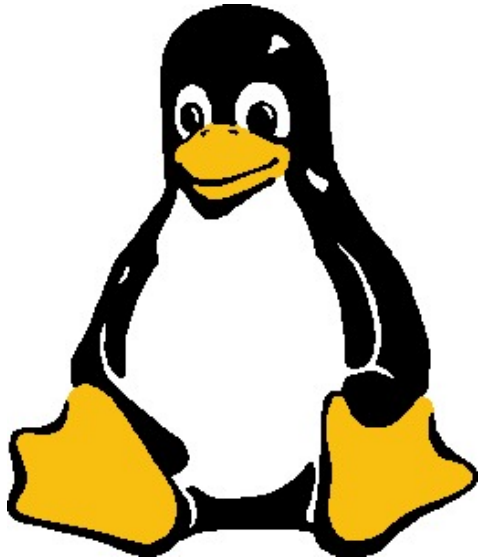
Linux Overview

- Though Cactus is cross-platform and may be developed in the developer's operating system of choice, Linux is the primary development environment for the Cactus framework. We will use a Linux environment for this tutorial.



- Linux was developed in 1991 by Linus Torvalds, after whom the operating system is named. The open-source philosophy underlying Linux inspires many software developers to license their software under the GPU (GNU Public License), allowing the software user access to the program source code as well as the right to modify and release forks of it.

Distributions

- For this reason, there are many varities of Linux called **distributions**. The major difference among distributions is often their package management system. The most popular distributions with original package management systems include:
  - Redhat
  - Debian
  - Gentoo
  - Slackware

Though Linux offers a feature-rich GUI with many options for desktop environments and window managers, it is nevertheless a command-driven system.

Linux has a mature command-line shell well-suited as a programming environment. One may write, compile, and debug programs from the CLI.

- Our OS is a minimalist Debian-based operating system that contains all the features required for Cactus development:

- Our OS uses a minimalist window manager called Fluxbox to ensure that it runs speedily inside of the virtual environment.

- We name our distribution CactusOS.
- We offer CactusOS as a virtual machine (VirtualBox) file.

- Our OS is a minimalist Debian-based operating system that contains all the features required for Cactus development:
  - Eclipse/Mojave

- Our OS uses a minimalist window manager called Fluxbox to ensure that it runs speedily inside of the virtual environment.

- We name our distribution CactusOS.
- We offer CactusOS as a virtual machine (VirtualBox) file.

- Our OS is a minimalist Debian-based operating system that contains all the features required for Cactus development:
  - Eclipse/Mojave
  - Cactus Computational Toolkit

- Our OS uses a minimalist window manager called Fluxbox to ensure that it runs speedily inside of the virtual environment.

- We name our distribution CactusOS.
- We offer CactusOS as a virtual machine (VirtualBox) file.

- Our OS is a minimalist Debian-based operating system that contains all the features required for Cactus development:
  - Eclipse/Mojave
  - Cactus Computational Toolkit
  - Einstein Toolkit

- Our OS uses a minimalist window manager called Fluxbox to ensure that it runs speedily inside of the virtual environment.

- We name our distribution CactusOS.
- We offer CactusOS as a virtual machine (VirtualBox) file.

- Our OS is a minimalist Debian-based operating system that contains all the features required for Cactus development:
  - Eclipse/Mojave
  - Cactus Computational Toolkit
  - Einstein Toolkit
  - Simfactory

- Our OS uses a minimalist window manager called Fluxbox to ensure that it runs speedily inside of the virtual environment.

- We name our distribution CactusOS.
- We offer CactusOS as a virtual machine (VirtualBox) file.

Virtualization

- Virtualization is used to run CactusOS within the working operating system.
- **Virtualization** is the process of emulating one operating system within another. The emulated OS is called the **guest OS**; the OS in which the emulation is done is the **host OS**.

Virtualization

- Virtualization is used to run CactusOS within the working operating system.
- **Virtualization** is the process of emulating one operating system within another. The emulated OS is called the **guest OS**; the OS in which the emulation is done is the **host OS**.
- **Virtualization software** is required to achieve this end.

Virtualization

- Virtualization is used to run CactusOS within the working operating system.
- **Virtualization** is the process of emulating one operating system within another. The emulated OS is called the **guest OS**; the OS in which the emulation is done is the **host OS**.
- **Virtualization software** is required to achieve this end.
- Examples of virtualization software packages include:

Parallels™

vmware®

VirtualBox

VirtualBox is:

- cost-free
- open-source
- cross-platform
- intuitive
- easily configurable

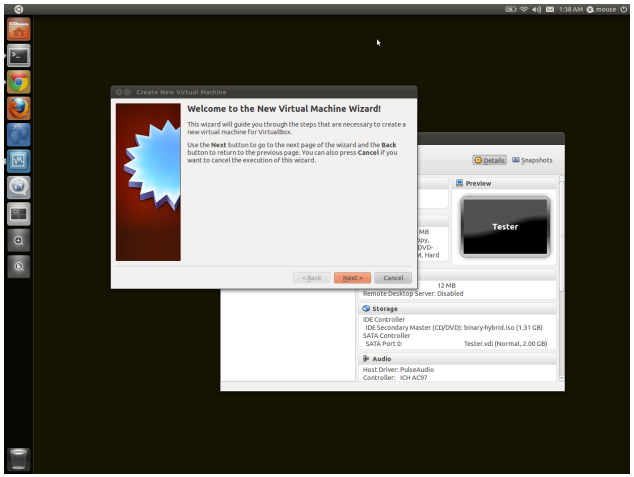
VirtualBox

VirtualBox is:

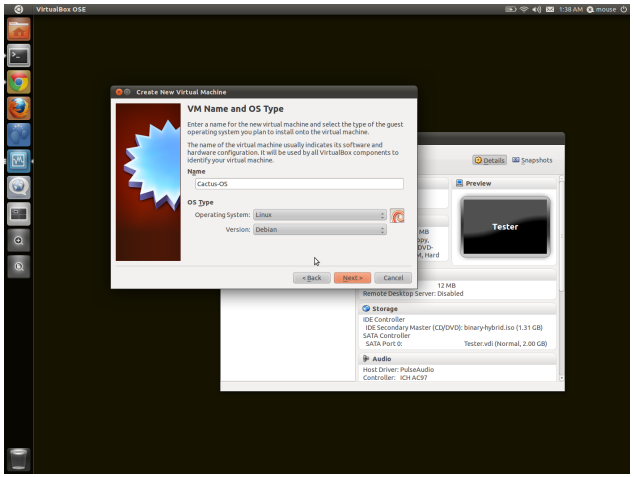
- cost-free
- open-source
- cross-platform
- intuitive
- easily configurable

Thus we choose it for our virtualization needs. In VirtualBox, the guest OS resides in a VBox machine file which is transferrable from one host OS to another.

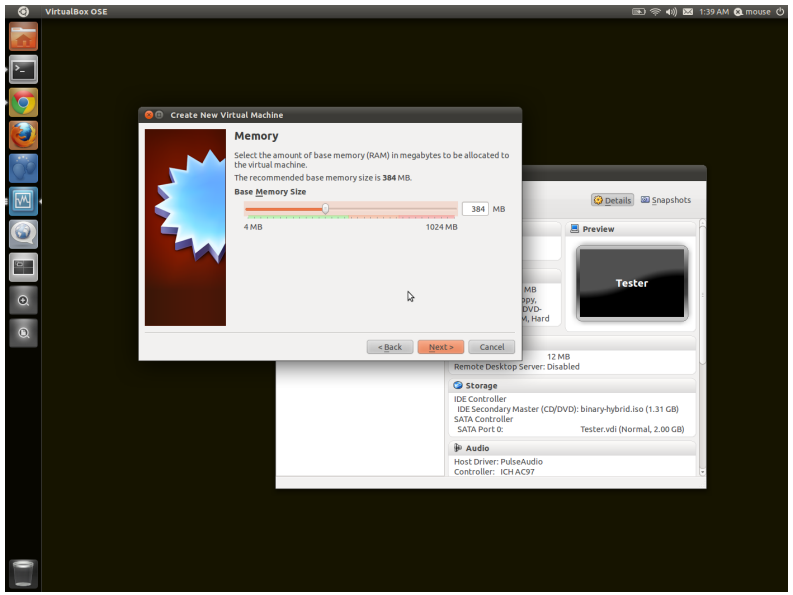
- You'll want to begin copying the .vdi file and the VirtualBox installer for your OS over from the provided DVD-ROM.



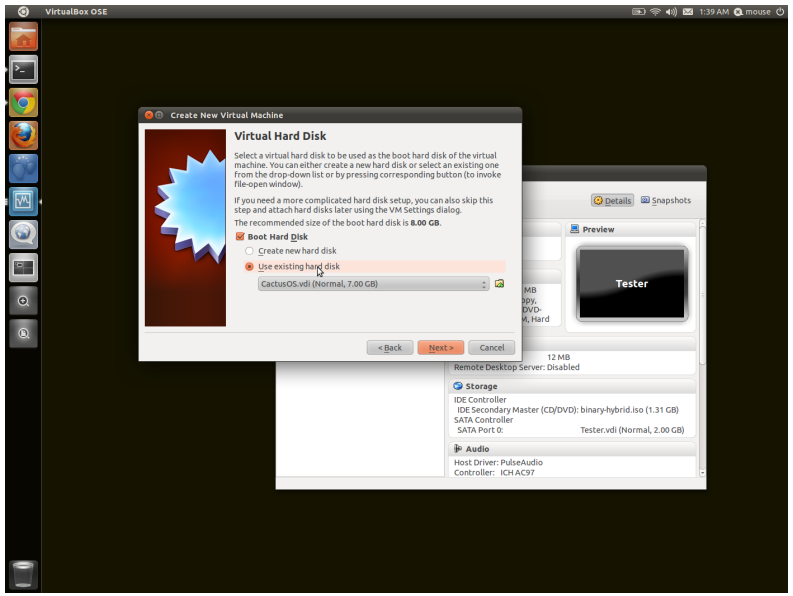
- Call the system ‘CactusOS’.
- It is a Debian-based Linux distribution.



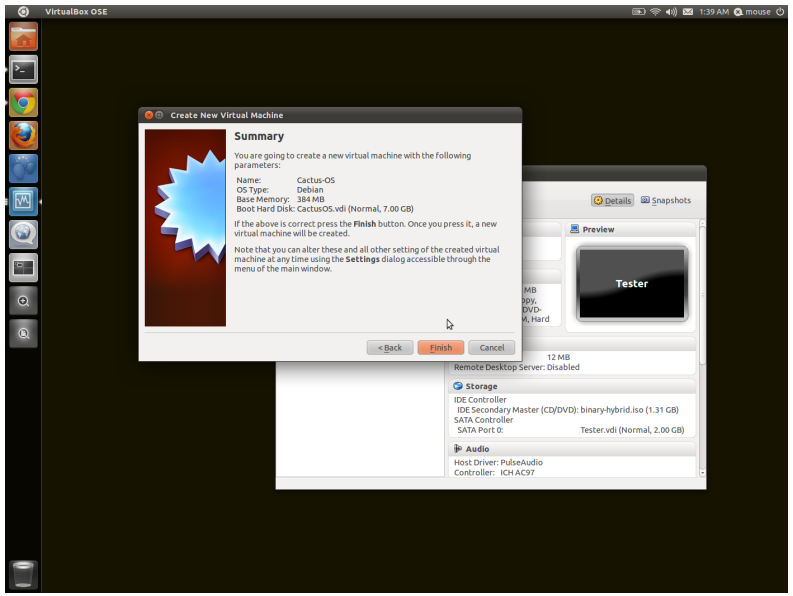
- Set the RAM equal to one-third to one-fourth of your total physical RAM.
- Setting the VM RAM too low may cause it to run out of available memory during the tutorial. However, setting it to high may cause it to slow down. VirtualBox requires RAM available to the host OS to support low-level processes which allow it to run.



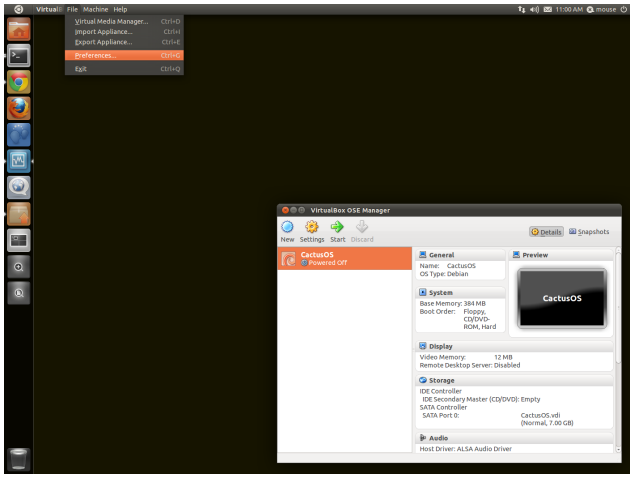
- Use the .vdi file from the DVD-ROM.

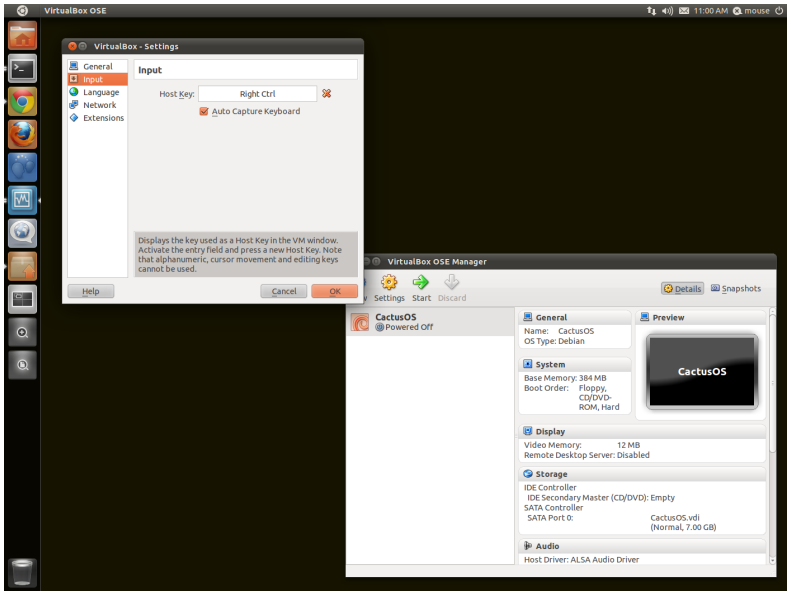


- Now click ‘Finish’. You may now run the Virtual OS by clicking the ‘Start’ button.



In order to escape the VirtualBox environment, you may press the VirtualBox Host Key. By default it is Right Ctrl. Since you may have to use the Ctrl key within the Linux enviroment, you may wish to change the Host Key. Under the VirtualBox menu, click ‘Machine’.





In the left-hand pane of the dialogue window that appears, click on ‘Input’. On the right-hand side, you may change the Host Key; simply click within the text window and hit the desired Host Key.

The Terminal

- During the tutorial, we will use a command-line interface (terminal).

The Terminal

- During the tutorial, we will use a command-line interface (terminal).
- On the desktop, click on the icon labelled ‘gterm’.

The Terminal

- During the tutorial, we will use a command-line interface (terminal).
- On the desktop, click on the icon labelled ‘gterm’.
- A terminal window will appear.

Essential Terminal Commands: Navigation and Files

pwd	present working directory	pwd
ls	list directory contents	ls [dir]
cd	change directory ('..' means up)	cd [dir]
cp	copy file/directory (-r)	cp [file/dir1] [file/dir2]
mv	move/rename file/directory	mv [file/dir1] [file/dir2]
rm	delete file/directory (-r)	rm [file/dir(s)]

Text File I/O

cat	print text file to screen	cat [file(s)]
head	print first n lines of file (default 10)	head -n [num] [file(s)]
tail	print last n lines of file (default 10)	tail -n [num] [file(s)]
nl	print text file to screen with line numbering	nl [file(s)]
less	print text file to screen, scrollable	less [file(s)]
nano	simple, intuitive text editor	nano [file(s)]
vim	full-featured programming text editor	vim [file(s)]

Text File I/O

cat	print text file to screen	cat [file(s)]
head	print first n lines of file (default 10)	head -n [num] [file(s)]
tail	print last n lines of file (default 10)	tail -n [num] [file(s)]
nl	print text file to screen with line numbering	nl [file(s)]
less	print text file to screen, scrollable	less [file(s)]
nano	simple, intuitive text editor	nano [file(s)]
vim	full-featured programming text editor	vim [file(s)]

- To navigate vim/less: h, j, k, l.
- Use i to insert in vim, ESC to quit insert mode.
- Use Ctrl+X to save/quit nano, ZZ to save/quit vim, and q to quit less.

Eclipse IDE

- Eclipse is a fully-featured IDE written in Java for development in Java, C++, and many more languages.
- Eclipse is open-source, cross-platform, and modular. Plug-ins may be written to extend its functionality.

Obtaining the Eclipse IDE

- Java and JDK 6 are required; these are installed on CactusOS.
- Note: This tutorial focuses on obtaining the Indigo version of Eclipse for compatability with the Mojave plugin.

Obtaining the Eclipse IDE

- Java and JDK 6 are required; these are installed on CactusOS.
- Note: This tutorial focuses on obtaining the Indigo version of Eclipse for compatability with the Mojave plugin.
  - Downloading and installing Eclipse:
  - http://www.eclipse.org/downloads/index.php
  - Select and download "Eclipse Classic 3.7".

Obtaining the Eclipse IDE

- Java and JDK 6 are required; these are installed on CactusOS.
- Note: This tutorial focuses on obtaining the Indigo version of Eclipse for compatability with the Mojave plugin.
  - Downloading and installing Eclipse:
  - http://www.eclipse.org/downloads/index.php
  - Select and download "Eclipse Classic 3.7".
    - Move to your Downloads folder and open the compressed file.
    - tar -xvvzf *gz
    - unzip *zip

Obtaining the Eclipse IDE

- Java and JDK 6 are required; these are installed on CactusOS.
- Note: This tutorial focuses on obtaining the Indigo version of Eclipse for compatability with the Mojave plugin.
  - Downloading and installing Eclipse:
  - http://www.eclipse.org/downloads/index.php
  - Select and download "Eclipse Classic 3.7".
    - Move to your Downloads folder and open the compressed file.
    - tar -xvvzf *gz
    - unzip *zip
    - Launch the Eclipse executable.
    - Click through the splash screen to create a workspace.

Obtaining the C/C++ Development Toolkit

- A C++ compiler is required; one is installed on your system.

Obtaining the C/C++ Development Toolkit

- A C++ compiler is required; one is installed on your system.
- Launch Eclipse.

Obtaining the C/C++ Development Toolkit

- A C++ compiler is required; one is installed on your system.
- Launch Eclipse.
- Click through Help – Install New Software – Add.

Obtaining the C/C++ Development Toolkit

- A C++ compiler is required; one is installed on your system.
- Launch Eclipse.
- Click through Help – Install New Software – Add.
  - (Indigo) http://download.eclipse.org/tools/cdt/releases/indigo

Obtaining the C/C++ Development Toolkit

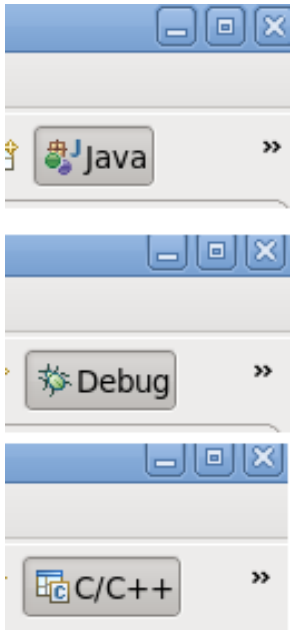
- A C++ compiler is required; one is installed on your system.
- Launch Eclipse.
- Click through Help – Install New Software – Add.
  - (Indigo) http://download.eclipse.org/tools/cdt/releases/indigo

Obtaining the C/C++ Development Toolkit

- A C++ compiler is required; one is installed on your system.
- Launch Eclipse.
- Click through Help – Install New Software – Add.
  - (Indigo) http://download.eclipse.org/tools/cdt/releases/indigo
  - Click CDT Main Features – Next – Next – Accept EULA and Finish.

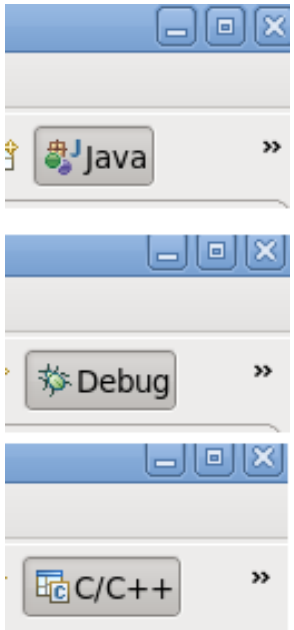
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.



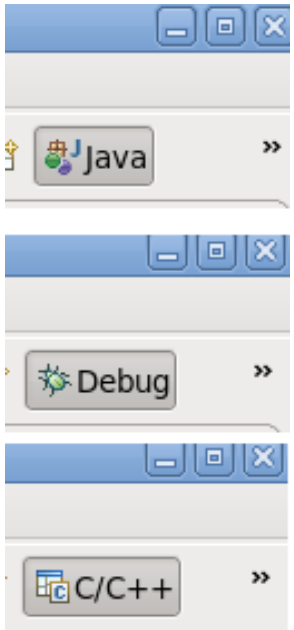
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.



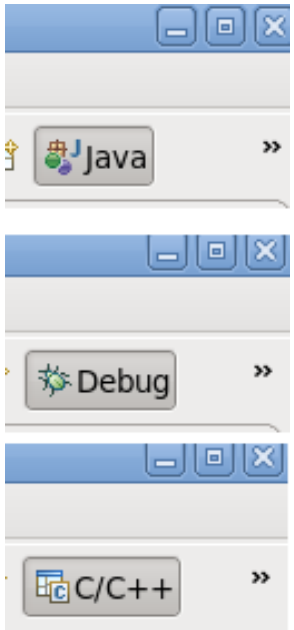
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.
- In addition to being customizable, Eclipse comes with several built in perspectives; for example:



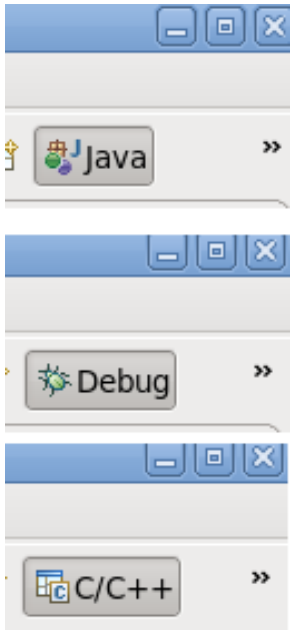
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.
- In addition to being customizable, Eclipse comes with several built in perspectives; for example:
  - Resource - shows various workspace resources irrespective of file extension



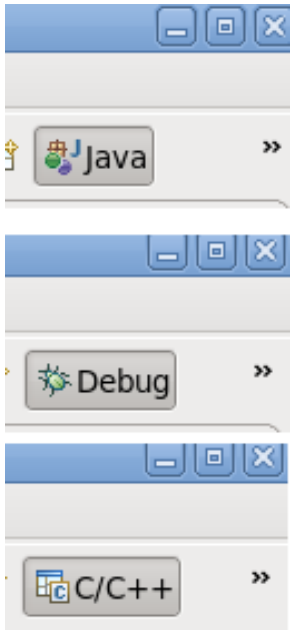
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.
- In addition to being customizable, Eclipse comes with several built in perspectives; for example:
  - Resource - shows various workspace resources irrespective of file extension
  - Java / Java Browsing - for Java development



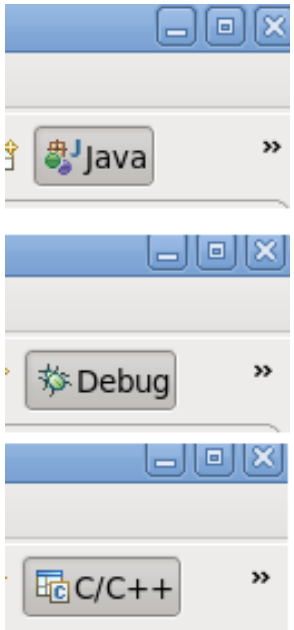
Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.
- In addition to being customizable, Eclipse comes with several built in perspectives; for example:
  - Resource - shows various workspace resources irrespective of file extension
  - Java / Java Browsing - for Java development
  - Debug - for debugging Java applications

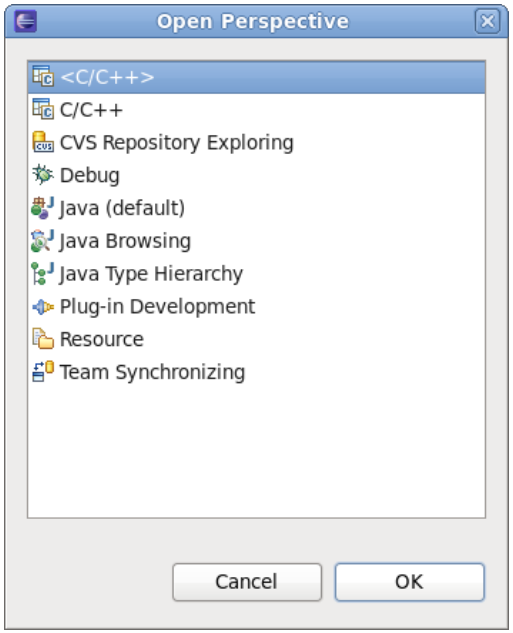


Views and Perspectives

- Eclipse Perspectives consist of views and editors associated with a specific task.
- The current perspective is indicated in the top right border of the Eclipse window.
- In addition to being customizable, Eclipse comes with several built in perspectives; for example:
  - Resource - shows various workspace resources irrespective of file extension
  - Java / Java Browsing - for Java development
  - Debug - for debugging Java applications
- And now that you have installed the CDT, you have the C++ development perspective.

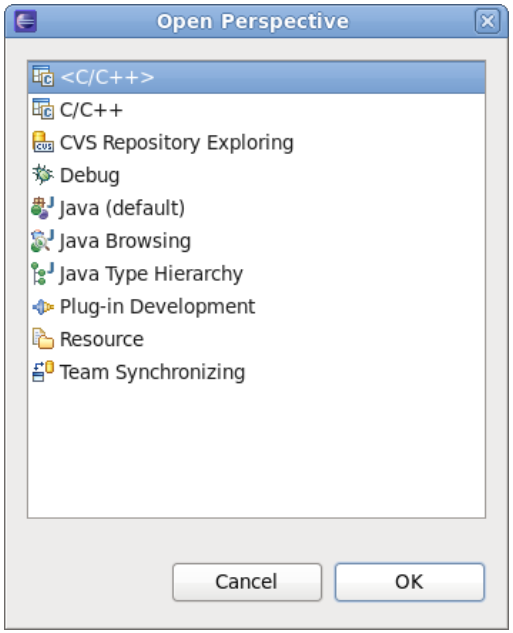


Views and Perspectives



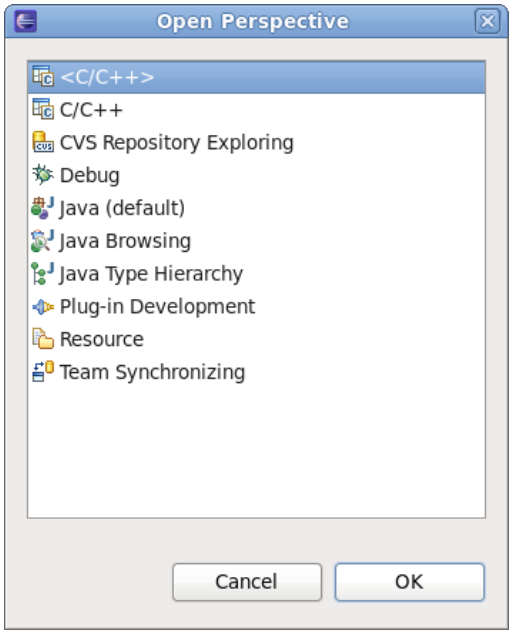
- To switch between perspectives:

Views and Perspectives



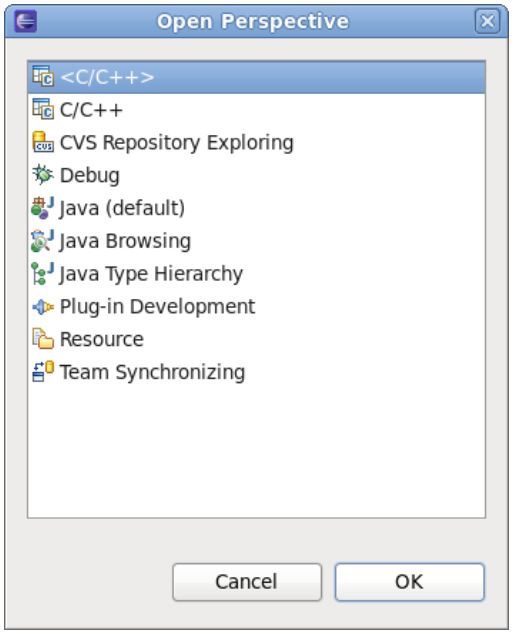
- To switch between perspectives:
  - Select Window from the drop down menu.

Views and Perspectives



- To switch between perspectives:
  - Select Window from the drop down menu.
  - Highlight Open Perspective and select Other.

Views and Perspectives



- To switch between perspectives:
  - Select Window from the drop down menu.
  - Highlight Open Perspective and select Other.
  - Select the appropriate perspective for your task.

Eclipse Hotkeys and Keyboard Commands

Ctrl-L	brings cursor to line number
Ctrl-F	find text
Ctrl-E	cycle through open files
Ctrl-Shift-F	re-format code
Ctrl-Shift-R	locate a file
Shift-Alt-Q C	open the console
Alt-$\leftarrow$	navigate back to previously-opened file
F3	go to type declaration
F5	refresh from disk
Left-Click & Drag	open multiple edit windows
Ctrl-Shift-L	open hotkey list

Hello World

- Select File – New – C++ Project.

Hello World

- Select File – New – C++ Project.
- Title your project and select Hello World C++ Project.

Hello World

- Select File – New – C++ Project.
- Title your project and select Hello World C++ Project.
- Click Run – Run.

Hello World

- Select File – New – C++ Project.
- Title your project and select Hello World C++ Project.
- Click Run – Run.
- Congratulations, you've run a program in Eclipse!

Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

The Eclipse IDE.

- is open-source,

Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

- is open-source,
- multi-platform,
- 

  The Eclipse IDE.

Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

- is open-source, multi-platform, and extensible;
- 

  The Eclipse IDE.

Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

- is open-source,
  multi-platform,
  and extensible;
  has an advanced editor

The Eclipse IDE.

Mojave: Eclipse Plugin for Cactus

Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

- is open-source,
  multi-platform,
  and extensible;
  has an advanced editor
  featuring multiple layouts

The Eclipse IDE.

Mojave: Eclipse Plugin for Cactus

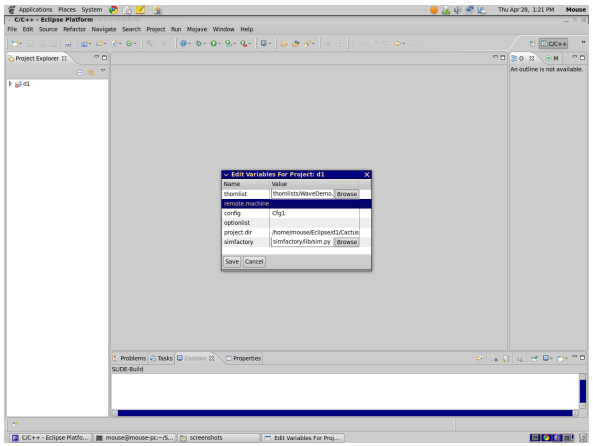
Mojave is an Eclipse plug-in for Cactus. The Mojave plug-in consists of the Mojave menu, which interfaces to several scripts that automate the process of configuring, compiling, and running Cactus. Eclipse is chosen as the because it:

- 

The Eclipse IDE.
- is open-source,
multi-platform,
and extensible;
has an advanced editor
featuring multiple layouts
and an SVN interface.

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!

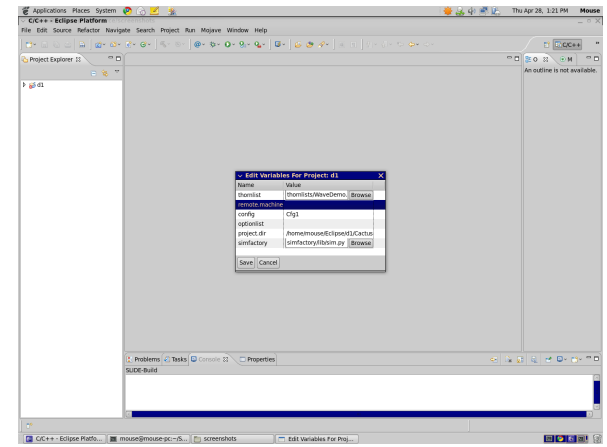


Mojave.

- edit variables

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!

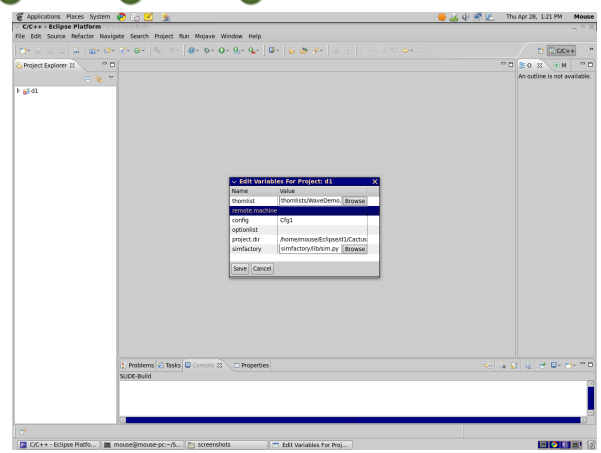


Mojave.

- edit variables
  build (remotely)

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!

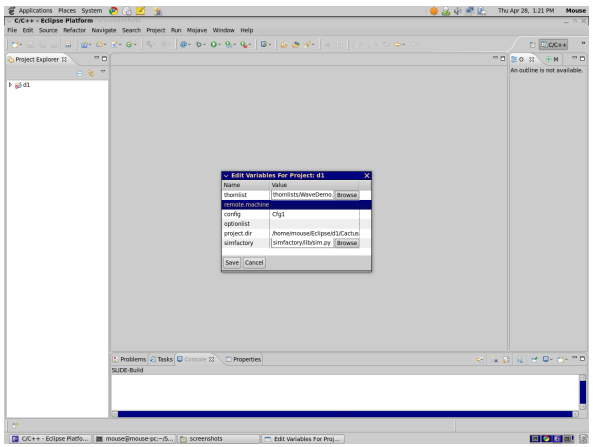


Mojave.

- edit variables
  build (remotely)
  clean build

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!

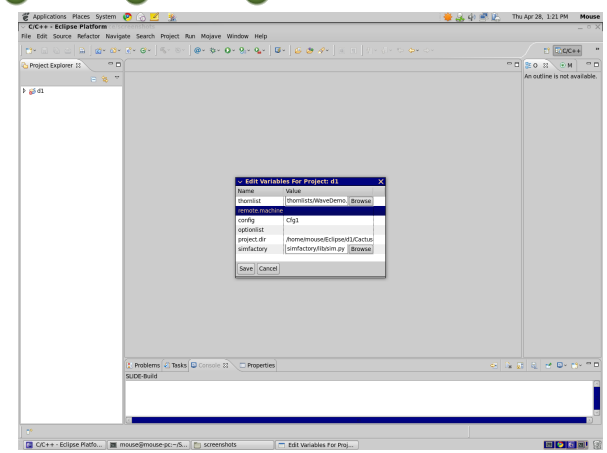


Mojave.

- edit variables
  build (remotely)
  clean build
  run

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!

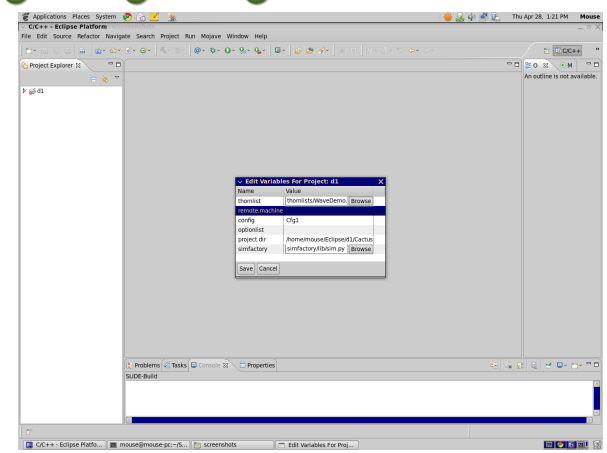


Mojave.

- edit variables
  build (remotely)
  clean build
  run
  create

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!

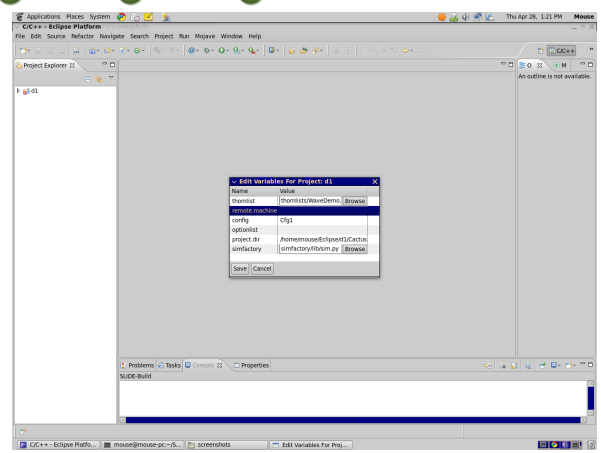


Mojave.

- edit variables
  build (remotely)
  clean build
  run
  create
  update

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!

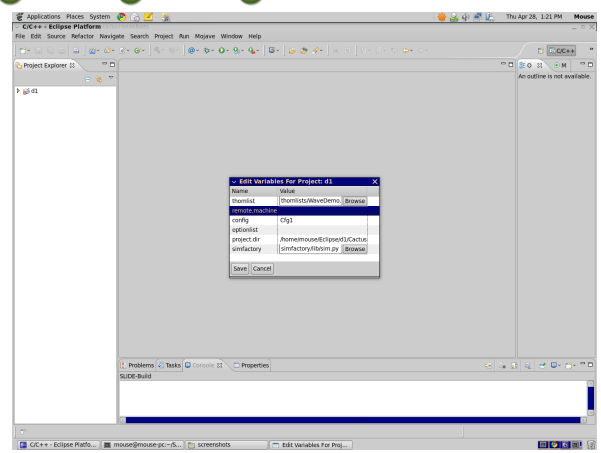


Mojave.

- edit variables
  build (remotely)
  clean build
  run
  create
  update
  make thorn

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!

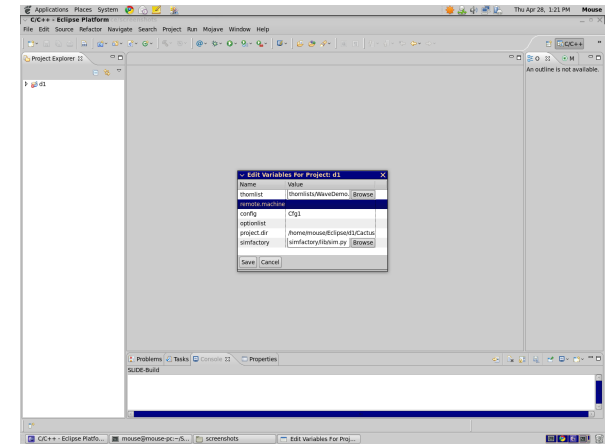


Mojave.

- edit variables
  build (remotely)
  clean build
  run
  create
  update
  make thorn
  clean runs

Mojave: Remote Build

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.

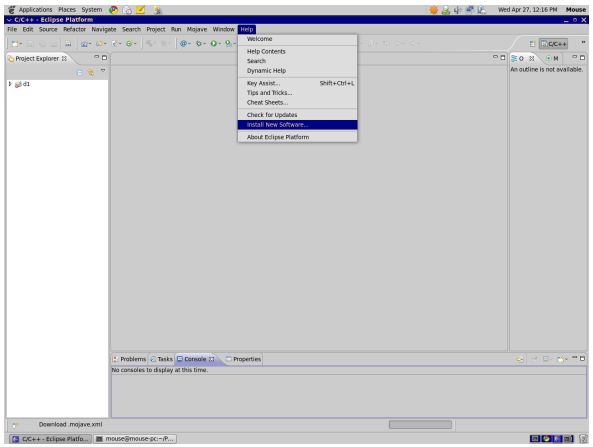


Mojave.

- edit variables
- build **remotely**
- clean build
- run
- create
- update
- make thorn
- clean runs

Mojave: Installation

The Mojave plugin is easy to install. Simply:

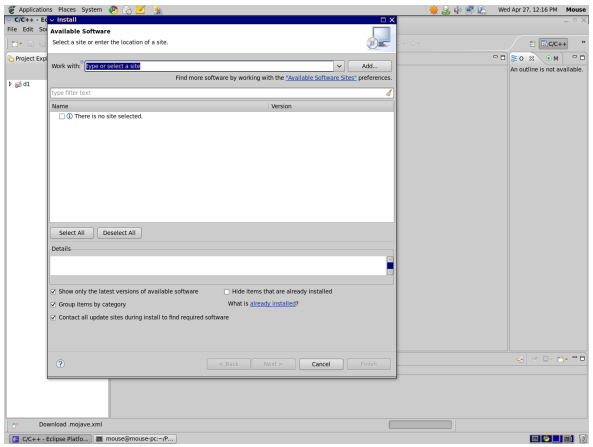


Step 1.

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

Mojave: Installation

The Mojave plugin is easy to install. Simply:

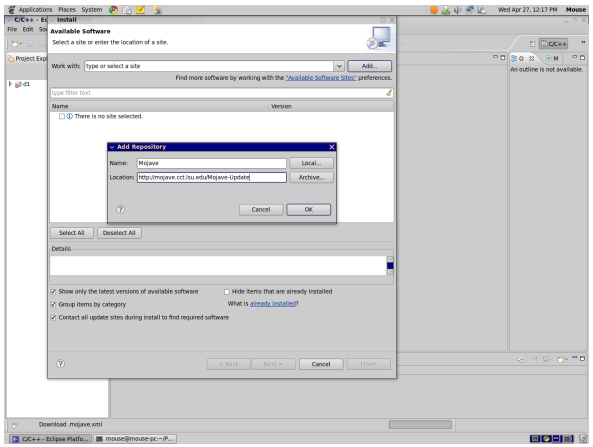


Step 2.

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

Mojave: Installation

The Mojave plugin is easy to install. Simply:

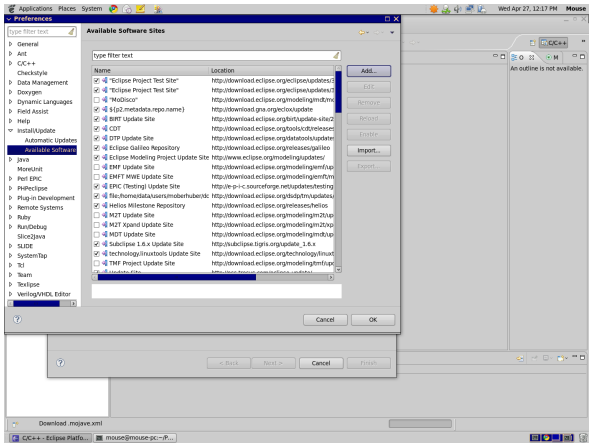


Step 3.

- 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.

Mojave: Installation

The Mojave plugin is easy to install. Simply:

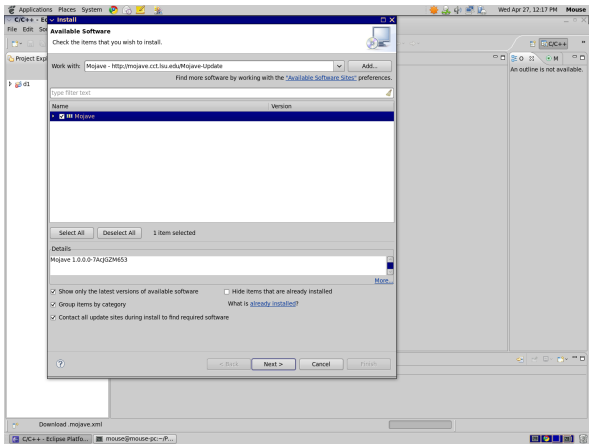


Step 4.

- Click the Available Software Sites link. In the dialogue window which opens, mark all checkboxes.

Mojave: Installation

The Mojave plugin is easy to install. Simply:

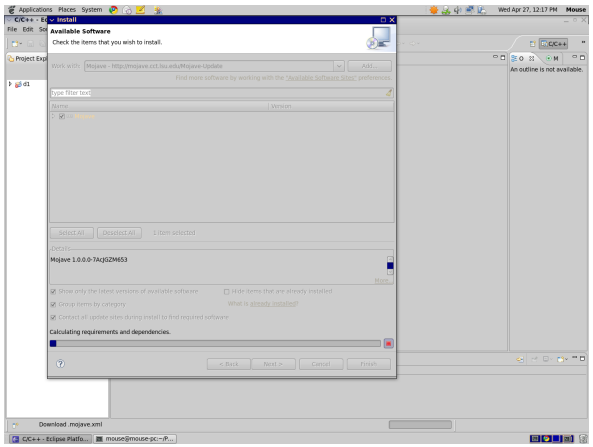


Step 5.

- 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**.

Mojave: Installation

The Mojave plugin is easy to install. Simply:



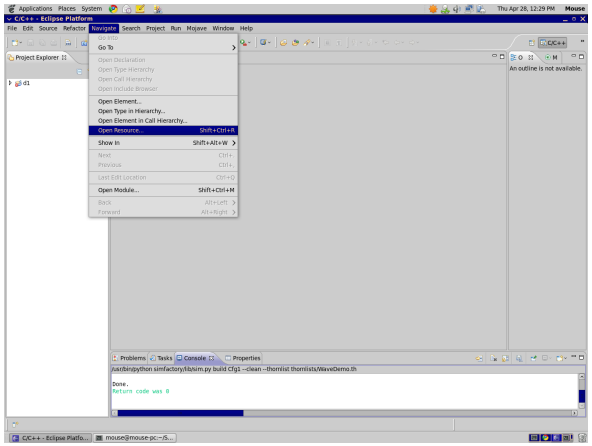
Step 6.

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

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:

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

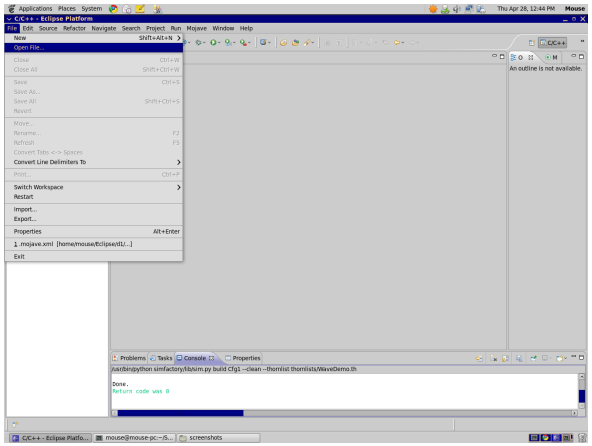


Finding .mojave.xml.

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** ...

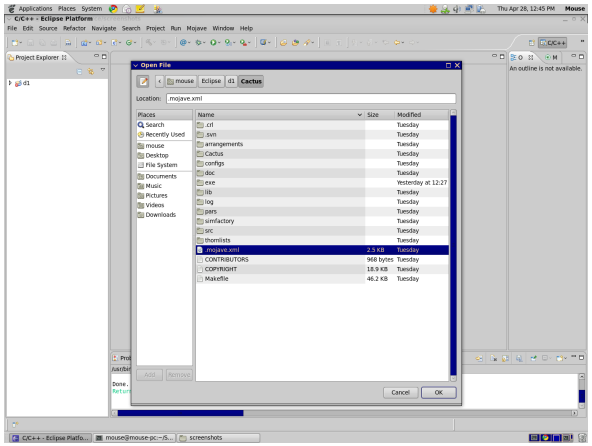


Finding .mojave.xml.

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** . . . then open *workspace-name/project-name*/Cactus/.mojave.xml in your home directory.

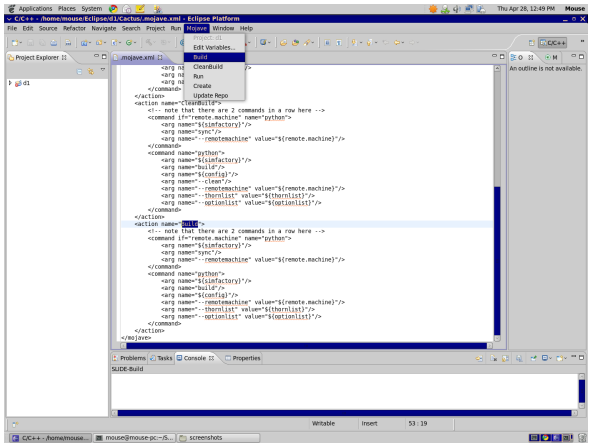


Finding .mojave.xml.

Mojave: Development

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.

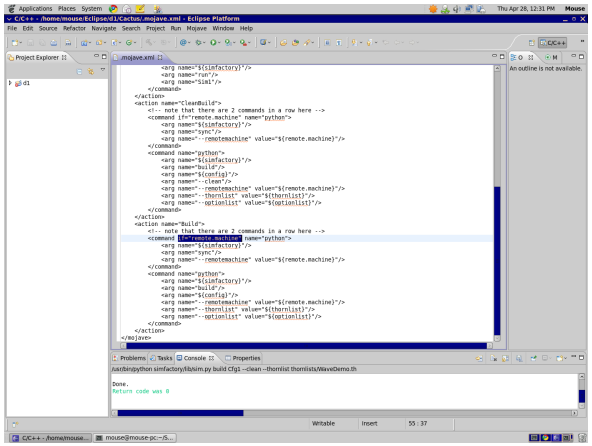


Structure of .mojave.xml.

Mojave: Development

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.

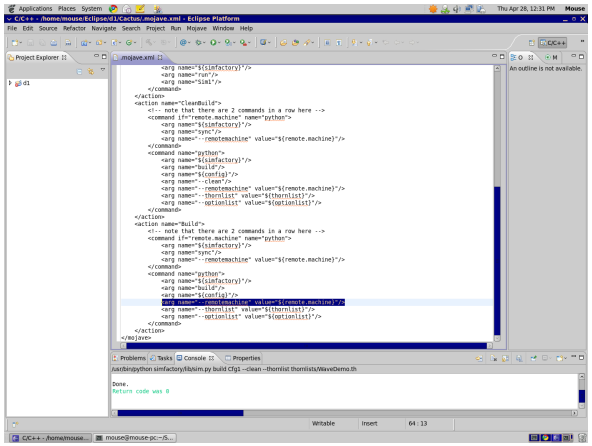


Structure of .mojave.xml.

Mojave: Development

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.



Structure of .mojave.xml.