

## Supported Browsers

- Current and recent Firefox, IE, Safari, Seamonkey, Opera, and Chrome versions are all supported
- Requires Browser to have Javascript enabled – the CORAL interface relies heavily on AJAX, JQuery and JQuery plugins

## Server-Side Requirements and Setup

### Database

CORAL runs with a MySQL 5 backend. Each CORAL module uses its own MySQL schema. The recommended format for schema name is `coral_MODULENAME_prod`, e.g. `coral_resources_prod`. Since the database name is set in the `configuration.ini` file this name can be easily changed. For security purposes it is recommended to have a “coral” user which has select, insert, update and delete privileges only (no create table privileges). This “coral” user should have access to all CORAL databases for interoperability.

The CORAL modules are heavily data-driven. As a result we have provided an initial “snapshot” of our data which will be inserted when the MySQL tables are created. Most of this data may be edited through the front-end if desired – refer to the user guide for more information. For data models and the MySQL Workbench file, refer to <http://coral-erm.org/documentation>

### Server-side scripting

A supported version of PHP 5 is a requirement for CORAL since it uses Object Oriented code. If you are unsure whether you’re running PHP 5, be sure to install using the web installation script since it will validate your version. See Installation section, below.

### Authentication

Authentication is required for CORAL since all users are given specific privilege levels (for more on privilege levels refer to the user guide). CORAL has two choices for authentication – either to tie in to your university/library’s authentication system or to install the CORAL Authentication Module. For more information on the CORAL Authentication module, please refer to the [Authentication Documentation](#).

To utilize your library’s existing authentication system, you will need to know the Server Variable name to use within PHP. This is a setting within `configuration.ini` called “`remoteAuthVariableName`” which will be set in the installation process (see Installation, below). The `remoteAuthVariableName` indicates the PHP server variable name that is set in apache upon authentication and so CORAL will recognize the user’s login. Different server setups may have different variable names. Assuming you’re using an authentication system already, you can do a `phpinfo` script to see if this variable is being set and what it is. (just make a `.php` file and put `<?php phpinfo(); ?>` in it ) Probably the easiest way to find the variable name would be to search for your user ID.

If you still have problems using your existing authentication system, we would recommend subscribing to the listserv and looking at the archives to see how other institutions have made it work, or searching the forum. As a last resort, each module has 'user.php' in the root directory which can be modified to retrieve your users' login ID.

Also, the `.htaccess` file should require your authentication system, or you should have another way to require authentication. If you have questions about this, contact your system administrator. CORAL has limited use directly with LDAP. Within all of the modules (except for Licensing) any user allowed by the `.htaccess` may visit the site; if LDAP has been set up in the `configuration.ini` it will display their name in the upper right hand corner. This connection is optional.

## Application Architecture

CORAL is written using object oriented code. All classes and configuration are defined under the `/admin/` directory. The admin directory has an `.htaccess` file in order to prevent web users from accessing config data.

`/directory.php` is used to load all of the classes. This is called from every page and for the ajax functions and also contains some globally used functions.

`/admin/configuration.ini` defines config settings (module settings, database and ldap)

Common classes are in `/admin/classes/common/` - includes configuration, database connection, LDAP, Utility, Email

### Database

All database tables have matching classes and all database SQL code is done within methods of these classes.

Every class extends the `DatabaseObject` for generic SQL handling (retrieving all values, inserts, updates, deletes). `DatabaseObject` is in `/admin/classes/common/` and the database classes are located in `/admin/classes/domain/`

### Web side

Pages - `admin.php`, `index.php`, detail view scripts and reports are all shells for positioning divs. Within each div data is updated via ajax calling `ajax_htmldata.php`. This way when changes are made to the data the display can be immediately updated without needing to refresh the entire page.

A jquery plugin called thickbox is used to display the "popup" forms with the grayed out background.

`ajax_forms.php` contains all of the html for forms that are being used by thickbox.

`ajax_processing.php` contains all of the processing (eventually sql insert/updates, etc) that utilize

the database classes - this is where data is sent after being entered in the forms

`header.php` and `footer.php` are defined within `/templates/` directory.

`user.php` and `setuser.php` are used for authentication.

`/css/styles.css` is the general css file used by all pages in coral.

## Javascript

All javascript code is stored under `/js/` directory. This directory contains javascript for base pages (`index.php`, `admin.php`) where the names of files correspond with the javascript file name. (e.g. `admin.php` uses `admin.js`)

The directory `/js/plugins/` directory contains third party javascript (jquery library, autocomplete, thickbox and other plugins) .

The `/js/forms/` directory contains all javascript files corresponding to the thickbox forms that are generated from `ajax_forms.php`

## Naming standards

If you would like to contribute code / database changes to the project it is recommended that you stay within the following naming conventions.

### Database Naming:

- Attribute and entity names in singular form and using present tense
- Entity names are uppercased with no spaces or underscores between words (camel case)
- Attribute names have first word lower cased, subsequent words have first letter upper case
  - e.g. `documentID`, `sentDate`, `documentText`
- Primary keys are always Auto Incrementing Integers beginning with the table name and 'ID' suffix
  - e.g. for the "Document" table, "documentID" is the name of the primary key
  - (note that the generic database object expects this naming convention – if you do not use this name, you must override the primary key name in the class)
- When possible, attribute names are suffixed with class words
  - Code - Has a list of possible values e.g. INV, ENC
  - ShortName - Name
  - Identifier (ID) - Unique Identifier, generally surrogate key
  - Number - Numeric value
  - Indicator (Ind) - A Boolean value (0/1)
  - Date - Date
  - DateTime - Date and time stamp

### Code:

- Refer to previous sections for directory structure, script locations and script naming.

- Variable, class and attribute names mimic the database standards with camel case.
- Method names are prefixed with:
  - Is – Answers a question, returns true/false, e.g. for User class `isAdmin()` method
  - Get – retrieves a value
  - Set – sets a value

## Installation

Installation can occur in one of two ways – either through the web installation script or manually. Web installation will provide advantages over manual installation because it will check MySQL privileges and PHP version.

When you first download the scripts, it is recommended to place each one into a parent `/coral/` directory if your server is sharing space with other applications. This is optional but it helps group all of the coral applications together.

For example you may have:

```
/coral/resources/  
/coral/licensing/  
/coral/organizations/  
/coral/usage/  
/coral/reports/  
/coral/auth/
```

**NOTE:** You will need to copy the CORAL-Main module files directly into the `/coral/` directory. This will provide an `index.php` file which displays links to the modules you have installed.

Depending on whether the module allows you to upload attachments or documents you may also need to `chmod 777` some directories:

For Licensing – both `/licensing/attachments/` and `/licensing/documents/` directories

For Usage Statistics – both `/usage/archive/` and `/usage/log/` directories

For Resources – `/resources/attachments/`

## Web installation

Under each module directory, visit `http://.../coral/modulename/install/` and follow instructions on the screen.

**Installation Notes:** \* There is a separate install for each module \* The CORAL-Main module does not have an installation script. You simply need to copy the module files into the parent `/coral/` directory. \* If you need to re-run installation for any reason you can do so **but it will remove any data in the database and reset to original data load!** \* Be sure to remove the `/install/` directory once installation is complete

## Manual Installation

### Install database tables

1. Create new database schema (recommended name is `coral_modulename_prod`)
2. Apply the SQL file to your new database
3. Manually add an admin user into User table – this is required for the first person to administer users.
  1. Open your MySQL client
  2. Navigate to your database (`coral_modulename_prod`) and to the User table
  3. Insert your LoginID (your externally authenticated login ID) with a Privilege ID of 1 (equates to admin) and your first and last name, if desired.
  4. This is the user who will have access to the Admin page to first administer other users. From the Admin Page, you can designate another user with the admin privilege for that person to also administer users.

### Update `/admin/configuration.ini`

1. Rename `/admin/configuration_sample.ini` to `/admin/configuration.ini`
2. Under `[settings]`
  1. `organizationsModule=`, `cancellationModule=`, `licensingModule=`, `resourcesModule=`, `usageModule=`

These switches define whether you have the other CORAL modules installed and would like the interoperability turned on. The switch will also turn on/off the link to other modules (the Change Module drop down menu in the upper right-hand corner). Valid values are Y or N. Interoperability currently occurs between:

    1. Resources to Licensing and Organizations
    2. Organizations and Licensing (both directions)
    3. Usage Statistics to Organizations
  2. `organizationsDatabaseName=`, `licensingDatabaseName=`

When you have interoperability turned on between Organizations and Licensing database or Organizations and Usage Statistics you must supply the database name (e.g. `coral_organizations_prod`) for the connection. Also note that the user you connect with must have select, insert, update and delete privileges to this database.
  3. `[resources only] defaultCurrency=`

This will set the currency that is selected by default in the Resource Payment section. Default currency needs to be a valid Code in the Currency tab on the Admin page.
  4. `[resources only] enableAlerts=`

This will turn on the alerting checkbox on the Acquisitions tab on the Resource page for subscription end dates. The days in advance and email addresses can be set in the Alert Settings on the Admin page. Also note this requires a scheduled job, see below on Resources Alerts Scheduling.
  5. `[resources only] catalogURL=`

If you would like to include a link directly into your catalog from the Resource page you can place your URL with the parameter for the System Number at the very end. For example, ours is: `"http://alephprod.library.nd.edu/F/?func=direct&doc_number="`
  6. `[resources only] feedbackEmailAddress=`

This is the global email address for all workflow notifications - starting, queue entries and completion. Emails for specific steps can be set in the User Group section on Workflow/User Group tab of the Admin page. The feedback email address also shows on the Resource Right Panel. It is optional.

7. [licensing only] useSFXTermsToolFunctionality=  
This will turn on/off the SFX tab and Terms Tool Report in the licensing module. Valid values are Y/N.
  8. [usage only] reportingModule=  
This will turn on/off the reporting link in the usage statistics module. Set to Y if you are using the reporting module add-on (this is a separate install). Valid values are Y/N.
  9. [usage only] useOutliers=  
This will turn on/off the outlier flagging feature. This feature will calculate abnormal spikes in usage and color code them red, orange or yellow based on the formula defined in the admin tab. Valid values are Y/N.
  10. [usage only] baseURL=  
This will turn on or off a created link within the module to your open URL.
  11. remoteAuthVariableName=  
This is the server variable used by php to determine the user authorized to log in. For example, we use `$HTTP_SERVER_VARS['REMOTE_USER']`, `$_SERVER['PHP_AUTH_USER']`, or `$_SERVER['WEBAUTH_USER']` may be other names.
  12. [resources only] defaultsort=  
Changes the default sort order for the resources. Example: `defaultsort="\TRIM(LEADING 'THE ' FROM UPPER(R.titleText)) asc"` It is optional.
3. Under [database]
    1. type=mysql  
currently only MySQL is supported but other database types could be supported if modifications to the generic database classes are made
    2. host=  
MySQL server - could be localhost as well
    3. name=  
Database name (e.g. coral\_resources\_prod)
    4. username=  
recommended to have a single user with select, update, insert, delete access to all CORAL modules
    5. password=  
password for above mentioned user
  4. Under [ldap] (optional – not for Licensing Module or Usage Statistics Module)
    1. Host=  
server name
    2. Search key =  
filter parameter used in `ldap_search` call – for more info refer to <http://php.net/manual/en/function.ldap-search.php>
    3. Base DN=  
The Base DN for the directory(`base_dn` parameter used in `ldap_search` call)
    4. First Name Field=  
Name of the field that LDAP uses for first name – for us it's "givenname"
    5. Last Name Field=  
Name of the field that LDAP uses for last name – for us it's "sn"
  5. For security reasons, remove the `/install/` directory once the installation is complete
  6. If you are using an external authentication system, you will need to require it in the `/coral/[module_name]/.htaccess` file
  7. If you are installing more than one module for interoperability you may see errors from either side until both modules are installed.
  8. Join the listserv used for product updates, support, and general discussion by sending an email to [listserv@listserv.nd.edu](mailto:listserv@listserv.nd.edu). Leave the subject line blank and include 'SUBSCRIBE CORAL-ERM'

Your Name' in the body; ex. SUBSCRIBE CORAL-ERM John Smith.

## Resources Alerts Scheduling

The Resources module includes a script called `sendAlerts.php` that can be run nightly to send out subscription alerts. To set it up via cron job, you will need to invoke php. You need to access the alerts page via curl or wget in order to generate emails with functional links:

```
0 0 * * * /usr/bin/wget http://YOUR_PATH.../coral/resources/sendAlerts.php
```

You will also need to verify that `enableAlerts` is set to Y in `/resources/admin/configuration.ini`

Note that this will send an email to the `feedbackEmailAddress` also set in `configuration.ini`. For temporary testing purposes you may wish to change or remove this email address. The number of days in advance and other email addresses can be modified in the Alert Settings tab of the Admin page.

## Additional CORAL Resources customization options

### Payment formatting

To change the format of the display of amounts (e.g. use a comma in the place of a period), you can modify the `/resources/directory.php` file, `integer_to_cost()` function. The `number_format` function is a php function – refer to PHP documentation for more information:

<http://php.net/manual/en/function.number-format.php>

### Date formatting

To change the format of the date from the US standard MM/DD/YYYY you can modify the `/resources/directory.php` file, `format_date()` function. The `date()` function is a php function – refer to PHP documentation for more information:

<http://php.net/manual/en/function.date.php>

The date will need to be convertible by the php `strtotime` function. For European dates the `strtotime` function requires dashes be used instead of slashes. Also for the year, be sure to use Y (uppercase) for 4 digit year and y (lowercase) for 2 digit year.

Additionally for date formatting you will need to update a javascript file for the date picker calendar functionality. Be sure this format matches the format you set in `directory.php`!

`/resources/js/common.js`, line 35:

```
Date.format = 'mm/dd/yyyy';
```

or

```
Date.format = 'dd-mm-yyyy';
```

## Email customization

The templates for the workflow notification and alert emails are located in `/admin/emails/`

You can modify the text in these as needed, leaving the words inside brackets `< >` as is – you can change where they appear but leave the format alone.

## Upgrades

Upgrades to CORAL will be announced through the CORAL ERM Listserv. If you wish to be added, please consult <http://coral-erm.org> or email [coral-admin@listserv.nd.edu](mailto:coral-admin@listserv.nd.edu) directly.

In general, small upgrades with simple coding changes will not be given their own version numbers. If an institution notifies us about a bug we will let them know directly when the bug fix is available. Versioning will happen when there is a database structure change or configuration file change.

When performing upgrades your configuration file (`/admin/configuration.ini`) should always be backed up first so that when you check out the new code you are sure to not overwrite it.

After you have checked out the new code through GitHub, there will be a file called `UPGRADE_README` in the install directory which will detail the database changes and configuration changes incorporated in that version change.

As with the Installation, you have two choices for how to upgrade – either manually or web install.

For the web install, go to `/install/update.php` and follow the instructions on the screen.

For the manual install, refer to the `UPGRADE_README` file in the install directory for instructions on how to upgrade – instructions may be different depending on what types of changes are included with the upgrade.

**After performing upgrade, be sure to remove your `/install/` directory.**

## New to PHP?

If your institution is new to the PHP/MySQL setup, installation may be a bit more tricky.

First off, make sure that you're both installing php with the `-with-mysql` option and installing the `phpmysql` package. To verify that php and mysql are installed and working correctly, you can make a test php script with only the following:

test.php contents:

```
<?php phpinfo(); ?>
```

To debug issues with authentication you can override authentication in the `/coral/modulename/user.php` script.



You may also wish to change the error reporting level in `php.ini` if you are getting blank pages in the install or with the CORAL pages themselves.

## Possible issues during installation (or - things to verify if CORAL is not working at this point)

Verify you're running PHP 5

Verify your database user can connect to the MySQL database and that all tables were created properly

If your login isn't working, verify that you have required your authentication service in the `.htaccess` file.

You should be prompted with your institutions default login screen before getting to the CORAL screens.

If you're not sure what your `remoteAuthVariableName`, you can try the following: Create a file called `test.php` on your web server with the following line:

```
<?php phpinfo(); ?>
```

Then, bring this up in your web browser. Search for your login ID and it should display the server variable in the lefthand column.

If interoperability isn't working (you'll know it's not working when you get red errors), verify that you have the database setting and that your database user can connect to that database. Also, be sure to complete installation of both modules involved.

If your remote user auth variable isn't working, verify that you have `register_globals = On`