportsAuthorizer, IMyProjectsAuthorizer, and IMyRolesChecker interfaces if it makes sense when analyzing the resources of the campus's own authentication system.

Mixed Implementation

If a campus's authorization system cannot provide support for the "My Projects" and/or "My Reports" business rules, then it can customize the authorization by implementing IAuthorizer while relying on the internal implementations of the self-role business rule(s) as needed. This approach then requires the campus to use the administration user interface to populate the exclusion tables with the user that should be excluded from the self-roles as needed.

Interface	Trivial Adapter
edu.ucop.ers.access.IMyProjectsAuthorizer	edu.ucop.ers.access.UCOPMyProjectsAuthorizerImpl
edu.ucop.ers.access.IMyReportsAuthorizer	edu.ucop.ers.access.UCOPMyReportsAuthorizerImpl

Internal Implementation

If a campus lacks an authorization system or it cannot support the decisions represented by the IAuthorizer interface, then it can rely on the internal ERS authorization system. This requires using the administration user interface to assign roles, permissions, etc to users, and to populate the exclusion tables.

Decentralized Security Administration

There is a true/false flag configuration item in ERSConfig.xml that indicates whether the ERS will use decentralized security administration features. This feature is dependent on use of the internal authorization mechanism provided by ERS.

AccessControl.Authorization.decentralized-administration

Decentralized security administration allows creation of departmental security administrators who have the ability to create permissions and roles but only for the organizational units which they are authorized to administer.

This feature is documented in further detail in the *ERS Installation and Operations Guide*.

Account Number (FAU) Display

ERS provides a feature to allow customization of the FAU presentation when this data element is displayed in the web application. Although the FAU has a common definition of 30 characters, across installations the values in this field can vary in their format and interpretation.

ERS provides an interface called IFauFormatter for the purpose of formatting the FAU for display. A trivial implementation of this interface is provided in class FauFormatterAdapter. This implementation may be modified to meet installation-specific requirements.

If a customized version of the FauFormatterAdapter class is used, it must be identified in the ERS configuration file, ERSConfig.XML

ProjectAltID Search Name

This tag

```
Interface.ProjectAltID.SearchName
```

in ERSConfig.xml is used by SponsoredProject Search Processor Interface for inserting/updating the <projectAltSearchName> search specified by this option value for a user.

PI Highlighting

Starting with R9.2, ERS highlights reports belonging to principal investigators, both on the report list, and on the individual report screens. The text used as a tag on the individual report screens is externalized, see below for more information. The highlight color can be set by overriding the background-color element of the following styles in ers_styles.css:

```css
.piHighlighting{
        background-color: #FFFF33;
        font-weight: bold;
        padding-left: 10px;
        padding-top: 2px;
        padding-bottom: 2px;
}

td.piCellHighlighting{
        background-color: #FFFF33;
        font-weight: bold;
}
```

See the above section on Campus customization for more information about overriding CSS styles.

Relabeling Fields and Customizing Message Text

ExternalizedStrings.XML
Several field labels in the web application are customizable to allow installations to choose different wording to refer to various elements of the effort report. This set of labels can be customized by editing the configuration file ExternalizedStrings.XML.

The supported customizable field labels and their default values are:

Field Name	Default Value	Description
ProjectName	Sponsored Project Name	Primary project name displayed on each sponsored project line of the effort report-CURRENTLY UNUSED
ProjectNameAlt	Grant Title	Alternate project name displayed on the project line "flyover"
ProjectId	Sponsored Project ID	Primary project identifier displayed on each sponsored project line of the effort report-CURRENTLY UNUSED
ProjectIdAlt	Alternate Project ID	Alternate project identifier displayed on the project line "flyover"
SponsoredTitleLabel	Sponsored Projects	Main caption for sponsored projects section of the effort report
SponsoredTotalLabel	Total Sponsored Projects	Footer caption for sponsored projects section of the effort report
SponsoredNoteLabel	requiring certification	Secondary footer caption for sponsored projects section of the effort report
OtherTotalLabel	Total Other Effort	Main caption for total line of the "Other" section of the effort report
OtherNoteLabel	not requiring certification	Secondary caption for total line of "Other" section of the effort report
GrandTotalLabel	Grand Total	Main caption for grand total line of the effort report
NegCSLabel	Cost Sharing Offset Against Other Sponsored Projects	Main caption for the negative cost sharing report option
ERcol1Head1	Original	Effort report column 1 heading line 1
ERcol1Head2	Payroll %	Effort report column 1 heading line 2
ERcol2Head1	Adjusted	Effort report column 2 heading line 1
ERcol2Head2	Payroll %	Effort report column 2 heading line 2
ERcol3Head1	Original	Effort report column 3 heading line 1
ERcol3Head2	Cost Sharing %	Effort report column 3 heading line 2
ERcol4Head1	Adjusted	Effort report column 4 heading line 1
ERcol4Head2	Cost Sharing %	Effort report column 4 heading line 2
ERcol5Head1	Original	Effort report column 5 heading line 1
ERcol5Head2	Total %	Effort report column 5 heading line 2
ERcol6Head1	Adjusted	Effort report column 6 heading line 1
ERcol6Head2	Total %	Effort report column 6 heading line 2
OriginalLabel	Original Payroll %	Effort report drill-down label
PeriodLabel	Period	Effort report list Period column heading
EmployeeLabel	Employee	Effort report list Employee column heading
DepartmentLabel	Department	Effort report list Department column heading
LastModifiedLabel	Last Modified	Effort report list Last Modified column heading
StatusLabel	Status	Effort report list Status column heading

UnderPrelimReviewLabel	Under Prelim Review	Effort report list Under Prelim Review column heading
PeopleTabLabel	People	Label for the Person search tab
DeptsTabLabel	Departments	Label for the Departments search tab
DeptsTabDropDownLabel	department	Label for the department drop down on the Depts search tab
DeptsTabDropDownHomeDept	Home Department	Depts search tab drop down home department choice; also used to label controls and columns in Compliance Reporting
DeptsTabDropDownAcctDept	Account Department	Depts search tab drop down account department choice; also used to label controls and columns in Compliance Reporting
DeptsTabDropDownFundDept	Fund Department	Depts search tab drop down fund department choice; also used to label controls and columns in Compliance Reporting
DeptsTabFilterLabel	Department	Depts search tab filter control label
PrimaryTabLabel	Funds	Label for the primary (first) projects search tab
PrimaryFilter1Label	Project ID	Label for the first filter field on the primary projects tab
PrimaryFilter2Label	Fund Name	Label for the second filter field on the primary projects tab
PrimaryFilter3Label	Sponsor Award #	Label for the third filter field on the primary projects tab
AlternateTabLabel	Projects	Label for the alternate (second) projects search tab
AlternateFilter1Label	Project #	Label for the first filter field on the alternate projects tab
AlternateFilter2Label	Award Title	Label for the second filter field on the alternate projects tab
AlternateFilter3Label	Sponsor Award #	Label for the third filter field on the alternate projects tab
FundTabLabel	FAUs	Label for the Funding Sources search tab
FundTabFilter1Label	FAU	Label for the first filter field on the Funding Sources search tab
FundTabFilter2Label	Fund Title	Label for the second filter field on the Funding Sources search tab
DrilldownFAULabel	FAU	Label for FAU column on payroll and FAU drilldown screens
PiIndicator	>>PI<<	Tag that appears on individual reports of PIs

Message Text

Currently, the email message that ERS generates when a user sends a shared search to a colleague can be configured as follows:

Field Name	Default Value	Description
ShareSearchSubject	Sending ERS Shared Search	Subject line of the share search email message
ShareSearchNameText	<NAME> has sent you a shared search from the Effort Reporting System.	Sentence indicating who is sending the message
ShareSearchPreLink	To add this search to your list of Saved Searches, click this link:	Arbitrary text that precedes the share search URL link
ShareSearchPostLink	If clicking this link does not work, copy the link and paste it into the address field in your browser.	Arbitrary text that follows the share search URL link

These four fields also support paragraph and break marks. "$P" will be substituted with two newlines, and "$BR" will be substituted with one newline.

Notification Message Templates

You can customize notification message text by either editing the ERS_INSTALL_BASE sql script before installation, or by editing EVENT_MSG column in the ERSNotificationText table. Each notification type has a message template and a set of parameter(s). When ERS dispatches notifications, the placeholders in the templates are replace by the parameter(s) to create a customized notification message. The default text for each type of notification, and its parameter(s) includes:

Notification Code	Message	Parameter(s)
RNEW	Effort reports for ${periodName} are now available for review.${campusURL}. Certifications are due on ${dueDate}. Questions? Please contact ${campusEmailContact}.'.	Reporting period name, application URL (as specified in ERSConfig.xml), reporting period due date. Postion of parameters is arbitrary.
RDUE	Effort reports for # were issued on #. Please monitor progress and insure that certification is complete by the due date of #.	Reporting period name; report issue date; report due date
IERR	An error occurred during interface processing for # on # that requires attention. Please use ERS System Administration to view the error log and take corrective action.	Interface job code plus the input filename; date job was run
CDUE	Effort report(s) in period ${periodName} (issued on ${issueDate} and due on ${dueDate}) contain effort on your project${plural}:${projList}Please insure that certification is complete as soon as possible.	Reporting period name, report issue date, report due date, and a list of the receipients projects (project ID + project name).
REIS	Effort report for # (#) in # was reissued due to payroll adjustments. Please review and recertify this report as soon as possible.	Report owner name; report owner EID; reporting period name
LDCT	Sponsored Project # on effort report for # in # was decertified by # on #.	Project name; report owner name; report owner EID; reporting period name; decertifier user ID; decertification date
ODPI	Effort report(s) in ${periodName} (issued on ${issueDate} and due ${dueDate}) contain effort on your project${plural}: ${projList} Please insure that certification is complete as soon as possible.	Reporting period name; report issue date; report due date; "s" if more than one project; the list of project(s) on reports for this PI in this period
RNLP	Effort report for # (#) in # was newly issued due to payroll adjustments. Please review and certify this report as soon as possible	Report owner name; report owner EID; reporting period name
INTF	Interface processing completed for file # on # with a completion code of #. If the completion code is non-zero, an error has occurred during processing of the file. Please use ERS System Administration to view the error log and take appropriate corrective action. #NL#NL Number of records processed #.	File name; current system date; job completion code; number of records processed

LPAY	Late Pay processing completed on #. #NL#NLProcessed # late pay records. #NLGenerated # new reports. #NLRebuilt # reports. #NLReissued # reports. #NLReconciled # Certified/Adjustment Required reports to Certified status. #NLRetained # certified reports as Certified.	Current system date; number of late pay records; number of new reports generated; number of reports rebuilt; number of reports reissued; number of reports reconciled; number of reports retained as certified

Note on Notification Application Link

All notifications include a link to the application. The value for the link is set in ERSConfig.xml, in the `Configuration.GlobalItems.ApplicationBaseURL` item. The text that it is embedded in is not the ERSNotificationText table. It is found in the `WEB-INF/classes/edu/ucop/ers/struts/ApplicationResource.properties` file, in two properties:

```
# Notification resources
n.url.msg1 = Click
n.url.msg2 = to access the Effort Reporting System
```

Parameter substitution rules

For all event types except ODPI and RNEW, the order of the parameters is fixed. Each '#' will be substituted with the value noted above, always in the order shown. The text surrounding the hash marks is arbitrary, with a max length of 200 characters. For ODPI and RNEW, ERS supports named parameters, allowing more flexibility in composing the template text. The parameter values will be substituted by matching the named parameters, regardless of the order they are found in the template.

Empty Values

Empty values in the externalized strings xml file default to the above values. Configured values and default values can be mixed and matched as needed.

Loading the Configuration

Changes to this file requires a restart of the application server to reload the new file contents or use of the "Reload Configuration" task on the System Administration menu. NB: this only works in non-clustered application server environments. A future enhancement to ERS will address this problem.

Any of the customizable field labels may be configured to display blank space by using the special keyword **$EMPTY$** as the value for the field label.

Appendix A of this document provides a graphical representation of where these customizable labels appear in the ERS web application pages.

ApplicationResources.Properties

All strings in the application, including field labels, message texts, prompts, etc., are defined in the file **ApplicationResources.Properties** which is located in the source package **edu.ucop.ers.struts**. Further *modification* of the application can be accomplished by editing this file with a normal text editor.

Note: Changes to ApplicationResources.Properties are considered *modifications* and will not be preserved across releases. Campuses modifying this file will be responsible for merging local changes after installation of a new release.

Report List Permission Icons

The default page design for the report list includes two icons that are meant to indicate whether a user has edit and/or view authorization for each listed report. The icons are dimmed out when the user lacks authorization for the corresponding function for that report. This improves the user experience.

The tradeoff is that for each report on the list, the web app must make multiple calls to the DBMS to determine the user's authorization status for that report. This results in a considerable performance penalty. Starting with Release 8, displaying these icons is optional. Set **Options.show-perm-icons** to **false** to turn them off.

With the permission icons are turned off, when the user tries to navigate to a specific report, the application attempts to take the user to the Edit tab (assuming the report is not certified), and failing that, then attempts to take the user to the View tab. If the user lacks both authorizations for that particular report, the application displays an error message at the top of the report list.

Icons Turned On:

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Icons Turned Off:

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Icons Turned Off: Not Authorized Message

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Below-the-Line Report Resources

There is a true/false flag configuration item in ERSConfig.xml that indicates whether to load (i.e. get/generate) report resources for below-the-line earnings while generating resources for an effort report.

> Options.load-btl-rpt-rsrcs

When this flag is true, besides loading above-the-line report resources, ERS includes report resources for Other (O) and Non-Sponsored (N) category earnings excluding Negative Cost Sharing (Z) and Unrecognized (I) category earnings for an effort report. When this option is disabled, ERS loads just only above-the-line report resources. Loading these below-the-line resources will allow users with permissions based on sponsored projects, funding sources, fund orgs, and account orgs to access report that only have those resources in the two below-the-line categories.

If this configuration item is omitted from ERSConfig.xml, the application will default to the false value (below-the-line resources will not be loaded).

Loading Organization Hierarchy

If you are not using decentralized security administration, but would still like to make your organization hierarchy available to ERS, set `Options.load-org-flat` to `true`. Loading this data will enable the creation of organization hierarchy-based searches and permissions. For searches, picking one or more Account or Fund Orgs on the Define Search Department tab, and checking the "Perform Hierarchical Search" box will cause the search to find reports linked to orgs in subtree(s) of the selected org(s). In the security context, a permission linked to Account or Fund org resource types can be made hierarchical, which will authorize access reports linked to orgs in the subtree(s) of the orgs linked to the permission. Use the new "FS Acct Dept Hierarchical" and "FS Fund Dept Hierarchical" resource types to create these types of hierarchical permissions. When hierarchical data is available, the online help contains information on both these features.

Using this option will require that the department interface files that you load into ERS contain valid hierarchical data. See the Interface Specification for more information.

Request URL Scheme

Options.req-url-scheme

This is a web/application server option that controls the scheme (i.e. protocol) to be used for application constructed request URLs such as for compliance reports. The possible values for the scheme include http/https/ftp etc. If this configuration item is omitted from ERSConfig.xml, the application will default to the "http" value.

Request URL Port

Options.req-url-port

This is a web/application server option that controls the port number to be used for the specified scheme in the application constructed request URLs such as for compliance reports. If this configuration item is omitted from ERSConfig.xml, the application will default to the "8080" value.

Document Change Log

Date	Build #	Changes
2/15/06	Release 0	Initial draft for technical review
2/17/06	Release 1	Finalization of the customization section
4/10/06	Release 2	Document new page label customization
5/31/06	Release 4	(document not revised for release 3) Document new option for decentralized security administration
6/16/06	Release 5	New parameter to control "My Projects" behavior
8/14/06	Release 6	Added notation about backup of "campus "folder
11/13/06	Release 7	Enhanced the table describing customizable field labels
		Added appendix A to document placement of labels
		Added appendix B to document FS interface field placement
4/13/07	Release 8	Added show-perm-icons entry to configuration options table
		Added section on report list permissions icons
7/27/07	Release 8.1	Added signout-redirect-url item to ERSConfig.xml contents
10/16/07	Release 8.2	Added Notification.Reminder.ODPI item to ERSConfig.xml contents
12/14/07	Release 8.3	Added load-btl-rpt-rsrcs entry to configuration options table
2/20/08	Release 9.0	Added above-the-line sorting options to ERSConfig.xml description; added Options.load-org-flat to same; added section on use of the load-org-flat option.
3/21/08		Added Options.req-url-scheme and Options.req-url-port options to ERSConfig.xml description; Added sections for each of the req-url-scheme and req-url-port options.
3/21/08		Added Interface.ProjectAltID.SearchName item to ERSConfig.xml description; Added section for ProjectAltID Search Name item.
6/20/08	Release 9.1	Added MessageText section; added NotificationText message and parameter information; added note about online configuration reloading and clustered environments
9/19/08	Release 9.2	Added new externalized strings, new LDCT option in ERSConfig.xml; new RNEW parameters; added new externalized string to Appendix A images as appropriate
12/22/08	Release 9.3	Added new RNLP option in ERSConfig.xml
03/26/09	Release 10.0	Added timeout-redirect-url option in ERSConfig.xml
06/30/09	Release 10.1	Added alert.payroll.transfer and show-prelim-rvw-flag option in ERSConfig.xml; Added new INTF and LPAY notification message templates; Modified ODPI and CDUE notification message

		templates; Added note on notification footer text and ApplicationResources.properties
08/27/09	Release 10.2	Externalized headings on the Effort Report List and changed the heading "For" to "Employee". Also changed the heading "HomeDepartName" to "Department" for consistency when the list is exported or sent.

Changes to this document from the prior release are indicated by revision bars in the left margin. Minor editorial corrections have not been flagged with revision marks.

Appendix A – Customizable Label Placement in Web Application
Department Search Tab

Payroll Drilldown

FAU Drilldown

Primary Project Search Tab

Alternate Project Search Tab

Effort Report Page

Effort Report Page (con't.)

Effort Report List

Appendix B – FS Interface Field Placement in Web Application

Primary Project Search Tab

Alternate Project Search Tab

Effort Report Page

Compliance Report Generation Screen