Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

The Eclipse IDE.
Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

The Eclipse IDE.
Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

- open-source
- multi-platform

The Eclipse IDE.
Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

- open-source
- multi-platform
- extensible

The Eclipse IDE.
Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

- open-source
- multi-platform
- extensible
- advanced editor

The Eclipse IDE.
Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

- open-source
- multi-platform
- extensible
- advanced editor
- multiple layouts

The Eclipse IDE.
Mojave is an Eclipse plug-in for Cactus. Eclipse is an open-source multi-platform graphical IDE written in Java for development in Java, C/C++, COBOL, and PHP. More information about Eclipse, including downloads, can be found at http://www.eclipse.org.

- open-source
- multi-platform
- extensible
- advanced editor
- multiple layouts
- SVN interface
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
- **build** the project

Mojave.
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
- build
- do a clean build

Mojave.
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
- build
- clean build
- **run** the executable

Mojave.
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
- build
- clean build
- run
- create a subfolder

Mojave.
Mojave: Features

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects.

- edit variables
- build
- clean build
- run
- create
- **update** the repository
Mojave: Remote Build

Mojave automatically downloads Cactus thornlists and prepares an environment for development. Mojave options are available through an Eclipse menu button. Through the Mojave menu, one may edit variables, perform builds, and run projects. It can even build remotely!

- edit variables
- **build** remotely
- clean build
- run
- create
- update

Mojave.
To build remotely, under **Mojave - Edit Variables...**, simply set the variable remote.machine to the host name of the target machine, then click **Mojave - Build**. Eclipse will automatically build the project on the target machine, provided the target machine is configured to allow remote builds.
The Mojave plugin is easy to install. Simply:

- Click **Help → Install New Software**.

**Step 1.**
Mojave: Installation

The Mojave plugin is easy to install. Simply:

- Click **Help → Install New Software**.
- In the 'Install' dialogue, click the **Add** button.

Step 2.
The Mojave plugin is easy to install. Simply:

- Click **Help → Install New Software**.
- In the ’Install’ dialogue, click the **Add** button.
- In the ” dialogue, type *Mojave* for the Name field and for the URL, http://mojave.cct.lsu.edu/Mojave-Update.
The Mojave plugin is easy to install. Simply:

- Click the **Available Software Sites** link. In the dialogue window which opens, mark all checkboxes for sites for C++ and Java resources.

Step 4.
Mojave: Installation

The Mojave plugin is easy to install. Simply:

- Click the Available Software Sites link. In the dialogue window which opens, mark all checkboxes.
- Select the Mojave site from the drop-down list of sites, then mark the checkbox for 'Mojave' in the list of software packages. Click Next.
The Mojave plugin is easy to install. Simply:

- Eclipse will automatically calculate dependencies and install the Mojave plug-in. This may take a while.

Step 6.
Before you may use the features of Mojave, you must create a Mojave project.

To create a new Mojave Project, first select New - Project from the File menu.
Before you may use the features of Mojave, you must create a Mojave project.

In the dialogue window that follows, select **C/C++** or **Fortran** for your primary development language.
Before you may use the features of Mojave, you must create a Mojave project.

- If you select Fortran, then select **Fortran Project**. You need the Photran plugin installed.
- In the dialogue window that follows, select **Empty Project** under **Makefile Project** near the bottom.
- Name the project, then click **Next**.
Before you may use the features of Mojave, you must create a Mojave project.

- If you select C++, select either C or C++ if you don’t have the Cactus code downloaded; otherwise select **Empty Project with Existing Code**.
- Name the project, then click **Next**.
Before you may use the features of Mojave, you must create a Mojave project.

- You may see a screen with checkboxes on the left-hand side for configurations you would like to use; make sure Default is checked.
Before you may use the features of Mojave, you must create a Mojave project.

- If you have Cactus already installed, you can create a project from disk.
- In the next dialogue window, select the **Mojave Disk Project** radio button, then specify the path to the Cactus source directory.
Before you may use the features of Mojave, you must create a Mojave project.

- If you don’t have the Cactus source installed, you may create a **Mojave Download Project**.
- Simply enter in your e-mail address and the user name of the account you use on remote machines.
Before you may use the features of Mojave, you must create a Mojave project.

- The source will refresh in case you chose a disk project; if you chose a download project, it will download. In either case, this will take some time.
- After it is done, you will have successfully created a new project.
- You may now use the features provided by the Mojave menu.
The Mojave menu code resides in an XML file, .mojave.xml. As the XML file shows, the Mojave menu items are front-ends to commands and scripts used in Cactus operations. How to open the file:

- Hit Ctrl+Shift+R or click on Navigate - Open Resource and type .mojave.xml in the text box.
Mojave: Development

The Mojave menu code resides in an XML file, .mojave.xml. As the XML file shows, the Mojave menu items are front-ends to commands and scripts used in Cactus operations. How to open the file:

- If .mojave.xml does not appear, go to File - Open File...
The Mojave menu code resides in an XML file, .mojave.xml. As the XML file shows, the Mojave menu items are front-ends to commands and scripts used in Cactus operations. How to open the file:

- If .mojave.xml does not appear, go to **File - Open File** . . . then open `workspace-name/project-name/Cactus/.mojave.xml` in your home directory.

Finding .mojave.xml.
The XML file is comprised of actions which correspond to the menu items. The series of commands set within each action tag is executed when the menu item is clicked.

- Notice how the **Build** action corresponds to the **Build** menu item.

Structure of `.mojave.xml`. 
The XML file is comprised of actions which correspond to the menu items. The series of commands set within each action tag is executed when the menu item is clicked.

- The first command under **Build** checks if the `remote.machine` variable is set.
The XML file is comprised of actions which correspond to the menu items. The series of commands set within each action tag is executed when the menu item is clicked.

- The first command under **Build** checks if the `remote.machine` variable is set.
- It is included with `–remotemachine` in the build command if so; otherwise both arguments are left off and the build is done locally.