## **VersionManager Help**

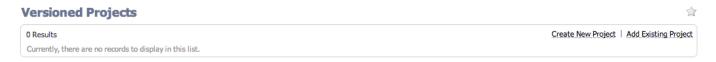
### **Initial configuration**

This plugin relies on background jobs to manage ElectricCommander artifacts as the backing store for XML exports of versioned projects. By default, these jobs are run on the "default" resource and "default" workspace, and the artifacts are published to the "default" repository. You can change these values by modifying the properties in the "settings" nested sheet for the VersionManager project. **Click here** to access these properties.

These jobs are run using the special "admin" user's credentials to prevent missing objects depending on access control settings. If you have changed the "admin" user's password, you will have to update a credential stored on the VersionManager project. **Click here** to access this credential and change the password.

# Viewing versioned projects

This plugin is used to manage versions of Commander projects. On installation and promotion, browse to the "Projects" tab and then the "Versioned" subtab to view versioned projects. At first, this table will be empty:

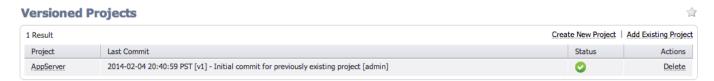


# Adding existing projects

Click "Add Existing Project" to begin versioning an existing project:



You will now see your existing project in the "Versioned Projects" table, set to version 1 with a commit comment of "Initial commit for previously existing project":



To start, add as many of your existing projects as you would like to be versioned:



### Creating new projects

Click "Create New Project" to create a project and immediately begin versioning it:

# Name: MyApplication Description: Continuous delivery workflows that tie together different components to build my app. Default Resource: Browse Default Workspace: Browse

Version 1 of the new project shows up with a commit comment of "Initial commit for new project":

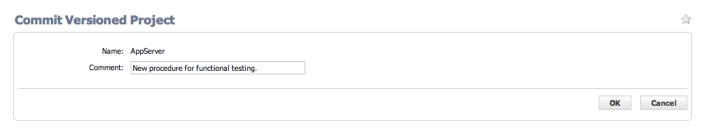


### **Committing changes**

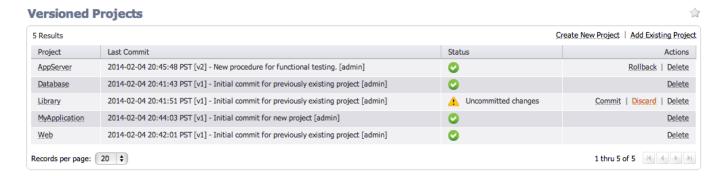
A job runs on a regular basis in the background checking to see if projects have changed. When *anything within the project* has changed (new procedure, modified step command, new property values, etc.), the project's status will say "Uncommitted changes":



Click "Commit" to commit the changes to a project. The user will be prompted for a commit comment to describe the changes:

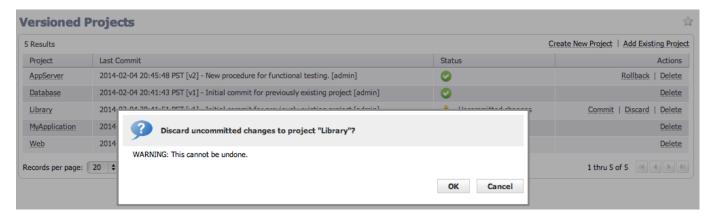


The changes are committed, the version is revved, and the user's commit comment is seen on the "Versioned Projects" table:



### **Discarding changes**

If you want to discard all changes made to a project since the last commit, simply click "Discard" and click "OK" when presented with the prompt to return the project to it's previous state:

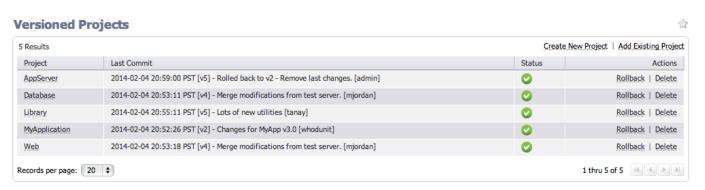


# Rolling back changes

If you want to revert to a previous version of a project and remove all changes since, click "Rollback", select the previous version, and provide a commit comment describing the change:



The project will return to the selected version. Note that the versioning scheme always moves forward, so the rollback version is simply stored as a new commit:



- Add existing projectCreate new projectCommit changesDiscard changesRollback changes