

University of California ~ Effort Reporting System Installation and Operations Guide

Release 10.5
February 23, 2011

Application Technology Services
Information Resources & Communications
Office of the President

Contents

Planning for Installation	1
Pre-Installation	5
Front-End Requirements.....	8
Deploy WAR.....	9
Database Setup	10
Initial Customization.....	12
Configure Hibernate.....	13
Installation Checklist	17
Batch Installation Verification.....	18
Online Installation Verification.....	20
Structure Version Check	21
Operations Overview	25
Batch Interface File Loading.....	25
Batch Report Period Processing.....	30
Additional Batch Programs	31
System Administration	32
Post-Installation Tasks.....	42
Application Source Documentation	43
Application Database Tables.....	44
Pre-Built Permissions	46

Planning for Installation

The Effort Reporting System relies on data that comes from other systems. ERS takes six input or “interface” files and stores this data. Initially, ERS has no data of its own until batch processing is run to create effort reports. At that point, ERS uses the data loaded via the interface files to create new data in the form of generated effort reports.

Planning for installation requires that the ERS input data requirements be analyzed and the appropriate data sources be identified from existing external systems. In most cases, additional programming will be required to extract the required data from these external systems and format this data for loading into ERS.

The format and content of the ERS input files is documented in the latest version of the *ERS Interface Specification*.

ERS accepts seven interface files:

PPS Interface (required interface file):

Input from the Base Payroll/Personnel System (PPS) is already formatted for use in ERS by a Base PPS program. The PPS sequential PAR file is read and selected data elements are extracted and formatted for loading into ERS.

FS Interface (required interface file):

The financial systems (FS) interface file contains data from the chart of accounts and/or contracts and grants system. It identifies all account/fund combinations that are used for payroll charges, as well as sponsored projects (grants) that are subject to effort reporting.

DEPT Interface (required interface file):

The departments (DEPT) interface file contains the organizational hierarchy which is used by ERS to associate account/fund combinations and sponsored projects with their responsible department or organizational unit. If using decentralized security administration, enable the option for this feature first, before loading the DEPT interface file for the first time. This will insure that the organization structure trees are built correctly for use with the decentralized security administration feature.

CS Interface (optional interface file):

The cost sharing (CS) interface file is optional. If present, it should contain the cost sharing commitments for each employee on each sponsored project. These commitments are presented on the effort report as default values. If this interface file is not loaded, employees can still enter cost sharing percents on the effort report manually.

USER Interface (optional interface file):

The user (USER) interface file is optional. If present, it should contain all users that will be allowed to access ERS and associates these users with their employee ID. Any user found on this file will be made *active*, indicating they have permission to access ERS. If a user is removed from this file, ERS will detect this and will mark the user *inactive*.

Pilot Period Interface

This interface is optional; use it to designate a pilot reporting period. Reports will only be generated for the employees who are included in the file. Pilot period status can be assigned to existing reporting periods – on a going forward basis, late pay will only apply to pilot participants; all other late pay earnings will be skipped. Running this job will have no affect on existing reports.

WOS Employee Interface (optional interface file):

The Without Salary employee (WOS) interface file is optional. If present, it should contain all employees that do not have a PAR record in the PPS interface file. Any employee found on this file will able to access ESR system for the purpose of viewing and certifying effort reports for others.

ORGDEPT Interface (optional interface file):

The ORGDEPT interface file is optional and is used in conjunction with the decentralized security administration feature. The file associates organization unit codes typically found in financial systems, with Home Department codes which are used by the Payroll/Personnel (PPS) system. In order for departmental security administrators to be able to create permissions based on PPS Home Department, the system must be loaded with the data that associates the Home Department code with the organization unit codes that are used to control ERS authorization.

Project ID Alt/User Interface

This interface allows the automatic generation of searches for supplied User IDs based on the value in the PROJECT_ID_ALT column of the FS interface. The search will find all reports containing effort on projects matching the project_id_alt value. The search appears in the user's managed searches list, tagged with the value found in the ERSSConfig.xml element Interface.ProjectAltId.SearchName, plus the user's first and last name.

Account PI Interface (optional interface file):

The Account PI interface file is optional. If present, it should contain all PI's employee number and the associated project Id. Any PI found on this file will have a build-in My Certifications search available to him.

All of the interface files are expected to be loaded into ERS on a recurring basis, based on campus processing schedules. For ERS processing purposes, the recommendation is

that PPS Interface files be created and loaded after each payroll compute. Campus financial system interface files should be loaded to be in sync with the PPS interface files so that account/fund combinations present in payroll transactions can be matched to their FAU. The CS interface can be loaded on a recurring basis but should at least be present prior to initiating a new effort reporting period. The USER interface should be loaded as frequently as is required to maintain access to the ERS web application.

For more information on the format and content of these interface files, refer to the current version of the *ERS Interface Specification*.

For more information on how the interface files are used during ERS processing, read the sections of this document entitled “Operations Overview” and “Batch Interface File Loading.”

Initial Definition of ERS Schedule and Payroll Characteristics

During the installation process, a database install script is executed to populate the ERS schedule tables and the payroll transaction selection criteria tables. The data loaded by the install script should not be used for production purposes.

In planning for installation, the database installation script should be reviewed and modified to implement the desired schedule and to conform to payroll system conventions.

ERS allows flexible definition of a schedule, which is essentially a set of reporting periods and the attributes that will cause employees to be assigned to that schedule. Planning for the ERS schedule should address the following questions:

- What will the effort reporting schedule be?
- How many reporting periods?
- Which types of employee will be included in each reporting period?

With respect to payroll transaction selection criteria, ERS requires a current list of DOS codes used by the Payroll/Personnel System to identify payments for summer or off-period research. These codes must be loaded via the database installation script. In addition, the DOS Pay Category and DOS Time Code tables are loaded via the installation script and the values should be reviewed for any special situations that may exist in the Payroll/Personnel System.

Selection of the Project_ID Field in the FS Interface

The value selected for the Project_ID field is critical to ERS because it influences many aspects of the system’s appearance and operation.

The Project_ID groups together all FAU's (account/fund combinations) that should be reported as a single line on the effort report. Each distinct Project_ID that is flagged as *requiring certification* will appear on a separate line of the effort report "above the line".

The Project ID may be a sponsored project identifier from an existing campus system or it may be some portion of the FAU itself. In many cases, the Fund element from the FAU is an appropriate value to use as the Project ID, either alone or in combination with other components of the FAU.

Pre-Installation

The pre-installation process involves establishing the application environment on the target server. The pre-installation steps are:

A Java application server should be installed that is compatible with Java 5 (Java Platform, Enterprise Edition version 5).

The name of the Java platform for the enterprise has been simplified. Formerly, the platform was known as Java 2 Platform, Enterprise Edition (J2EE), and specific versions had "dot numbers" such as J2EE 1.4. The "2" is dropped from the name, as well as the dot number. So the latest version of the Java platform for the enterprise is Java Platform, Enterprise Edition 5 (Java EE 5).

Examples of suitable products include Apache Tomcat, IBM Websphere Application Server (version 6), or Sun Java System Application Server.

The Java application server must be configured with appropriate JDBC driver for the campus back-end DBMS. ERS ships with JAR files for native "Type 4" drivers for Sybase (jconn2), MS SQL Server (jtds), Oracle (ojdbc), and DB2 versions 7 and 8 (db2jcc) in the `WEB-INF/lib` folder. These JAR files may be copied to classpath locations as required by the app server's configuration documentation or may be used directly from the ERS application folder.

The application server must be configured to recognize JSP page fragment files (*.JSPF) which are used by the application. The steps to create this configuration are specific to the application server product being used.

In Tomcat, the following servlet mapping should be added to the server's WEB.XML:

```
<servlet-mapping>
  <servlet-name>jsp</servlet-name>
  <url-pattern>*.jspf</url-pattern>
</servlet-mapping>
```

In the Sun Java System Application Server, this mapping should be added to the server's DEFAULT-WEB.XML.

In IBM Websphere, the file IBM-WEB-EXT.XMI must be created with the "jsp.file.extensions" property set. A working version of this file is included in the application WAR file and will be installed during application deployment.

Configure EHCache Temp Directory

ERS uses EHCache as a secondary cache to speed up application performance. This mechanism stores frequently used, read-only data in memory so that it does not have to be accessed from the database repeatedly. EHCache uses the Java tempdir system parameter to determine where it will place its temporary disk files. In an environment where multiple instances of the ERS application are run on a single server, the tempdir parameter must be configured to point to a unique path so that the application instances do not attempt to write to the same cache files.

Consult the application server documentation for information on how to specify a value for the `java.io.tmpdir` system parameter.

The batch scripts provided with the ERS distribution provide this value and can be customized to supply an installation-specific value.

Hibernate Caching

ERS uses the caching features of Hibernate to improve performance. Certain tables that are read-only are loaded in the cache. In a clustered environment, each application server maintains its own cache and ERS does not provide mechanism to synchronize caches across the cluster. Therefore, tables that are updateable cannot be cached using this mechanism without creating potential data integrity issues.

Configuration File Management

Review the section below entitled “Configuration File Management” for additional requirements for establishing the application server’s CLASSPATH.

Session Management and Time-Out

The application server performs session management on behalf of the application. The session management services control the “time out” feature which invalidates a session after a period of activity. The steps to change the default value of 30 minutes are specific to the application server product being used.

Transaction Management and Clustering

ERS is enabled for cluster support by insuring that all updating is accomplished by database supplied transaction management. ERS uses the underlying transaction capability of the supported back-end database systems.

In addition, for critical online processes, ERS checks to insure that long-running transactions (any transaction that spans a network interaction), enforces an optimistic lock strategy that checks before updating that the data has not been altered by another online user.

Finally, ERS uses Hibernate facilities to insure that system assigned primary keys also come from the database and are guaranteed by the database to be unique.

Front-End Requirements

ERS supports most modern browsers on both PC and Mac platforms. Specific products and versions that have been tested are:

PC

Microsoft Internet Explorer (recommended)

Firefox (recommended)

Mozilla

Netscape

Mac OS X

Safari (recommended)

Firefox (recommended)

Mozilla

Netscape 7+

Microsoft Internet Explorer 5.2+ (not recommended)

Mac OS 9

Netscape 7

Mozilla

Microsoft Internet Explorer 5.1+ (not recommended)

Javascript must be enabled. The ERS application makes extensive use of Javascript which must be enabled on the browser in order for the application to load.

Cookies must be enabled. The application server makes use of cookies for session management which the ERS application relies on.

Deploy WAR

The WAR file should be deployed using the application server's web administration console. This insures that the application is placed in the correct directory or folder, and that application server configuration XML files are updated correctly.

If using the application server's web administration console, follow that product's instructions for deploying a new web application packaged in WAR format.

Note that multiple installations of ERS on a single application server will require choosing a different Context Root for each instance of the application.

It is possible to expand the WAR file from the command line using the Java `jar` command with similar parameters as `tar`:

```
jar -xvf /pathtowar/EffortReporting.war
```

If using the `jar` command for manual extraction, the command should run from the application target directory. First, create an `EffortReporting` directory within the application server's web applications folder, then change to that directory and run the `jar` command above. When installing multiple instances of ERS using this method, it will likely be necessary to manually edit the application context XML file to alter the Context Root name.

This selected installation directory or folder will be referred to as the `<application-folder>` throughout the rest of this document.

NB: with Release 9, if you are using tomcat, you must set an environmental variable on the catalina JVM:

```
CATALINA_OPTS=-Djava.awt.headless=true
```

This is most easily done by editing `catalina.sh` in your tomcat installation `bin/` directory.

Database Setup

The database setup process involves establishing the ERS application database on the target DBMS. The database must first be created in the target DBMS using the product's administrative tool.

The Create Script

The database create scripts are placed in the <application-folder>/sql folder.

Run the appropriate DBMS-specific create script to create all the ERS database tables, indexes and views:

ERS_CREATE_TABLE_SCRIPT_SYBASE	for Sybase and MS SQL Server
ERS_CREATE_TABLE_SCRIPT_ORACLE	for Oracle 8-10
ERS_CREATE_TABLE_SCRIPT_DB2_V8	for DB2 V8 (UDB or LUW)

The Install Scripts

The database install scripts are placed in the <application-folder>/sql folder.

The BASE installation scripts load several database tables with static data. These scripts should not require modification for a base installation.

Run the appropriate DBMS-specific install BASE script to load the ERS database tables:

ERS_INSTALL_BASE_SYBASE	for Sybase and MS SQL Server
ERS_INSTALL_BASE_ORACLE	for Oracle 8-10
ERS_INSTALL_BASE_DB2V8	for DB2 V8 (UDB or LUW)

The remaining tables contain static data used by the application and should not require modification for initial installation.

The MODIFY installation scripts load several tables with campus-specific static data and starter data to facilitate testing. These scripts should be reviewed for campus requirements and modified as necessary. Tables loaded by the MODIFY scripts which should be reviewed include:

ERSDOSCode – should include all DOS code values used in the campus Payroll Personnel System (PPS) to pay summer quarter or off-period research. These codes are typically named “Additional Compensation-Research.” These DOS code values must be identified so that the correct payroll transactions can be placed on a special off-period effort report produced for academic staff.

All other payroll transactions will be included in the effort report calculation, subject to the filtering values on the Time Code and Pay Category tables.

ERSScheduleType, ERScheduleModel and ERSReportingPeriod – these tables are used in the creation of the campus effort reporting schedule. A default schedule is created by the installation script. The default schedule can be modified by editing the installation script or can be modified using the System Administration screens after initial installation. See the System Administration section of this document for a complete discussion of the schedule definition process.

ERSUser – creates a login ID called “sys.admin” for initial setup and “sec.admin” for initial security administration. Change the email address on these INSERT statements to one that is accessible to the person installing ERS.

Run the appropriate DBMS-specific install script to load the ERS database tables:

ERS_INSTALL_MODIFY_SYBASE	for Sybase and MS SQL Server
ERS_INSTALL_MODIFY_ORACLE	for Oracle 8-10
ERS_INSTALL_MODIFY_DB2_V8	for DB2 V8 (UDB or LUW)

Database UserID

The web application uses a single login ID to access the back-end database. This login ID and password are coded into the Hibernate configuration files and should be updated based on the target DBMS.

The web application login ID must be GRANTED full access (INSERT, UPDATE and DELETE) to all database tables. No other database privileges are required. The application does not make use of any stored procedures.

A sample script that contains the necessary GRANT statements is provided in the <application-folder>/sql folder. The database userid should be substituted for the marker *WEBUSER* in this script.

ERS_TABLE_GRANT_SCRIPT_ALLDBMS	for all databases
--------------------------------	-------------------

Initial Customization

Although the master configuration file is documented in detail in the *ERS Customization Guide*, some minimal customization of this file is required to complete the installation and verification steps.

Configuration File Management

Sample ERS configuration files are provided in the <application-folder>/config folder. These files should not be used as distributed because they will be overwritten by deployment of a new ERS release. The sample files only require minimal modification to prepare them for an operational ERS installation. Copy the sample files to a location outside the ERS installation folder and edit the files as necessary to customize for your installation.

The configuration files that must be copied and may be modified are:

- `ERSTConfig.XML` – contains all ERS operating parameters
- `hibernate.properties` – contains database connection settings
- `ExternalizedStrings.XML` – contains customizable field labels for various parts of the web application

These contents of these files are documented in the *ERS Customization Guide*.

Multiple sets of configuration files can be created and stored separately for development, QA and production environments. The ERS application server searches for its configuration files on the CLASSPATH. To insure that the correct set of configuration files is used, modify the application server CLASSPATH by placing the path to the appropriate configuration file folder at the beginning of the CLASSPATH.

Configure Mail Server settings

The `ERSTConfig.XML` file contains settings to define the campus mail server which the application should use to send system-generated messages.

```
<Notification>
    <MailServer>smtp.ucop.edu</MailServer>
```

Modify the value of the MailServer item to a valid host name for a campus SMTP server that can send messages to campus e-mail addresses.

Configure Hibernate

Hibernate is the software layer that provides object persistence for the ERS application. It provides the database transaction management service for the application. It can also manage the database connection using JDBC services if you choose to run it in that manner. It supports several connection pooling services when running in directly-managed connection mode. See below for a fuller discussion of these issues.

The Hibernate configuration is contained in the DatabaseConfiguration section of ERSConfig.XML:

```
<DatabaseConfiguration>
  <PropFiles>
    <PropFile>hibernate.base.properties</PropFile>
    <PropFile>hibernate.trn01.properties</PropFile>
  </PropFiles>
</DatabaseConfiguration>
```

- The first properties file contains several Hibernate configuration settings that are common to all environments. The `hibernate.base.properties` file should not require modification.
- The second properties file contains several Hibernate configuration settings that are specific to the DBMS being used for the ERS application.

```
# Hibernate sample properties for Sybase
hibernate.connection.url=jdbc:sybase:Tds:sybserve4.ucop.edu:2125/ersdev
hibernate.connection.username=ersdbo
hibernate.connection.password=xxxxxx
hibernate.dialect=org.hibernate.dialect.SybaseDialect
hibernate.connection.driver_class=com.sybase.jdbc2.jdbc.SybDriver
```

With Release 9.1, ERS upgrades Hibernate to 3.2. This change will require that you change your DBMS connection properties file. In particular, the package name of all dialects has changed from `net.sf.hibernate` to `org.hibernate`. There is an example DBMS connection file under the `config/` directory in the Base ERS distribution. You must also explicitly declare the secondary cache provider, and the query parsing engine:

```
hibernate.cache.provider_class org.hibernate.cache.EhCacheProvider
hibernate.cache.use_second_level_cache true
hibernate.cache.use_query_cache false
hibernate.query.factory_class
org.hibernate.hql.ast.ASTQueryTranslatorFactory
```

Edit the values for these properties as appropriate for the campus installation. The `username` and `password` properties should be set to the database userID created in the previous installation step.

Note that multiple properties files can be created to support development, test and production environments. The `ERSEConfig.XML` file can be modified to reference a different Hibernate properties file thus changing the DBMS to which the application will connect.

Managing the JDBC Connection

Hibernate can be run in two different JDBC modes: either directly managed connections, or JNDI-referenced, container-managed datasource connections. With Release 8, ERS ships two example Hibernate connection configuration files: **hibernate.jdbc_conn_example.properties** and **hibernate.datasource_conn_example.properties**. The first gives the properties needed when using Hibernate to directly manage JDBC connections. The second file shows how to configure Hibernate for datasource mode. See your application server's documentation for information on how to define a JNDI datasource. The main advantage of using the latter method is that it gives access to the container's connection pooling.

Connection Pooling

When running in directly-managed JDBC mode, Hibernate supports plugin connection pooling layers. The supported CP implementations include:

Name	More information
c3p0	http://sourceforge.net/projects/c3p0
Apache DBCP	http://jakarta.apache.org/commons/dbcp/
Proxool	http://proxool.sourceforge.net/

ERS ships c3p0. It is suitable for testing and development. Experience has shown that it may not be production-quality. We recommend using one of the other available choices, or running hibernate in container-managed datasource mode, in which case you can use the connection pooling provided by the application server.

JDBC Isolation Level

With Release 9.2, we recommend running with the JDBC isolation level set to Committed Reads, which is a value of 2. If you are using a hibernate-managed JDBC connection, then use the **hibernate.connection.isolation=2** entry as found in the sample file. If you are using a container-managed datasource, then use the container's mechanism for setting JDBC properties.

Hibernate Database Dialects:

The following table lists all supported DBMSs and the corresponding value to set in the hibernate connection file for the **hibernate.dialect** property (prepended with **org.hibernate.dialect.**):

DMBS	hibernate.dialect value
DB2 390	DB2390Dialect
DB2 400	DB2400Dialect
DB2 V8	DB2Dialect
Firebird	FirebirdDialect
FrontBase	FrontBaseDialect
HSQL	HSQLDialect
Informix	InformixDialect
Ingres	IngresDialect
Interbase	InterbaseDialect
Mckoi	MckoiDialect
MimerSQL	MimerSQLDialect
MySQL	MySQLDialect
Oracle 10	Oracle10gDialect
Oracle 9	Oracle9Dialect
Oracle	OracleDialect
Pointbase	PointbaseDialect
PostgreSQL	PostgreSQLDialect
Progress	ProgressDialect
SAP DB	SAPDBDialect
SQLServer	SQLServerDialect
Sybase 11.9.2	Sybase11_9_2Dialect
Sybase Anywhere	SybaseAnywhereDialect
Sybase	SybaseDialect
TimesTen	TimesTenDialect

Customize Batch Processing Scripts

Shell scripts for invoking ERS batch processing programs are provided in the batch folder for both Unix and Windows. These scripts require some customization before they can be executed in the installed environment.

`SET-JAVA-PATH` – change the `JAVA_BIN_DIR` to point to the location where your Java JDK is installed

Change `CACHE-DIR` to the instance-specific temp directory where the EHcache temporary files can be written.

`SET-CLASSPATH` – change `WEB_ROOT` to the path to the installation directory of ERS on the application server.

Change `CLASSPATH` to the instance-specific configuration directory where the application configuration files are located.

Optionally, you may further append to the `CLASSPATH` to specify the location of a JDBC driver JAR that is required for your environment.

These scripts are references by the other batch processing scripts which do not require modification.

Read the README file in the `<application-folder>/batch` folder for more information on running ERS batch programs.

Installation Checklist

- ☒ Perform pre-installation steps on the host server to establish a Java environment and customize the Java application server.
 - ☐ Full JDK installed
 - ☐ Application server installed
 - ☐ Application server modified for JSPF
 - ☐ Config directory created and path added to CLASSPATH
- ☒ Deploy the application WAR file to the target server
- ☒ Create the application database
 - ☐ Run the database create script
 - ☐ Create a database user ID for the application and grant SELECT, INSERT, UPDATE and DELETE permissions on all ERS tables
 - ☐ Review and modify the database install MODIFY script
 - ☐ Run the install BASE script
 - ☐ Run the install MODIFY script
- ☒ Configure Hibernate
 - ☐ Edit `ERSConfig.XML` and Hibernate properties files
 - ☐ Copy to a config folder
 - ☐ Add the config folder path to the application server CLASSPATH
- ☒ Customize batch processing scripts
- ☒ Perform batch installation verification
- ☒ Start application server and monitor initialization for any error messages
- ☒ Perform online installation verification
 - ☐ Access web application URL
 - ☐ Request password for Sys.Admin
 - ☐ Follow password reset procedure
 - ☐ Login as Sys.Admin
- ☒ Prepare input files according to ERS Interface Specification
- ☒ Load interface files using batch programs
- ☒ Generate effort reports for one or more reporting periods

Batch Installation Verification

Change directory to the <application-folder>/batch folder.

Edit the set-classpath.sh and set-java-path.sh as indicated, then run `ers-install-test.sh` shell script to verify connectivity from the Java environment to the database verifying Hibernate, JDBC and driver components. This script now presents you with a command line prompt. Some test queries include:

```
from ReportingPeriod
from ItemCategory
```

Type a return to exit this program.

Note: This is a batch verification only and does not validate the application server configuration.

Modifications of this shell script may be required for the local environment. Check platform-specific variables that reference file system paths and edit as necessary – specifically `AS_ROOT` (for the application server root directory) and `JAVA_HOME` (for the JDK). Also, file permissions may need to be modified to allow script execution.

Expected output from test script:

```
02 Nov 2006 16:19:47,788 INFO TestHQL:137 - Starting test...
02 Nov 2006 16:19:57,685 INFO C3P0Registry:77 - Initializing c3p0-0.9.0.4 [built 23-
January-2006 22:20:29 -0500; debug? true; trace: 10]
02 Nov 2006 16:19:58,005 INFO PoolBackedDataSource:257 - Initializing c3p0 pool...
com.mchange.v2.c3p0.PoolBackedDataSource@142bece [ connectionPoolDataSource->
com.mchange.v2.c3p0.WrapperConnectionPoolDataSource@96cf11 [ acquireIncrement -> 2,
acquireRetryAttempts -> 30, acquireRetryDelay -> 1000, autoCommitOnClose-> false,
automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0,
connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester,
factoryClassLocation -> null, forceIgnoreUnresolvedTransactions -> false,
identityToken -> 96cf11, idleConnectionTestPeriod -> 3600, initialPoolSize -> 5,
maxIdleTime -> 1800, maxPoolSize -> 20, maxStatements -> 50,
maxStatementsPerConnection -> 0, minPoolSize -> 5, nestedDataSource ->
com.mchange.v2.c3p0.DriverManagerDataSource@16dadf9 [ description -> null, driverClass
-> null, factoryClassLocation -> null, identityToken -> 16dadf9, jdbcUrl ->
jdbc:jtds:sqlserver://erslab4.ucop.edu:1181/erstrain, properties -> {user=*****,
password=*****} ], preferredTestQuery -> null, propertyCycle -> 300,
testConnectionOnCheckin
-> false, testConnectionOnCheckout -> false, usesTraditionalReflectiveProxies -> false
], factoryClassLocation -> null, identityToken -> 142bece, numHelperThreads -> 3 ]
>>from ReportingPeriod
19 Sep 2008 14:40:37,395 INFO TestHQL:44 - TestHQL instance running test HQL=from
ReportingPeriod

Printing Record number: 1

acadEndDate = 2008-12-18 00:00:00.0
acadSrtDate = 2008-09-27 00:00:00.0
```

```
addDate = 2008-09-19 14:18:22.32564
addUser = *INSTALL*
certDate = 2009-03-15 00:00:00.0
chgDate = null
chgUser = null
endDate = 2008-10-31 00:00:00.0
id = 1
perCd = 1
perName = Fall 04/05 9/12 Academics
periodStatus = N
rdueSent = N
rptDate = 2008-12-15 00:00:00.0
srtDate = 2008-07-01 00:00:00.0
```

Printing Record number: 2

```
[...]  
>>
```

Online Installation Verification

Start the ERS application within the application server and monitor the console. The ERS startup servlet should produce output similar to the following:

```
Nov 2, 2006 2:34:15 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
Nov 2, 2006 2:34:16 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1271 ms
Nov 2, 2006 2:34:16 PM org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
Nov 2, 2006 2:34:16 PM org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/5.0.29
Nov 2, 2006 2:34:16 PM org.apache.catalina.core.StandardHost start
INFO: XML validation disabled
Nov 2, 2006 2:34:16 PM org.apache.catalina.core.StandardHost getDeployer
INFO: Create host deployer for direct deployment ( non-jmx )
Nov 2, 2006 2:34:16 PM org.apache.catalina.core.StandardHostDeployer install
INFO: Processing Context configuration file URL file:C:\jakarta-tomcat-5.0.29\conf\Catalina\localhost\admin.xml
Nov 2, 2006 2:34:17 PM org.apache.struts.util.PropertyMessageResources <init>
INFO: Initializing, config='org.apache.struts.util.LocalStrings', returnNull=true
Nov 2, 2006 2:34:17 PM org.apache.struts.util.PropertyMessageResources <init>
INFO: Initializing, config='org.apache.struts.action.ActionResources', returnNull=true
Nov 2, 2006 2:34:18 PM org.apache.struts.util.PropertyMessageResources <init>
INFO: Initializing, config='org.apache.webapp.admin.ApplicationResources', returnNull=true
Nov 2, 2006 2:34:19 PM org.apache.catalina.core.StandardHostDeployer install
INFO: Processing Context configuration file URL file:C:\jakarta-tomcat-5.0.29\conf\Catalina\localhost\balancer.xml
Nov 2, 2006 2:34:20 PM org.apache.catalina.core.StandardHostDeployer install
INFO: Processing Context configuration file URL file:C:\jakarta-tomcat-5.0.29\conf\Catalina\localhost\EffortReporting.xml
02 Nov 2006 14:34:27,544 INFO C3P0Registry:77 - Initializing c3p0-0.9.0.4 [built 23-January-2006 22:20:29 -0500; debug? true; trace: 10]
02 Nov 2006 14:34:28,404 INFO PoolBackedDataSource:257 - Initializing c3p0 pool...
com.mchange.v2.c3p0.PoolBackedDataSource@4391f0 [ connectionPoolDataSource ->
com.mchange.v2.c3p0.WrapperConnectionPoolDataSource@2209db [ acquireIncrement -> 2,
acquireRetryAttempts -> 30, acquireRetryDelay -> 1000, autoCommitOnClose -> false,
automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0,
connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester,
factoryClassLocation -> null, forceIgnoreUnresolvedTransactions -> false,
identityToken -> 2209db, idleConnectionTestPeriod -> 3600, initialPoolSize -> 5,
maxIdleTime -> 1800, maxPoolSize -> 20, maxStatements -> 50,
maxStatementsPerConnection -> 0, minPoolSize -> 5, nestedDataSource ->
com.mchange.v2.c3p0.DriverManagerDataSource@1ad6884 [ description -> null, driverClass
-> null, factoryClassLocation -> null, identityToken -> 1ad6884, jdbcUrl ->
jdbc:jtds:sqlserver://erslab4.ucop.edu:1181/erstrain, properties -> {user=*****,
password=*****, }, preferredTestQuery -> null, propertyCycle -> 300,
testConnectionOnCheckin -> false, testConnectionOnCheckout -> false,
usesTraditionalReflectiveProxies -> false ], factoryClassLocation -> null,
identityToken -> 4391f0, numHelperThreads -> 3 ]
02 Nov 2006 14:34:32,507 INFO ConfigLoader:130 - ERS configuration loaded from disk
02 Nov 2006 14:34:32,507 INFO ConfigLoader:131 - R7.001
02 Nov 2006 14:34:32,767 INFO StartupServlet:174 - ***** ERS initialization
successful
Nov 2, 2006 2:34:36 PM org.apache.catalina.core.StandardHostDeployer install
INFO: Processing Context configuration file URL file:C:\jakarta-tomcat-5.0.29\conf\Catalina\localhost\manager.xml
Nov 2, 2006 2:34:36 PM org.apache.catalina.core.StandardHostDeployer install
```

```
INFO: Installing web application at context path  from URL file:C:\jakarta-tomcat-5.0.29\webapps\ROOT
Nov 2, 2006 2:34:36 PM org.apache.catalina.core.StandardHostDeployer install
INFO: Installing web application at context path /tomcat-docs from URL
file:C:\jakarta-tomcat-5.0.29\webapps\tomcat-docs
Nov 2, 2006 2:34:36 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Nov 2, 2006 2:34:36 PM org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:8009
Nov 2, 2006 2:34:36 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/40  config=C:\jakarta-tomcat-5.0.29\conf\jk2.properties
Nov 2, 2006 2:34:36 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 20774 ms
```

Verify that the messages “ERS Initialization Successful” and “Server startup” appear, and that no exceptions have occurred within the startup servlet, then continue with the verification procedure.

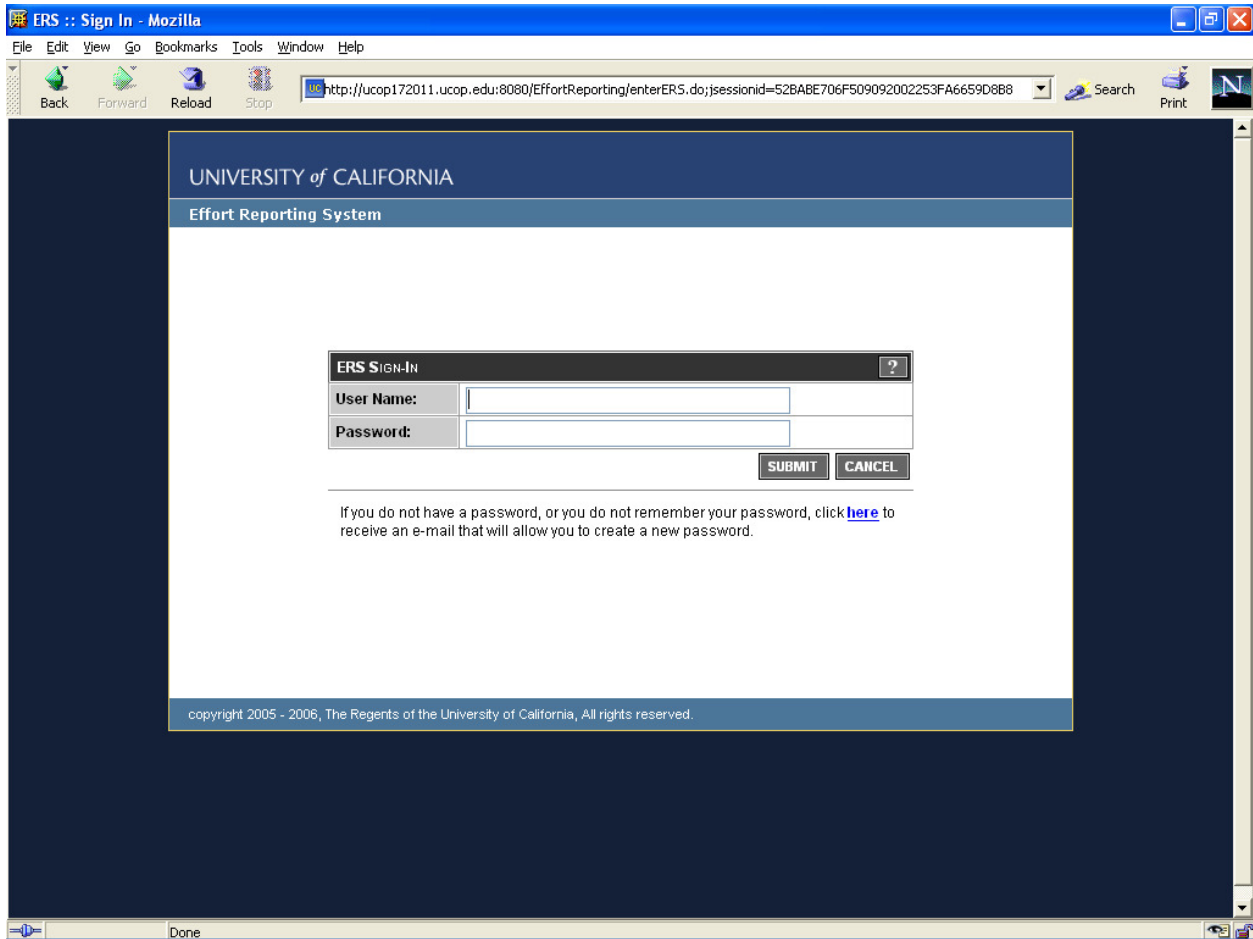
Structure Version Check

Starting with R9.2, ERS will log an error message if it finds the database structure does not match the current running code version, it will log an warning message:

```
[...] WARNING--the current database version is not in sync with the
application codebase! database version=9.1 app version=9.2 You will
most likely experience fatal errors trying to run the online
application!
```

Access the web application URL:
http://<hostname:port>/<application-folder>

The index.jsp file should be selected by default and a redirect will occur to the ERS signon page.

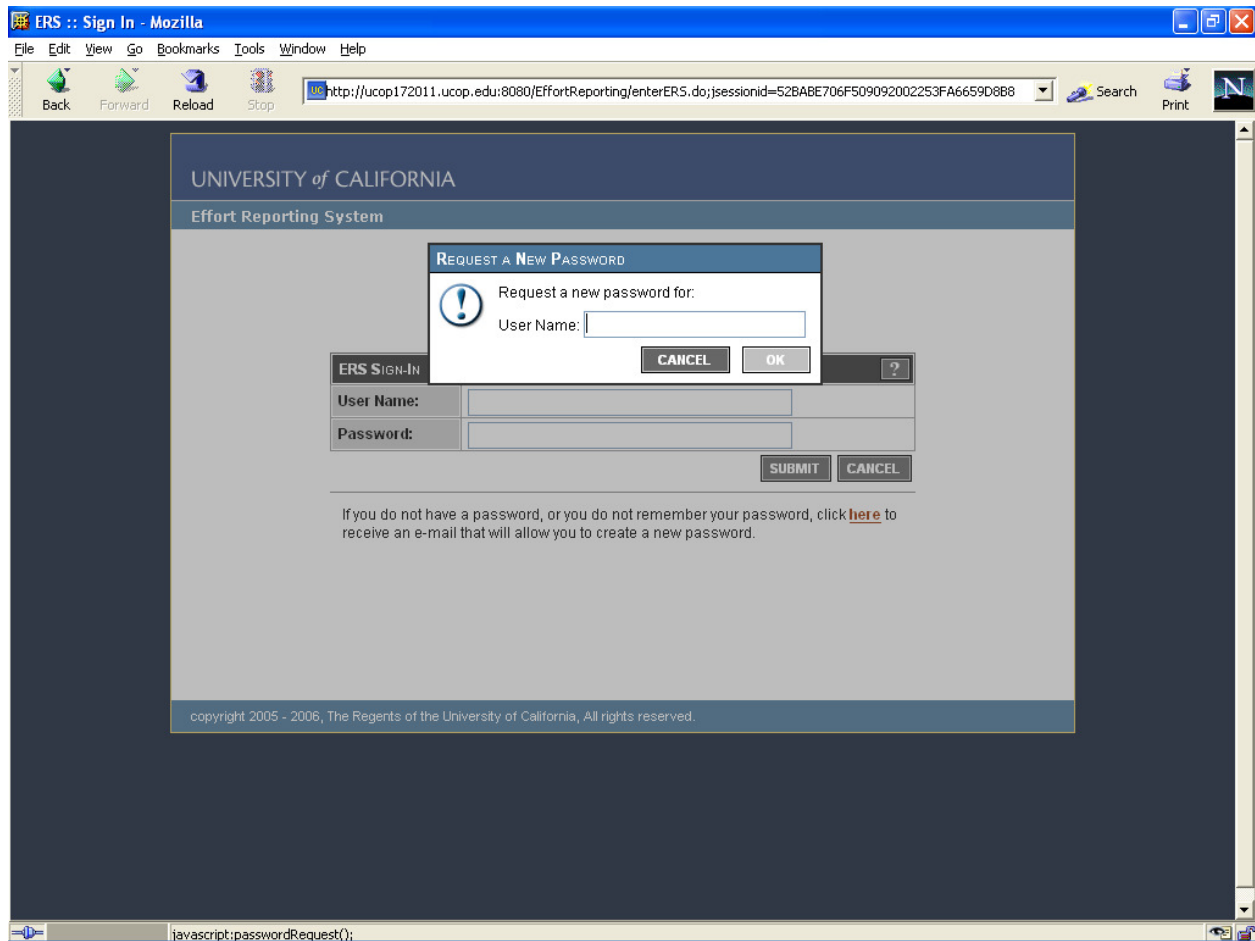


The screenshot shows a Mozilla browser window titled "ERS :: Sign In - Mozilla". The address bar displays the URL: `http://ucop172011.ucop.edu:8080/EffortReporting/enterERS.do;jsessionid=52BABE706F509092002253FA6659D8B8`. The page content is titled "UNIVERSITY of CALIFORNIA" and "Effort Reporting System". It features a sign-in form with the following elements:

- ERS SIGN-IN** header with a help icon (?)
- User Name:** text input field
- Password:** text input field
- SUBMIT** and **CANCEL** buttons

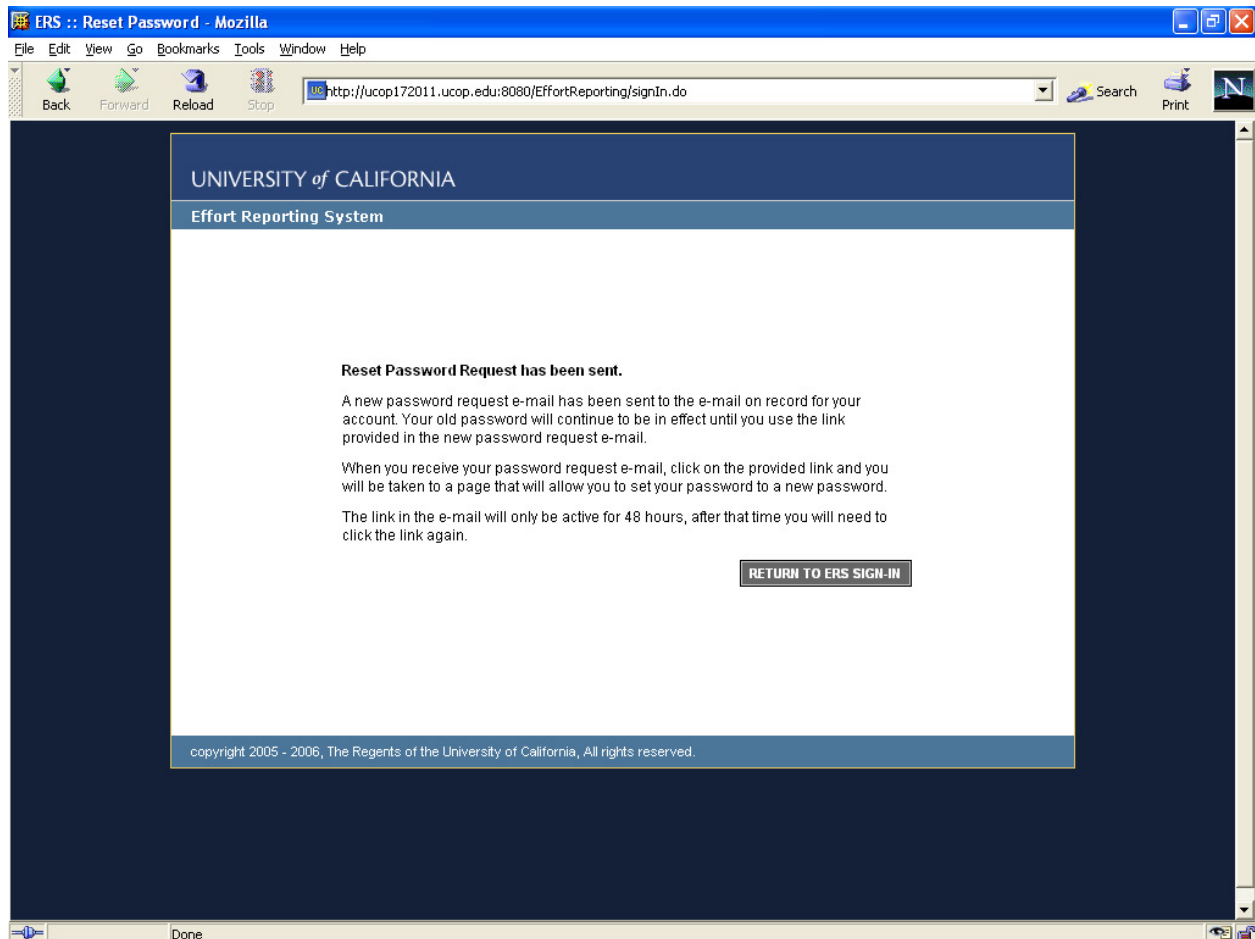
Below the form, a message states: "If you do not have a password, or you do not remember your password, click [here](#) to receive an e-mail that will allow you to create a new password." The footer contains the copyright notice: "copyright 2005 - 2006, The Regents of the University of California, All rights reserved."

From the signon page, click the link to request a new password.



Enter the sys.admin userID created with the installation script and click OK.

Receive confirmation that the password reset request has been accepted.



If the mail server configuration was successful and the sys.admin userID was created with a valid email address, a message should be received momentarily that contains a link to complete the password reset process.

When the password reset is completed, proceed to login normally to the application.

Note that no data will be visible in the web application until the batch programs to load the database have been run.

Operations Overview

ERS Operations consist of three main segments:

1. batch interface file loading
2. batch report period processing
3. online (web) application access

Note: ERS is not designed to support concurrent batch and online updates. It is important to insure that web application access is disabled during batch processing. The recommended technique for disabling application access is to stop the web application within the application server, or to stop the application server entirely during batch processing.

Batch Interface File Loading

ERS takes three required and two optional interfaces files as inputs. These files must be prepared using campus data according to the published specification for ERS interface files. Once the files are prepared they should be transferred to the ERS server and stored there for processing. A naming convention is recommended for identifying multiple files of the same type.

Each interface file should be loaded in the following order:

	<u>Interface</u>	<u>Description</u>	<u>Load Program</u>
1.	DEPT	Campus department list and organizational hierarchy	RunDepartmentInterface
2.	FS	Financial/contracts & grants system interface	RunFinanceInterface
3.	PPS	Payroll/Personnel System - payroll data. Can load multiple files as needed.	RunPARInterface
4.	CS	Cost sharing system data. This is an optional interface.	RunCostSharingInterface
5.	USER	User table. This is an optional interface for campuses using an external authentication mechanism.	RunUserInterface
6.	ORGDEPT	Org-HomeDept Interface	RunOrgDeptInterface

		Processes the optional interface file that associates organization unit codes with PPS Home Department codes. For use with the decentralized security administration feature. This program should be run after the DEPT interface is loaded.	
7.	Project Alt ID	Optional creation of searches based on sponsored project data	RunSponsoredProjectSearchProcessorInterface
8.	Pilot Period	Optional designation of a reporting period as a pilot period	RunPilotParticipantInterface
9.	WOS	Optional without salary employee (WOS) interface	RunWosEmployeeInterface
10.	ACCTPI	Optional account PI interface	RunAccountPIInterface

All of the load programs are in package `edu.ucop.ers.generate.batch` and should be qualified with the full package name when executed from the command line.

A sample shell script `ers-run-ILP-sample.sh` has been provided in the `/scripts` folder to demonstrate how to run ERS batch programs from the command line.

The ERS master configuration file `ERSConfig.XML` must be copied and customized prior to batch interface file loading. The path and folder name which contains the input files must be defined in the Interface/Filepath item:

Example for Unix:

```
<Interface>
  <Filepath>/home/ers/interfaces/</Filepath>
</Interface>
```

Example for Windows:

```
<Interface>
  <Filepath>z:\interfaces\ersfiles\</Filepath>
</Interface>
```

Note that the trailing slash or backslash is required on this parameter.

Required Parameters

All of the batch interface file loading programs require a single parameter – the name of the input file to be processed.

The path and folder information is taken from the configuration option described above.

If this parameter is missing, or references a invalid file, the programs will print a message to the console and exit.

Re-Runs

All batch interface file loading programs are designed to be re-runnable. In the event of a fatal exception that causes the program to exit, simply rerun the program with the same input file after correcting the problem.

Frequency

The frequency of loading the ERS database depends on campus operational considerations.

Typically PPS interface files will be loaded on a regular, ongoing basis as the PPS compute is run. The other interface files should be run on a regular basis but it is not required that they run as frequently as each payroll compute. It is only important that the latest DEPT, FS and CS data be loaded prior to beginning an effort reporting period.

All of the batch interface file loading programs are designed to process changes that occur over time. That is, it is not necessary to empty the database of old data prior to loading new data. The programs examine the input data and detect what information needs to be added and what information already exists in the ERS database and must be updated.

Monitoring

The batch interface file loading programs write diagnostic messages to the ERSMessageLog table. All data-related issues will be identified here rather than on the console. Error messages may be viewed using the System Administration web pages.

A notification type IERR (interface error) is provided to allow the notification subsystem to be used for monitoring batch interface file processing. In the event of a data error in a batch interface file loading program, any roles which are defined to receive this notification type will receive an email message recommending examination of the ERSMessageLog table.

JVM Parameters

One strategy R9.1 uses to improve batch processing includes upfront caching of certain domain data, such as funding sources. You must run these jobs with increased heap space. We recommend the following parameters:

```
-Xmn64M -Xms128M -Xmx768M
```

The affected jobs include:

- FS interface
- PAR interface
- Report Period Initiation
- Report Period Generation
- Report Resource Generation
- Late Pay

Batch Report Period Processing

Once all of the required input files are loaded into the ERS database, the next segment of ERS processing can begin. Report Period processing consists of three steps:

1. Initiating the reporting period
2. Generating effort reports
3. Generating security resources for new effort reports

<u>Acronym</u>	<u>Description</u>	<u>Program</u>
RPI	Report Period Initiation Identifies employees who should receive an effort report	RunReportPeriodInit
GEN	Report Generation Creates an effort report for each employee identified for the period	RunGenerate
RRG	Report Resource Generation Builds security descriptors for each newly created effort report	RunResourceGenerator
LP	Late Pay Process Re-issues effort reports affected by late-arriving payroll transactions	RunGenerateLateReports

Required Parameters

RPI, GEN and RRG accept a single parameter which is the period ID number to be processed. This value can be obtained by viewing the reporting period list in the System Administration web pages.

LP does not require any parameters.

Re-Runs

The batch report period processing programs cannot be rerun without manual intervention to clear any partial results and reset the period status indicator. Attempting to re-run a batch period processing program will result in an error message.

Frequency

Report period processing is based on the campus' defined reporting schedule. Typically the RPI, GEN and RRG programs will be run in sequence for each reporting period.

The LP process should be run after each PPS interface file is loaded to update any existing reports with new payroll data and to create any newly required effort reports for existing report periods that have been run.

Monitoring

The batch report period programs write diagnostic messages to the ERSMessageLog table. All data-related issues will be identified here rather than on the console. Error messages may be viewed using the System Administration web pages.

A notification type RNEW (new reporting period) exists to allow the notification subsystem to be used for monitoring batch report period processing. At the successful completion of a report period, any roles which are defined to receive this notification type will receive an email message announcing the availability of effort reports for the new period.

Additional Batch Programs

ERS includes additional batch programs for the notification subsystem and PI Normalization:

<u>Description</u>	<u>Program</u>
<p>Notification Dispatcher</p> <p>Processes the notification event queue and sends email based on system events.</p> <p>This program should be run periodically throughout the day using a system scheduler service.</p>	RunNotificationDispatcher
<p>Nightly Maintenance</p> <p>Performs various housekeeping tasks for system maintenance. As shipped, the only task performed is to send reminder notifications for reports coming due.</p> <p>Additional maintenance tasks may be added to this program in the future.</p>	RunNightlyMaintenance
<p>PI Normalizer</p> <p>Processes each record of the SponsoredProject table, gets the PI and Co-PI for each project, and inserts a new record into ERSPI table for each PI and Co-PI for a given project. [The objective is to normalize the PIs from a SponsoredProject by eliminating PI and Co-PI attributes and create a new schema ERSPI with SPID, EmpID, and PIInd (flag specifies whether the Employee is PI or Co-PI)]</p> <p>This program also processes each record of the ERSPIEmpl table, gets the PI and Co-PI, ProjectId, and inserts a new record into ERSPIEmplTank table for each PI and Co-PI for a given project. [The objective is to modify the ERSPIEmpl schema by normalizing the PIs and create a new schema ERSPIEmplTank with ProjectId, EmpID, and PIInd (flag specifies whether the Employee is PI or Co-PI)]</p>	RunPINormalizer

System Administration

This section describes the design and major concepts of ERS schedule definition, security administration and notification administration.

Schedule Definition

The ERS schedule is a customizable feature which defines the effort reporting periods that the campus chooses to process, and the characteristics of the employees that will be included in each period. Schedule definition is accessed from the System Administration web pages using the sys.admin login ID created during installation.

Schedule terminology:

Schedule Type: A schedule type defines a group of employees. Employees are assigned to a schedule type based on PPS attributes such as exempt status, appointment basis/paid over, personnel program code or other attributes.

Reporting Period: For each schedule type, one or more reporting periods are created. The reporting period defines primarily which payroll periods will be included in the effort reporting period.

Schedule Model: A schedule model is a set of sample periods that apply to the schedule type. The model acts as a “template” for creating a set of reporting period records for a given schedule type.

The normal process for creating the campus ERS schedule involves two required steps and one optional step:

1. Define schedule types and identify employee attributes that will be used to group employees. The supported employee attributes that can be used for schedule assignment are:
 - Personnel Program Code
 - Exempt Indicator
 - Academic Appointment Basis/Paid Over
 - Title Code
 - Home Department Code
2. Define reporting periods for each schedule type which correspond to the academic periods in use at the campus. For example, academic quarters or semesters. Make sure to include a reporting period for summer or off-period research for any schedule type that will include academic staff.

The recommendation is that a set of reporting periods be created in advance of a new fiscal year. It is permissible to create multiple sets of reporting periods for multiple years in advance, if desired.

3. Optionally, the campus can create a schedule model to simplify the creation of a set of new reporting periods for each year. The model can be used to generate reporting period records so that the reporting period information doesn't have to be manually entered.

Security Administration

ERS includes a built-in authorization module to manage permissions and roles which control access to effort reports. Campuses may use this internal authorization module or may use an external authorization mechanism described in the *ERS Customization Guide*.

When using internal authorization, after installation, the sec.admin login ID should be used to establish application security. This login ID is granted the security administrator role during installation and should then be used subsequently to establish users, roles and permissions. It is recommended that the sys.admin login ID not be used to administer security. The sys.admin login ID should only be used to create additional security administrators.

The internal authorization module involves establishing permissions which define a level of access to a set of effort reports. The permission includes a function, which is a user's intended action toward the effort report, and a resource, which is an identifier of that effort report.

ERS functions are:

View	user can view an effort report
Review	user can edit an effort report
Certify	user can certify an effort report
Payroll	user can view the payroll details of an effort report

ERS resource types are:

<i>Resource Type</i>	<i>Contains value</i>
Employee ID	Employee ID of the employee on the effort report
Home Department	Home department code of the employee on the effort report
Sponsored Project	Sponsored project – one for each project ID associated with a sponsored project line on the effort report
Funding Source	Funding source – one for each FAU associated with a sponsored project on the effort report

Funding Source- Fund Department	Funding source – fund department - one for each FAU associated with a sponsored project line on the effort report
Funding Source- Account Department	Funding source – account department – one for each FAU associated with a sponsored project line on the effort report

If you load your organization hierarchy into ERS, either first by turning on decentralized security administration, or by turning on the load-org-flat option in the application configuration file, and second, preparing a valid Department interface file that captures your organization hierarchy, there will be two other resource types available to you when creating permissions:

<i>Resource Type</i>	<i>Contains value</i>
FS Acct Dept Hierarchcial	Same as Funding Source – Account Department, but includes all departments in the subtree of the specified department linked to the effort report as an account department
Funding Source- Account Department	Same as Funding Source – Fund Department, but includes all departments in the subtree of the specified department department linked to the effort report as an fund department

ERS uses its internal authorization module to control access to parts of the application, in addition to controlling access to effort reports. To accomplish this, ERS defines these additional functions and resource type:

Application administration functions:

Coord	Provides access to tasks in the system administration module
Ops	Provides access to tasks in the system administration module
Security	Provides access to administer users, permissions and roles

Application administration resource type:

Admin Access Point	Defines a specific system administration task
--------------------	---

Defining a permission requires selecting the appropriate function or functions to control the access being given, and then specifying one or more resources to identify which effort reports the function will apply to.

Once permissions are defined, they may be added or grouped into roles. A role is a mechanism for collecting sets of permissions together so that the authority that the permissions represent can be shared among multiple users. This avoids needing to create sets of permissions multiple times.

Once a role is defined, it can be attached to one or more users, granting that user the access defined in the set of permissions contained in the role.

Security Administration – Mode of Operation

Centralized security administration is a mode of using the internal authorization module. All administration of permissions, roles and users is performed using the sec.admin login ID or another login ID granted the Security Administrator role.

Access to each of the components of security administration can be granted individually. It is possible to create a security administrator that can only create roles and permissions, but cannot create or edit users, or assign roles to users. Conversely, it is possible to create a security administrator that can only edit users and assign existing roles, but cannot create new permissions or roles.

Decentralized security administration is a mode of using the internal authorization module. Enabling this feature allows campuses to delegate authority for creating and managing ERS roles and permissions to departmental security administrators.

When the feature is enabled, several screens in System Administration change to allow editing and display of a security organization that owns a permission or role. A security administrator for that organization can manage any permissions or roles that are owned by that organizational unit. Without this feature, administration of permissions is not restricted and any user with security administrator role can create permissions and roles granting access to any effort reports in the system.

When creating permissions, departmental security administrators can only access resources that are within their scope of control. For example, a departmental security administrator can create a permission to view effort reports for people in their home department – but cannot create a permission to view effort reports for people that are in another home department.

In order to create a departmental security administrator, a user *without any security admin* roles must be granted the Decentralized Security Admin Administrator role. This role will allow access to the View Security Administrators screen, where you can see all previously designated departmental security administrators, edit them, and create new ones. This is the only means by which departmental security administrators can be edited or created.

Delegated security administration rights are hierarchical. If a security administrator is created for a high-level organizational unit, such as a division, that security administrator will automatically be given access to all subordinate organization units

such as sub-division or department. In this manner, security administration can be delegated to the appropriate level based on local operational considerations.

Note that this hierarchical behavior only applies to the ability to create permissions and roles. If you are a security administrator for a high level unit, you can create permissions for all of the subordinate units. However, each permission created must reference a specific organizational unit. A permission that references a high-level organizational unit as a resource does not imply that access is being granted to any subordinate organizational units.

In order to successfully implement decentralized security administration, the DEPT interface file must be properly setup to represent the campus organizational hierarchy. Each organizational unit in this file (OrgID) must be associated with a parent organizational unit (POrgID). Each parent organizational unit must exist in the file as its own organizational unit (OrgID.) The interface loading program will diagnose any rows that carry a POrgID that is not present in the file as an OrgID.

A valid organizational hierarchy must have at least one *root organization*. A root organizational unit is defined by a row that has the same value for OrgID and POrgID.

When preparing to enable the decentralized security administration feature, it is important to first enable the option via *ERSConfig.XML*, then load the DEPT interface file. The interface loader program performs special processing on the DEPT interface file when it detects that the decentralized security administration feature is enabled.

To create a departmental security administrator, a user with the security administrator role must add the security administrator role to a user and associate the new administrator with an organizational unit code. This is most simply accomplished by using the sec.admin user to assign the pre-defined security administrator role to the new administrator. The new administrator will then be able to create specific permissions and roles to control access to effort reports.

Users can be designated as departmental security administrators in two ways. Using system administration, a User can be edited to add the Security Administrator role and the associated organization unit for which they are being given control. Or, the organization tree can be browsed to select a specific organizational unit. Once selected, the Organization unit record can be edited to add the user ID of the person who should be given security administration rights.

Editing the security details for an organization unit also provides the ability to associate PPS Home Department codes with the organization unit code. While the normal mechanism for creating these associations would be by loading the ORGDEPT interface, it is also possible to add, edit and delete these associations manually using this page.

For more information on this feature, consult on-line help within the system administration module.

System Administration Access

Each of the tasks on the System Administration menu requires a special role for access. These roles are predefined during installation and consist of pre-built permissions that reference the Access Point resource type. The available roles are shown in the table below:

<u>Section</u>	<u>Task</u>	<u>Required Access</u>
Organization Administration	View Organization Tree	Decentralized Org Tree Viewer role
	View Administrators	Decentralized Security AdminAdministrator role
Schedule Definition	Manage Schedule Types Manage Report Periods Manage Schedule Models Create Reporting Periods from Model	Schedule Administrator role
Operations	View system message log Reload configuration	Operations Administrator role
Security	View Permissions	Permissions Administrator role
	View Roles	Roles Administrator role
	View Users	User Administrator role
Table Maintenance	View 'Additional Employees' List	Employee Include Editor role
	View 'Cannot Certify' List	Employee Settings Editor role
	Update Employee PI Indicator Update Employee Schedule Assignment	
	Update Effort Report Status	Effort Report Status Editor role
Report Generation	Generate Effort Report for Employee	Report Generator role
	Compliance Reporting	Compliance Report Generator role

These roles are available for selection on the View Users page and can be added to a User record in the same manner as campus-defined roles are added.

System Administration Tasks – Functional Descriptions

Security – Organization Administration

See *Security Administration*, above.

Schedule Definition

See *Schedule Definition*, above.

Table Maintenance

View 'Additional Employees' List

The 'Additional Employees' list is a list of employees that should always receive an effort report even if they would not normally be included because they do not meet the standard criteria ERS uses to determine for whom to issue an effort report. Adding an employee to this list will cause ERS to produce an effort report for the employee even if they received no pay on a sponsored project.

View 'Cannot Certify' List

The 'Cannot Certify' list is a list of employees that should be blocked from the normal implied access to their own effort report. There are three options in this function.

Exclude Self will prevent a user from viewing or editing their own report. *Exclude Self Certify* will prevent a user from certifying their own report. *Exclude My Projects* will prevent a principal investigator from having view, edit and certify access to all of the reports within their sponsored projects.

Update Employee PI Indicator

This function allows the employee record to be edited to flag the employee as a PI, if the system has not been able to detect this automatically for some reason.

Update Employee Schedule Assignment

This function allows the employee record to be edited to change the schedule to which an employee is assigned, if the system has not been able to detect this automatically or has derived the schedule incorrectly for some reason.

Update 'Not Required' Status

This function allows the status of an individual effort report to be set to 'Required' if it was initially generated as 'Not Required'. A report may be generated in a 'Not Required' status if there is no pay on a sponsored project, but the employee is a principal investigator (PI), or the employee is not a PI but was added to the 'Additional Employees' list.

Reports generated in these scenarios are set to 'Not Required' so they do not appear to be uncertified and outstanding, but if the administrator wants to require the report to be

certified they should set the status to 'Required' so that the report will be treated as an open report that must be certified.

Update 'Adjustment Required' Status

This function allows the status of an individual effort report to be set to 'Certified' if it was in a status of 'Certified/ Adjustment Required'. Normally, a report that is 'Certified/ Adjustment Required' will be set to 'Certified' automatically, when pending expense transfers are processed and recognized by ERS. However, there are some limited cases where a report may be in 'Certified / Adjustment Required' status, but will never be resolved by an expense transfer. This function may be used to set these reports to 'Certified' so they are treated as closed and not consider to be open reports that require intervention.

Report Generation

Generate Effort Report for Employee

There are occasionally situations when an administrator might need to generate an Effort Report on demand for an employee. For example, when effort reports were issued for a reporting period but an employee was erroneously excluded, or when an employee is terminating and an effort report is required for the period or portion of the period for which the employee worked so that certification can occur prior to the employee's departure.

Compliance Reporting

With Release 10.4, ERS now supports 7 compliance reports. This link leads to the form where the desired parameters can be set for a report.

Operations

View System Message Log

The System Message Log tracks and displays all system messages generated by ERS batch processing. These messages are used by central administrators and system operators to diagnose and resolve problems that may arise during normal ERS processing cycles and database loading.

Reload Configuration

The Reload Configuration option allows the system operator to update configuration parameters while ERS is up and running. These configuration parameters are modified by editing the ERSConfig.XML and ExternalizedStrings.XML files. Choosing this system administration will force ERS to reload these files and begin running with the modified settings.

See the appendices for the lists of pre-built permissions and roles that ship with ERS.

Notifications

ERS includes a notification subsystem which sends automated email messages to alert ERS users to various events that occur in the system.

The notifications that ERS will generate are:

1. when new effort reports are created as a result of report period initiation
2. when an error has occurred in batch processing
3. when an effort report is reissued by the system due to payroll changes
4. when an effort report is issue by the system due to payroll changes
5. when a certified line on a multi-line certification report is decertified

Notifications are related to security by way of the role definition. Users are selected to receive notifications by being members of a role that is defined to receive notifications of a selected type.

A role can have one or more notification event types associated with it. A typical configuration would be to create a role with permissions for a departmental ER coordinator, and then associate multiple notification types with this role so that the coordinator has both permissions and also receives notifications. It is possible to have a role with permissions only, and no notifications.

In addition to the event-driven notification types described above, ERS will generate reminder notifications when effort report certifications are coming due. The system supports two configuration options which specify the number of days prior to the certification due date to send reminders:

1. to departmental effort reporting coordinators, and,
2. to the principal investigator and co-principal investigator for each sponsored project that has open effort reports.

These reminder notification types may be disabled completely with a configuration option.

Post-Installation Tasks

Campuses should consider the following questions in planning for a production implementation of ERS:

- Review and modify installation script which loads campus-specific data values:
 1. ERSDOSCode
- Review and modify installation script which establishes a “starter” schedule and reporting periods:
 1. ERSScheduleType
 - a. ERSAppointment
 - b. ERSPersPgmCode
 - c. ERSExemptInd
 - d. ERSTitleCode
 2. ERSScheduleModel
 3. ERSReportingPeriod
- Establish operational procedures and scheduling of report period processing and monitoring batch processing
- Consider approaches for authentication and authorization and potential integration with existing campus frameworks
- Perform customization according to the *ERS Customization Guide*
- For initial startup for the first reporting period, insure that all required PPS PAR files have been gathered are loaded so that all transactions are available for report generation.
- Analyze database performance and tune as necessary. Several large tables are sensitive to updated statistics and index organization. Running routine maintenance on EREarnings, ERSEmpl and ERSFundingSource tables is advisable.

Application Source Documentation

Complete inventory and documentation of the application source code is provided in Javadoc format in a separate document.

Application Database Tables

Table Name	Description
ERSAcadApptBasis	Match values for schedule assignment
ERSAcadPayPer	Match values for schedule assignment
ERSAppointment	Match values for schedule assignment
ERSActualER	Each emp in a period that will get an ER
ERSCatEarnings	Earnings assigned to an ER "below the line"
ERSComments	Comment text associated with effort reports
ERSCriteriaAccountDepartment	Saved search criteria type: Account Dept
ERSCriteriaEmployee	Saved search criteria type: Employee ID
ERSCriteriaFundDepartment	Saved search criteria type: Fund Dept
ERSCriteriaFundingSource	Saved search criteria type: FAU
ERSCriteriaHomeDepartment	Saved search criteria type: Home Dept
ERSCriteriaMaster	Saved search criteria - parent record
ERSCriteriaMasterUser	Owners and sharers of saved searches
ERSCriteriaProject	Saved search criteria type: Project ID
ERSCSEmpl	Imported cost sharing data
ERSDOSCode	Match values for pay selection
ERSEarnings	Original earnings loaded from PPS PAR
ERSEarningsVersion	Earnings associated with each effort report
ERSEffortRpt	Parent record for an effort report version
ERSEmpl	Employee data built from PPS PAR input
ERSERResource	Security identifiers for each effort report
ERSExcludeUserToken	List of employee IDs who cannot certify
ERSExemptInd	Match values for schedule assignment
ERSFunction	Static list of security functions
ERSFundEarnings	Earnings assigned to an ER "above the line"
ERSFundingSource	Each FAU loaded from the FS interface file
ERSHomeDept	Home department list derived from employees
ERSIncludeEmpl	Employee IDs who should always get an ER
ERSItemCategory	Static list of below-the-line lines
ERSLatePay	Earnings that affect existing effort rpts
ERSMessageLog	Messages generated by batch processing
ERSNotification	Notification events triggered by the app
ERSNotificationRole	Associates event types and roles who get
ERSNotificationText	Notification message templates
ERSOrganization	Organization information from DEPT interfce
ERSPayCat	Match values for pay selection
ERSPeriodType	Static list of report period type codes
ERSPermission	Permission names created by security admin
ERSPersPgmCode	Match values for schedule assignment
ERSPI	Assign PIs for SponsoredProjects
ERSPIEmpl	Employee IDs of PIs identified thru FS load
ERSPIEmplTank	Employee IDs of PIs identified thru FS load
	Each emp that belongs in a given rpt period. Dead table (but not dropped)
ERSPossibleER	starting with R9.2.
ERSPreference	Stores user preferences for web application
ERSResetToken	Stores password reset tokens

ERSResource	Resource IDs created as part of permissions
ERSResType	Static list of security identifier types
ERSRole	Role names created by security admin
ERSRolePermission	Associates roles with permissions
ERSRptCatItem	Effort report lines - "below the line"
ERSRptFundingSources	Funding sources related to each ActualEffortReport
ERSRptFundItem	Effort report lines - "above the line"
ERSRptPeriod	Stores defined effort reporting periods
ERSSchDept	Match values for schedule assignment
ERSSchAppt	Match values for schedule assignment
ERSSchModel	Defines reporting period templates
ERSSchType	Defines groups of rpt pds and emp attribs
ERSSponsoredProject	Each sponsored project from the FS intf
ERSStatusCode	Static list of report status codes
ERSTimeCode	Match values for pay selection
ERSTitleCode	Match values for schedule assignment
ERSUseFunction	Associates permissions and functions
ERSUser	Each user loaded via the USER interface
ERSUseResource	Associates resources and permissions
ERSUserRole	Associates roles with users
ERSWeightedEarning	Earnings that have been weighted for combos
ERSFilterCategory	Groups report statuses for filtering
ERSStatusFilter	Individual status codes in filter groups
ERSOrgAdmin	Associates users (admins) with org codes
ERSOrganizationFlat	Denormalized version of ERSOrganization
ERSOrgDept	Associates org code with PPS home dept code
ERSVersion	Stores release level of current DB structure. Do not edit!
ERSCompRptParams	Saved compliance report parameters
ERSPilotParticipant	Maintains list of pilot participants for each pilot reporting period.

Pre-Built Permissions

Permission ID	Permission Name
SUPERUSER	Superuser Permission
SYSTEMADMIN	System Administrator
SECURITYADMIN	Security Administrator
COORDINATOR	ER Coordinator
SYSTEMOPERATOR	System Operator
AccessRptGen	Access Report Generation Screen
AccessRptPerInit	Access Report Period Initialization Screen
AccessEmpSettings	Access Employee Settings Screen (PI Indicator, exclusion lists, schedule setting)
AccessIncludeEmp	Access Include Employee Screen
AccessERStatus	Access Change Report Status Screen
AccessSchdAdmin	Access Schedules and Periods Administration
AccessOps	Access Message Log and Config Reloader
AccessPermAdmin	Access Permissions Administration
AccessRoleAdmin	Access Roles Administration
AccessUserAdmin	Access Users Administration
AccessOrgTree	Access Org Tree Viewer (decentralized security only)
AccessOrgAdmin	Access Org Administration (decentralized security only)
AccessCompRpt	Access Compliance Report Screen
AccessSecAdmin	Access Security Administration (decentralized security only)

Pre-Built Roles

Role ID	Role Name
SUPERUSERS	Superusers
SUPERUSERS	SUPERUSER
SYSTEMADMIN	System Administrator
SYSTEMADMIN	SYSTEMADMIN
SECURITYADMIN	Security Administrator
COORDINATOR	ER Coordinator
SYSTEMOPERATOR	System Operator
AccessRptGen	Report Generator
AccessRptPerInit	Report Period Initiator
AccessEmpSettings	Employee Settings Editor
AccessIncludeEmp	Employee Include Editor
AccessERStatus	Effort Report Status Editor
AccessSchdAdmin	Schedule Administrator
AccessOps	Operations Administrator
AccessPermAdmin	Permissions Administrator
AccessRoleAdmin	Roles Administrator
AccessUserAdmin	User Administrator
AccessOrgTree	Decentralized Org Tree Viewer
AccessOrgAdmin	Decentralized Org Administrator
AccessCompRpt	Compliance Report Generator
AccessSecAdmin	Decentralized Security Admin Administrator