

The Easy Edit Suite - Installation Guide

For EES Phase 1 (Matrix 3.28.0 to 3.28.2) and EES Phase 2 (Matrix 3.28.3 to 4.4.0).

This guide will help you install the *Easy Edit Suite* on your Squiz Matrix System.

This guide assumes that you are installing Squiz Matrix into */home/websites/* - please make adjustments to commands as required.

You cannot use both the *Easy Edit Suite* and a *Simple Edit Interface* on a single *Design file*.

If you have added the *Easy Edit Suite* to a *Design* and need to remove it, refer to the *Removing the Easy Edit Suite From a Design* section in this chapter.

Downloading the Easy Edit Suite to your Server

Running the following commands will download a copy of the latest stable version of the *Easy Edit Suite* using *wget* and unpack it in to the *matrix/data/public* folder.

The downloaded file will contain a version number, so be sure to replace **[version]** with the current version number in the commands below. If the downloaded version does not match your Squiz Matrix version, please download a different copy from the download page on the Squiz Matrix web site.

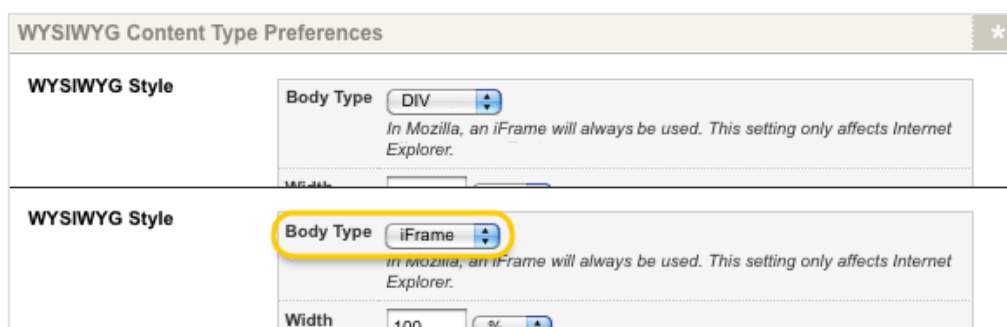
```
$ cd /home/websites/matrix/data/public
$ wget http://matrix.squiz.net/download/ees.tar.gz
$ tar -zxf ees_[version].tar.gz
$ rm ees_[version].tar.gz
```

Changing Global Preferences

The *Easy Edit Suite* requires that the *WYSIWYG Style - Body Type* option be set to *iFrame*. Please note that this change will not affect current WYSIWYG usage.

To set this option:

1. Click on the *System Configuration* button on the top right hand corner of the *Administration Interface* - a drop-down list will appear under this icon. From this list select *Global Preferences*. The *Global Preferences* screen will appear.
2. Acquire the lock(s) on the *Global Preferences* screen and locate the *WYSIWYG Content Type Preferences* section, as shown in the figure below.



The *WYSIWYG Content Type Preferences* section of the *Global Preferences* screen

3. In the *Body Type* field of the *WYSIWYG Style* options, select *iFrame* from the drop-down list, as highlighted in the figure above.
4. Click *Commit* to save this change.

The Javascript API

A *Javascript API* asset needs to be created for the *Easy Edit Suite* to communicate with the Squiz Matrix system.

Creating the Javascript API and Setting Permissions

To create the *Javascript API*:

1. Right click on the *Web Services Folder* in the *Asset Map* and select *New Child -> Web Services -> JavaScript API*.
2. Enter the name **Easy Edit Suite** for the asset and click *Commit*.
3. Right click on the *Javascript API* and select *Permissions*.
4. Acquire the lock(s) on the page and, under the *Read Permissions* section, select *Grant* for *Public Permissions*.
5. Click *Commit* to save this change.

Configuring the Javascript API

On the *Details* screen of the **Easy Edit Suite Javascript API**, change the following fields:

1. **Status**: set the *Status* of the *Javascript API* to *Live*.
2. **Root Nodes**: select the *Site* assets that are to use the *Easy Edit Suite*. This should include the root nodes of any assets that are to be editable on the *Easy Edit Suite* that may not be located under a *Site*, such as *File* or *Image* assets linked under the *Media Folder*.
3. **Allow Attributes on Create**: select *Yes*.
4. **Return JSON**: select *Yes*.
5. **Ignore Permissions**: select *No*.
6. Enable all the API functions in the **Core, Linking, Permissions, Workflow, Traversing** and **Metadata** sections.

Once you have made these changes, click *Commit*.

Tip: Keep a note of the API Key that is displayed on this screen while you are editing. You will need it later in the installation process.

Applying URLs

Each *Site* with a unique URL that is to use the *Easy Edit Suite* will require their URLs to be applied to various locations. This will ensure that there are no cross-domain conflicts with the EES JavaScript calls to the *Javascript API* asset.

Upon first installing the *Easy Edit Suite* and/or adding new *Site* assets to use the *Easy Edit Suite*, check the following:

- The new URL(s) has been applied to the *Site* asset.
- The new URL(s) has been applied to the *Designs Folder*. Please note that if you are using an alternate design configuration to control your *Site*'s design, then ensure the new URL(s) are applied to the relevant parent asset instead.
- The new URL(s) has been applied to the *Web Services Folder*. Please note that if your EES *Javascript API* is not located under the *Web Services Folder*, you will need to apply the URL(s) to the relevant parent asset instead.

Tip: Remember to maintain the URL format of system folders when applying the URL of your Site. For example, if the URL of your Site is <http://www.example.com>, you would apply this URL to the *Designs Folder* as http://www.example.com/_designs and to the *Web Services Folder* as http://www.example.com/_web_services.

Constructing the Configuration File

The *Easy Edit Suite* assumes the presence of a configuration file containing JSON. A template of this configuration file can be found in `<easy_edit_system_root>/Examples/EasyEditConfigSample.js`.

To setup an editable configuration file inside your system, locate an accessible location inside Squiz Matrix. This file will need to be accessible via a URL, be live and have public read access.

1. Create a *JS File* asset, naming it **EasyEditConfig**, as shown in the figure on the right.
2. Edit the file and save the content sourced from the `EasyEditConfigSample.js` file.
3. Apply *public read access* to the file.
4. Change the status of the file to *Live*.



The configuration file

Tip: Go to the *Web Paths* screen of the file and take note of the URL displayed there as you will need it later in this guide. The URL will be in the format `/__data/assets/js_file/xxxx/[assetid]/EasyEditConfig.js`

Configuration Options

You can set a number of configuration options on the *Easy Edit Suite*'s configuration file:

- **debug**: set this option to either **true** or **false** to tell the system if it is allowed to perform debugging operations.
- **titleBar**: this text will be shown in the title bar of the browser. Use the keyword replacement `@asset_name@` to use the name of the asset in the title bar.
- **lockRefreshInterval**: this option determines how often the system should re-acquire locks for the currently active screen. The default value for this is 120 seconds, the system default. This should be altered to match the current system's lock refresh interval, if it has been changed.
- **assetScreenCache**: set this option to either **true** or **false** to decide whether the system should re-acquire new asset information each time a screen change is initiated. Setting this option to true will result in less ajax calls to the *Javascript API* but may result in stale data, particular if an asset has the potential to be edited by more than one user at a time.
- **JSAPI -> key**: this option must contain the *API Key* of the *Javascript API*, as shown in the figure below. This key is obtained on the details screen of your **Easy Edit Suite Javascript API**. If the *API Key* at any time changes, the configuration file will need to be updated.

```
// in stale data if users are likely to be editing the same asset
assetScreenCache:      false,

// JS API configuration options
JSAPI: {
  key:                  "1111111111"
},

// Suffixes for matrix urls - This needs to match matrix system c
// values
editSuffix:            " edit",
```

Setting the API Key in the configuration file

- **Suffixes:** a representation of the suffixes found in the *Squiz Matrix System Configuration* menu. If these suffixes are not using the system defaults, the values should be reflected in this option.
- **cascadeThreshold:** at times, the *Easy Edit Suite* will warn editors if they are about to affect a large number of assets. This number is determined in this option. A high value will result in less warnings for users but could potentially result in users unwittingly cascading changes across large sections of the system, resulting in failed actions or stale hipo jobs.
- **overlayTimeout:** When the loading animation is presented to an editor, it is generally waiting on a response back from the server. The action the editor is performing may result in a failed request on the server (due to memory exceeding, execution timeouts, database errors etc.). This option sets a timeout (in seconds) after which a warning will be displayed to the editor informing them that their request may have failed. Editors will have the option to continue with the request or cancel the request, which will reload the current asset in the editing interface.

Adding the Easy Edit Suite to your Design's Parse File

To use the *Easy Edit Suite* on a *Site*, the *Parse File* of the site's *Design* must be edited to include the EES design.

To do this:

1. Locate the *Design* of the *Site* in the *Asset Map*. Right click on the *Design* and select *Details*.
2. Acquire the lock(s) on the *Details* screen. Locate the *Details* section of the screen and in the *No Frames in Simple Edit Interface* field, select *Yes*. Click *Commit* to save this change.
3. Right click on the *Design* and select *Edit Parse File*. Acquire the lock(s) on the page.
4. At the very top of the parse file, before any other code, add the following:

```
<MySource_AREA id_name="exit_area" design_area="exit"
print="no"></MySource_AREA>

<MySource_AREA id_name="nested_EES_customstylesandscripts"
design_area="nest_content" print="no" />

<MySource_AREA id_name="simple_edit_EES" design_area="show_if">

    <MySource_SET name="condition" value="simple_edit_mode" />

    <MySource_THEN>

        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
        <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
            <head>

                <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />

                <!-- The TITLE is dynamically updated by EES code, see
your EES config.js file to customise -->
                <!-- Related install step:
http://manuals.matrix.squizsuite.net/ees/chapters/installation#Constru
cting-the-Configuration-File -->
                <title>Asset Name : Squiz Matrix : Easy Edit
Suite</title>

                <!-- Easy Edit CSS -->
```

```

        <link href="/__data/ees/easyedit.min.css"
rel="stylesheet" type="text/css" media="screen" />

        <!-- MATRIX/Easy Edit JS API (replace with correct
file path to JS API within web services folder, file path may vary per
implementation) -->
        <!-- Related install step:
http://manuals.matrix.squizsuite.net/ees/chapters/installation#The-
Javascript-API -->
        <script type="text/javascript"
src="/_web_services/xxxx.js"></script>

        <!-- jQuery Files Required -->
        <script type="text/javascript"
src="/__data/ees/Libs/jquery/jquery-1.4.2.min.js"></script>
        <script type="text/javascript"
src="/__data/ees/Libs/jquery/ui-1.8/ui.custom.min.js"></script>
        <script type="text/javascript"
src="/__data/ees/Libs/jquery/jquery-ellipses.js"></script>
        <script type="text/javascript"
src="/__data/ees/Libs/jquery/validate-
1.6.0/jquery.validate.min.js"></script>

        <script type="text/javascript">
            // This global variable is only required in rare
cases.
            var eesSiteURL = '';
        </script>

        <!-- Easy Edit Configuration (replace with correct
file path, file path may vary per implementation) -->
        <!-- Related install step:
http://manuals.matrix.squizsuite.net/ees/chapters/installation#Constru
cting-the-Configuration-File -->
        <script type="text/javascript"
src="/__data/assets/js_file/xxxx/[assetid]/xxxx.js"></script>

        <!-- Easy Edit Core Scripts (path to install directory
within data/public/) -->
        <script type="text/javascript"
src="/__data/ees/easyedit.min.js"></script>

        <!-- Refer to nested standard page (set to raw html)
for custom CSS and JS file references. Avoids editing Design file. -->
        <MySource_PRINT
id_name="nested_EES_customstylesandscripts" />

        </head>
        <body></body>
        </html>

        <MySource_PRINT id_name="exit_area" />

    </MySource_THEN>

</MySource_AREA>

```

5. Edit the URL under the heading *Easy Edit Configuration* in the parse file. Ensure you replace **xxxx** and **[assetid]** with the values from your configuration file's URL.
6. Ensure the URL under the heading *MATRIX/Easy Edit JS API* points to the location of the JP API asset you created earlier.
7. Click *Commit* to save this change.

Tip: You may need to use the global site URL print tag in your Design's Parse File if you are experiencing 404 not found errors when trying to load the EES scripts. An example of this is as follows:

```
<script type="text/javascript" src="<MySource_PRINT id_name="__global__" var="site_link" full_url="true" />/_data/ees/eesscript.js"></script>
```

To add your own CSS or JS customisations, enable the *nested_ESS_customstylesandscript* nested design area on the base customisation of your design. In this design area, nest a *Standard Page* asset containing your script or css tag source file references. For more information, refer to the CSS Customisations and JavaScript Plugins chapters in this manual.

Setting the Easy Edit Suite Login Design

A *Login Design* is available for the *Easy Edit Suite* to maintain a consistent look and feel for users.

To apply the *Login Design*:

1. Right click on the *Site* that you have configured the *Easy Edit Suite* for and select *Settings*.
2. Acquire the lock(s) on the *Settings* screen and locate the *System Defined Login Design* section, shown in the figure below.
3. Select the *Change* button in the *New?* field (titled *Change?* if a *Login Design* is already applied).

The screenshot shows a settings interface titled "System Defined Login Design". It is divided into three sections: "Current", "Change?", and "Override?".

- Current:** A table with two columns: "URL" and "Effective Design". The "URL" cell contains "www.example.com/ees". The "Effective Design" cell contains an information icon, a lock icon, and the text "ees_login_design (Id: #516)".
- Change?:** A dashed input field contains "ees_login_design", followed by "Id : # 516". To the right are three buttons: "Cancel", "Clear", and "Reset". Below this is the text: "The Design that is to be used whenever someone attempts to access this asset without".
- Override?:** A dashed input field is empty, followed by "Id : #". To the right are three buttons: "Change", "Clear", and "Reset". Below this is the text: "You can override the design this asset would inherit from its parent, with a design that only. This override design does not affect the inheritance to this asset's children of any".

The System Defined Login Design section

4. Locate the *ees_login_design* asset (located in the *System Design Folder*), right click on it and select *Use Me*.
5. Click *Commit* to save this change.

Removing the Easy Edit Suite from a Design

To remove the *Easy Edit Suite* from a *Design*:

1. Acquire the lock(s) on the *Edit Parse File* screen of the *Design* file.
2. Remove the *Exit Design Area* from the top of the parse file and click *Commit*. This will remove references to the EES in the parse file of the *Design*.
3. Acquire the lock(s) on the *Details* screen of the *Design* file.
4. In the *No Frames in Simple Edit Interface* field, select *No* and click *Commit*.

The *Easy Edit Suite* will now be removed from your *Design*.