

# Plugin testing

## Building a plugin

In order for the plugin to be picked up by the ETICS client you will have to create source packages while you build the plugin. This is done by expanding the **Packaging & Repository** section and checking **source: Create source package**

## Plugin-list.xml

The plugin-list.xml contains a list of all the plugins telling the ETICS client how to install them and what their current version is. To test plugins with putting them in the release version we have a special **0.0.0** version of the plugin-list.xml that we point to the unreleased plugin versions.

This list can be updated directly by anyone with access to ETICS' AFS space. It is located here:  
**/afs/cern.ch/project/etics/builds/default/org.etics/org.etics.plugins/0.0.0/noarch/plugins-list.xml**

If you are upgrading a plugin, increment the version number. If you are creating a new plugin you can copy the configuration from a similar plugin and edit the fields that differ.

## Testing a plugin locally

The first step in testing a plugin should be to test it on a local machine and run the build commands directly on it. This allows you to easily change the code if you need to do some adjustments. If you have completed the steps above building the plugin with ETICS and updating the plugin-list.xml, installing or updating a plugin is simple.

1. Install the ETICS client
2. Set up a workspace folder (run the `etics-workspace-setup` command on a new folder)
3. Inside your workspace folder, call the **etics-plugin-manager install --file 0.0.0-1@default --verbose FindBugsPlugin** command to install a plugin. Replace FindBugsPlugin with the name inside the name tag in the plugin-list.xml. 0.0.0-1 points to the version and age of the plugin-list.xml and default specifies the default volatile repository.
4. You will now find the plugin under **etics/lib/python2.6/site-packages/plugins**.
5. To run the plugin you need to use one of the properties mentioned below in the "How to run a plugin" in the **etics-build** command.

## Testing a plugin remotely

When you are testing a plugin you need to tell ETICS that it should not use plugins from the production build-status.xml, but the 0.0.0 version.

1. Add **plugins\_release=0.0.0-1@default** to the Append Requirements field in the Host section (click on more options) of the ETICS remote build submitter. 0.0.0-1 points to the version and age of the plugin-list.xml and default specifies the default volatile repository.
2. To run the plugin you need to use one of the properties mentioned below in the "How to run a plugin". Use the Custom Build Options field, to see the field you need expand (click on more options) in the Build Execution section.

## How to run a plugin

There are three options for making a plugin run:

1. Setting the **default.profile** property to the plugin name. The plugin name is not what is in the <name> tag in the plugin-list.xml. It is usually in a short form and is given in the plugin's register method `profiles = [ 'pluginname' ]`. Example: **-p default.profile='FindBugs'**
2. Setting the default.profile to a predefined profile. Plugins are sorted into profiles based on which languages they support. So if you set **default.profile='java'** you will run all the Java plugins. In order to do this you need to add the plugin to the Java profile (TODO: how to do this). Example: **-p default.profile='java'**
3. Add the **--autoprofile**. The autoprofile is the SLOC count plugin to automatically select language profiles for each components. Again it is necessary to add the profile to the correct language profile. Example: **--autoprofile**

Look at the [EMI SA2 metrics guidelines for more info](#).