# Table of Contents

1. Overview ........................................................................................................... 1  
   Cadena .................................................................................................................. 1  
2. Prepare ............................................................................................................... 3  
   Introduction .......................................................................................................... 3  
   Install: Java .......................................................................................................... 3  
   Install: Eclipse ..................................................................................................... 3  
   Conclusion ............................................................................................................. 4  
3. Installing Cadena ................................................................................................. 5  
   Introduction .......................................................................................................... 5  
   Launch the Update Manager .................................................................................. 5  
   Configure the Update Manager with Cadena Features ....................................... 6  
   Configure the Update Manager to use the Eclipse update site ......................... 7  
   Configure the Update Manager to use the Alternate Eclipse update site .......... 7  
   Select Cadena distribution .................................................................................... 9  
   Add Dependencies ............................................................................................... 10  
   Accept License Agreements ................................................................................ 11  
   Final Confirmation .............................................................................................. 12  
   Unsigned JAR Warning ....................................................................................... 12  
   Conclusion ............................................................................................................ 13  
4. Installing Platform Plugins .................................................................................. 14  
   Introduction .......................................................................................................... 14  
   Installing a Platform: TinyOS ............................................................................... 14  
   Launch the Update Manager ................................................................................ 14  
   Configuring Update Sites .................................................................................... 14  
   Selecting the Platforms to Install ....................................................................... 14  
   Accept License Agreements ................................................................................ 15  
   Final Confirmation .............................................................................................. 16  
   Conclusion ............................................................................................................ 16  
5. Common Problems .............................................................................................. 17  
   Eclipse Download Not Found .............................................................................. 17  
   EMF and GEF Downloads Not Found ................................................................... 17  
   Wrong Java Version ............................................................................................. 17  
6. Tips ...................................................................................................................... 18  
   Tips on Installing Cadena ..................................................................................... 18  
   Setting your JAVA_HOME Environment Variable .............................................. 18  
Glossary .................................................................................................................... 20  
Bibliography ............................................................................................................ 23
List of Figures

1.1. The Cadena meta-modeling language ................................................................. 2
3.1. Eclipse Update Manager Menu ............................................................................. 5
3.2. Eclipse Update Manager Dialog .......................................................................... 6
3.3. New Remote Site: Cadena .................................................................................... 7
3.4. New Remote Site: Eclipse ..................................................................................... 8
3.5. Update Site Selection ............................................................................................ 9
3.6. Cadena Feature Selection ..................................................................................... 10
3.7. Select Required Plugins ....................................................................................... 11
3.8. Accept License Agreements ................................................................................ 12
3.9. Confirm Install ..................................................................................................... 12
3.10. Confirm Install .................................................................................................... 13
4.1. TinyOS Platform Feature Selection ..................................................................... 15
4.2. Accept TinyOS License Agreement ...................................................................... 15
4.3. Confirm TinyOS Platform Install ......................................................................... 16
Chapter 1. Overview

The Cadena 2.0: Install Guide was created to be a guide to installing Cadena as well as the Platform plugins that extend Cadena. The guide starts with a little background information as well as some pointers to more details about those topics. It then continues by walking you through the details of installing Cadena. This guide also includes some tips and tricks that we have collected that are related to the install process.

Cadena

Cadena is an Eclipse-based extensible integrated modeling and development framework for component-based systems. Cadena's models are type-centric in that multi-level type systems are used to specify and enforce a variety of architectural constraints relevant to development of large-scale systems and software product lines.

Cadena provides the following capabilities to system architects, infrastructure developers, and system developers:

- Define modeling environments for widely-used component models: Cadena's meta-modeling capabilities can be used to formally capture the definition of widely used component models such as the CORBA Component Model (CCM), Enterprise Java Beans (EJB), and nesC (a component model for sensor networks built on TinyOS). Meta-models can include attributes that represent settings and parameters for underlying middleware frameworks on which systems will be deployed.

- Define domain-specific component models: Cadena meta-modeling can also be applied to specify new component models, including domain-specific component models that are tailored to the characteristics of a particular domain or underlying middleware capabilities.

- Flexibly combine and extend multiple component models in a single system: Cadena meta-models (called styles) can be directly manipulated using style operations. This provides a variety of powerful and useful capabilities to system architects.

- Styles can be extended through inheritance. This enables reuse of meta-model definitions, and facilities refinement of platform definitions (multi-step platform-independent to platform-specific model refinement).

- Multiple styles can be combined within the same architecture model environment to support development of systems of systems that incorporate multiple component models.

- Define end-to-end model-driven development environments: Cadena's base set of capabilities can be extended using plug-in mechanisms based on the Eclipse plug-in architecture. This enables infrastructure developers to build end-to-end model-driven development environments that include facilities for editing component implementations, model-level configuration of middleware capabilities, code generation, simulation, verification, and creating system builds. Plug-ins can also be developed to link other development tools including tools for requirements capture and down-stream class-level modeling tools such as Rational Rose or Modeler or iLogix Rhapsody.
Figure 1.1. The Cadena meta-modeling language
Chapter 2. Prepare

Introduction

Cadena is built upon several other technologies which must be installed prior to installing Cadena. This chapter will walk you through that list of requirements and help you get those installed properly.

Cadena is built using the Java programming language and is a collection of plugins for the Eclipse platform. This means that a Java Runtime Environment (JRE) must be available and the Eclipse platform must be installed and using that JRE (the Java Development Kit (JDK) might work better than the JRE in the long run). The following sections will walk you through installing Java and Eclipse.

Install: Java

Cadena was built using many of the new features available in the Java platform so the JRE (or JDK) used must be at least 5 or higher.

A Java VM can be downloaded from Sun's Java website (http://java.sun.com). That site also contains instructions to help you install on your chosen platform (architecture, operating system, and windowing system).

Once you complete the installation you should make note of the directory in which it was installed. You will need this once you install Eclipse.

Install: Eclipse

At this time, Cadena only functions in Eclipse version 3.2* and 3.3*.

The Eclipse SDK is available through the Eclipse download site (http://www.eclipse.org/downloads/) as well as mirrored on our download site (http://projects.cis.ksu.edu/plugin/ksuafrs/showfiles.php?group_id=7). You will notice that we only provide a mirror for some of the available platforms. For a complete list you should visit the Eclipse download site (or one of its mirrors).

To install Eclipse you simply need to un-archive the file that you downloaded into an appropriate location. For example, you may want to install it in on your Linux/GTK machine in your home directory (e.g., /home/myUser). Using tar you can run `tar zxf eclipse-SDK-3.3.0-gtk.tar.gz` from your home directory. This will create an `eclipse` directory (e.g., `/home/myUser/eclipse` will be created) that contains the Eclipse installation files.

Once you have it installed, you will need to decide where you want your workspace to be stored. To continue our example, you might want to create a directory in your home directory named workspace (mkdir /home/myUser/workspace). When you start up Eclipse, you can now choose this directory when prompted. You can also specify it on startup with the `-data` flag (e.g., `-data /home/myUser/workspace`).

You might also consider passing the path to the JDK to the Eclipse binary. This is done with the `-vm` flag. For example, if you are on Linux and have the JDK installed in /opt/java5 your command line could include `-vm /opt/java5/bin/java`.

Eclipse also provides a command line argument that allows you to pass Java VM arguments through to the underlying JVM. This comes in handy when working with large models in Cadena so that you can increase the amount of memory available to Eclipse/Cadena. The flag is named `-vmargs`. For example, if
you wanted to set the maximum heap size to 512 MB you could pass the following argument to Eclipse:
\[ -vmargs -Xmx512M. \]

Now that you have Eclipse installed you are ready to run it. Combining the flags mentioned before you
would run \texttt{eclipse -vm /opt/java5/bin/java -data /home/myUser/workspace -vmargs -Xmx512M}.

Once you have Eclipse installed and running you will need to install two extra features available for
Eclipse: 1) EMF and 2) GEF. We will guide you through that process when we install Cadena using the
Eclipse Update Manager.

\section*{Conclusion}

Once you complete the installation of Java and Eclipse you are ready to install Cadena and any requested
Platform plugins. To do that, see Chapter 3, \textit{Installing Cadena} and Chapter 4, \textit{Installing Platform Plugins}. 
Chapter 3. Installing Cadena

Introduction

Cadena is a set of plugins that extend the functionality of the Eclipse platform. You should already have completed the installation of Java and Eclipse before starting this chapter (this is detailed in the section called “Install: Java” and the section called “Install: Eclipse”).

In this chapter you will install the Cadena plugins (or features) as well as the dependencies using the Eclipse Update Manager. The following sections will walk you through this process.

Launch the Update Manager

Eclipse features an elegant mechanism for installing new plug-ins: the Update Manager. By configuring third-party websites as known sources of new software inside the Update Manager, you can allow Eclipse to manage the dependencies, downloading, unpacking, and installation of these sets of packages (called features).

Open the Update Manager’s dialog to install new features by choosing Help -> Software Update -> Find and Install... from the main menu. This can be seen in Figure 3.1, “Eclipse Update Manager Menu”.

Figure 3.1. Eclipse Update Manager Menu

The Update Manager can, in addition to downloading new packages, also search for upgrades to already-installed features. For now, choose Search for new features to install in the dialog that appears. This dialog can be seen in Figure 3.2, “Eclipse Update Manager Dialog”.

Figure 3.2. Eclipse Update Manager Dialog
Figure 3.2. Eclipse Update Manager Dialog

The other option (Search for updates to the currently installed features) can be used to update to the latest release of Cadena if you already have a previous version installed.

Configure the Update Manager with Cadena Features

Depending on your prior configuration, Eclipse will already have one or more external websites declared as sources of packages. At a minimum, you should see the main Eclipse Project Updates and Callisto Discovery Site entries. This contains official Eclipse add-on projects. We will now inform the Update Manager of the website where it can find Cadena features.

First, click New Remote Site.... In the resulting dialog box, enter the following values:

Name    Cadena update site
URL    http://cadena.projects.cis.ksu.edu/update

This dialog can be seen in Figure 3.3, “New Remote Site: Cadena”.
Installing Cadena

Figure 3.3. New Remote Site: Cadena

Click OK to dismiss the dialog. The Cadena site should now appear as a checkbox-selectable item in the Sites to include in search list.

Configure the Update Manager to use the Eclipse update site

Cadena depends on two plugins that may not be installed in Eclipse by default: 1) GEF and 2) EMF. Because of this, you should select the Callisto Discovery Site as a site to include in the search of plugins and features to install. If you are using Eclipse 3.1* you may need to use an alternate update site (details in the section called “Configure the Update Manager to use the Alternate Eclipse update site”.

Once you have the Cadena update site specified and selected and the Callisto Discovery site selected press Finish. When you do this, Eclipse starts the process of downloading and inspecting the update site contents to see what plugins and features are available.

Configure the Update Manager to use the Alternate Eclipse update site

If you are installing Cadena in Eclipse 3.1*, you may need to use an alternate update site to get the proper versions of GEF and EMF installed. We have found that these versions of EMF and GEF are not available on the main Eclipse update site. Therefore, it will work best for users to use an alternate update site. There is also a note in the section called “EMF and GEF Downloads Not Found” that talks about alternate ways to get GEF and EMF.

First, click New Remote Site.... In the resulting dialog box, enter the following values:

Name Alternate Eclipse.org update site
URL http://update.eclipse.org/updates/3.0

This can be seen in Figure 3.4, “New Remote Site: Eclipse”.

**Figure 3.4. New Remote Site: Eclipse**

Click OK to dismiss the dialog. The alternate Eclipse update site should now appear as a checkbox-selectable item in the Sites to include in search list.

After adding the Cadena and the alternate Eclipse update sites to the Sites to include in search list, select the Alternate Eclipse.org update site and the Cadena update site to indicate that some packages from each of these sources will be installed. This dialog can be seen in Figure 3.5, “Update Site Selection”. Now click Finish and the Update Manager will begin querying these websites to find their current software offerings.
Select Cadena distribution

Once the Update Manager has completed its inspection of the specified update sites it will provide you a list of available features and plugins to install. This is displayed in the Search Results dialog. By expanding the tree, you can view all of the software available on the Cadena update site.

The Core category will always contain the most recent release of Cadena. It may also include other milestone releases of particular noteworthiness and other extensions to the core platform. The Platforms category contains releases of platform extensions to Cadena (the platforms shown will likely include CCM, OpenCCM, and TinyOS).

Choose Cadena (the current release may be more current than the screen capture pictured). Optionally, you may also select any other extensions or platforms to install as well. This dialog can be seen in Figure 3.6, “Cadena Feature Selection”.

Figure 3.5. Update Site Selection
Figure 3.6. Cadena Feature Selection

![Image of Cadena Feature Selection](image)

Note: Eclipse may require several seconds to process the expansion of a site or category or the selection of a feature. When this expansion happens, the delay is caused when the Update Manager downloads and inspects the JAR files which specify the features available. Please be patient and wait for the user interface to again be responsive.

After selecting these checkboxes, you may get an error that the Update Manager cannot be installed because features Cadena depends are not installed. If this is the case, the next step will fix this by selecting required features.

**Add Dependencies**

Cadena depends on a few features (including GEF and EMF) to be installed before it can be used. If these features are not installed, the Update Manager will display a warning.

To install these required features along with the Cadena installation, click on the Select Required button. This will automatically select all of the necessary features. Note: due to a bug in Eclipse, it may be necessary to "prime" the feature list by expanding the Eclipse.org item before clicking the Select Required button. Otherwise, the required features will not be included in the selection. This dialog can be see in Figure 3.7, “Select Required Plugins”.
Figure 3.7. Select Required Plugins

Before moving on you should ensure that you have the following features selected to be installed.

- Cadena 2.0.*
- Cadena Jython 2.0.*
- Eclipse Modeling Framework (EMF) (2.1.0 for Eclipse 3.1*)
- Graphical Editing Framework (GEF) (3.1.1 for Eclipse 3.1*)

After selecting those features, click Next to move on.

Accept License Agreements

The SAnToS Lab provides its software under the SAnToS Academic License. Before completing the install, you must agree to its stipulations. In addition, the Eclipse Foundation has its own open-source license which applies to any features installed that come from Eclipse.org. Other library dependencies of Cadena which we bundle in the Cadena core distribution feature also impose their own license terms.

The Feature License screen (seen in Figure 3.8, “Accept License Agreements”) presents all of these documents. Please read through all of these terms of agreement and, after you are familiar with them to your satisfaction, click Next.
Final Confirmation

Eclipse’s update manager offers one final chance to review your choices (as seen in Figure 3.9, “Confirm Install”). Click Finish to trigger installation of the features.

Figure 3.9. Confirm Install

Unsigned JAR Warning

While installing the Cadena feature, you may be presented with a warning which reports that no encryption signature has been used to certify its authenticity. You should still allow installation to proceed despite this condition by clicking Install All. An example of this dialog can be seen in Figure 3.10, “Confirm Install”.
Figure 3.10. Confirm Install

![Figure 3.10. Confirm Install](image)

Conclusion

At this point, all software should be unpacked and installed. You will probably be asked to restart the workbench to make the new plugins active. After doing this, Cadena will be ready to use.
Chapter 4. Installing Platform Plugins

Introduction

One of the convenient features of the Cadena project is that each target platform is provided as a plugin (or specifically, a feature) for Eclipse. Some example target platforms include:

- OpenCCM (Platform/OpenCCM) Platform for development of CCM applications deployed using OpenCCM.
- TinyOS (Platform/TinyOS) Platform for development of TinyOS applications written in the nesC programming language.

This means that when you want to work with a specific target platform, you can simply install it (and its dependencies) and forget about the other platforms that are available. For example, developers of TinyOS applications have no need for the OpenCCM platform plugin so they don't have to install it.

This section walks you through some details on installing platform plugins and provides some tips-n-tricks that might be useful.

Installing a Platform: TinyOS

This section will walk you through the steps required to install a platform plugin (or feature) that will extend Cadena's abilities. We will use the TinyOS platform as an example but any platform will require similar steps to install successfully.

This process is very similar to the initial installation of Cadena. The only difference is what features you select. During the install, you will start the Update Manager, search for features to install, select the platform and dependencies, accept the license agreement, and confirm the install.

Launch the Update Manager

In the same way you started the Update Manager in the section called “Launch the Update Manager”, you should select the Help -> Software Update -> Find and Install... menu items to start. An example of this dialog is shown in Figure 3.1, “Eclipse Update Manager Menu”.

And, just like the previous installation instructions, you will choose to Search for new features to install in the dialog presented (as seen in Figure 3.2, “Eclipse Update Manager Dialog”).

Configuring Update Sites

If necessary, you can now configure any new updates sites that you want to query for updates. But in most cases, you will only need the Cadena update site (previously created in the section called “Configure the Update Manager with Cadena Features”) to install platforms. This is true in the case of the TinyOS platform. Select the Cadena update site and press Finish.

Selecting the Platforms to Install

As before, the Search Results dialog shows the set of all features found on the requested update sites. By expanding the tree, you can view all of the software available on the Cadena update site.

When installing platforms, the Core category can be ignored. Instead, you will want to explore the Platforms portion of the tree to find the platform that you wish to install. In this case, you are looking for
the TinyOS platform. Select it from the list and proceed with the install (the dialog will look like Figure 4.1, “TinyOS Platform Feature Selection”). In some cases, you may need to also add dependencies. Using the Select Required button of this dialog can be very useful in resolving these connections. For more detail on this, see the section called “Select Cadena distribution” that was previously discussed in this guide.

**Figure 4.1. TinyOS Platform Feature Selection**

![TinyOS Platform Feature Selection](image)

**Accept License Agreements**

Each feature selected for installation will have its own license agreement defined. Before proceeding through the Update Manager, you must read and agree to the license terms specified. In the case of the TinyOS platform, you must agree to the SAnToS Academic License. This dialog will look similar to Figure 4.2, “Accept TinyOS License Agreement”.

**Figure 4.2. Accept TinyOS License Agreement**

![Accept License Agreement](image)

After reading the license agreement select the I accept radio button and press the Next button to continue installing the selected platform.
Final Confirmation

Before Eclipse completes the installation, it will prompt you to confirm the fact that it is about to install the previously selected features. Double-check your selections and the installation location and when ready, press the Finish button to complete the install process (as seen in Figure 4.3, “Confirm TinyOS Platform Install”).

Figure 4.3. Confirm TinyOS Platform Install

Note: As before (in the section called “Unsigned JAR Warning”), you may be prompted with a warning about unsigned jars. You should accept this and press the Install All button.

Conclusion

You have now successfully installed a platform and its dependencies. You may be prompted to restart Eclipse. If so, restart and Cadena will be ready to help you develop applications for the newly installed platform.

Note: For some platforms, more detailed installation is required. For more on that see the associated platform guides. For example, the OpenCCM platform requires access to an OpenCCM installation on your local machine. This is a very complex process so the Cadena 2.0: OpenCCM Tutorial provides more guidance on this.
Chapter 5. Common Problems

While installing and using Cadena there are many problems that can arise. The following sections provide some common problems and their solutions.

Eclipse Download Not Found

Sometimes the Eclipse.org download site is slow or it is hard to find the correct version of Eclipse (3.1.2) to use with Cadena. Because of this, we provide a mirror for a few of the available platforms upon which Eclipse will run. So when you can't find them on Eclipse.org, look through our mirror to see if we have what you will need.

EMF and GEF Downloads Not Found

Sometimes the Alternate Eclipse.org update site is down and the mirrors don't contain the correct versions of EMF and GEF. One solution to this dilemma is to use the mirrored EMF and GEF downloads that we provided on our download site. You will simply need to download the following archives and unarchive them in your Eclipse installation directory.

- GEF-SDK-3.1.1.zip
- emf-sdo-runtime-2.1.0.zip

Wrong Java Version

It is important to install the correct version of the Java VM (either JRE or JDK). If this isn't done, there are some error messages that will occur. When you see these, you should double-check the version of Java that Eclipse is using to make sure it meets the system requirements specified in the section called “Install: Java”.

One of the errors that you might come across is described in Bug #828 in the Cadena projects Bug Tracker (title: Inability to start new Cadena or OpenCCM Projects.). The error message was "Plug-in edu.ksu.cis.cadena.platform.openccm was unable to load class edu.ksu.cis.cadena.platform.openccm.NewOpenCCMProjectWizard." while trying to create a new project. This happens because Cadena uses features of Java 5 and the user was using Java 4.
Chapter 6. Tips

Cadena relies upon several different technologies. Therefore, a user of Cadena must have a good breadth of knowledge to be able to work through any problems that might be encountered. In this chapter we will provide some common tips that will make our users more effective.

Tips on Installing Cadena

The following are problems that may be encountered while installing Cadena and possible solutions to those problems.

• **Eclipse is unable to find updates in the primary Eclipse update site**

  We encountered this problem while trying to install Cadena in the "updates" window and attempting to use the Select Required button to automatically select the plugin dependencies required by Cadena from the Eclipse update site. After pushing the Select Required button, nothing happened and the dependency error alert remained. If you encounter this problem, try using this workaround alternative site in place of the default "Eclipse update site" [http://update.eclipse.org/updates/3.0].

• **After completing the Cadena install, only 2 plugins are installed.**

  We have seen this error before but the exact cause is currently not known. The best suggestion is to make sure you are 1) using Eclipse a supported version (3.1.2 or 3.2*), 2) using Java 5, and 3) have the correct versions of EMF and GEF installed (for Eclipse 3.1* and 3.2* that would be 2.1.0 and 3.1.1 respectively).

• **OpenCCM not working properly.**

  There is a known, but abstract, issue when OpenCCM is installed into a directory tree where spaces exist. For example, it seems common to have spaces in directory names in Windows. So if OpenCCM is installed in "C:\Documents and Settings\someUser\Desktop\Cadena", issues might arise.

Setting your JAVA_HOME Environment Variable

In Linux:

If you're running a BASH shell or a C shell then you will add or edit the JAVA_HOME variable in your .bashrc or .cshrc files respectively, or simply at the command line.

Editing variables in an rc file will make the changes permanent, changing the value of variables from the command line only sets them until you exit the shell or run the rc file again, thereby resetting the value. On the other hand, setting a variable value at the command line gets you immediate results, while setting it in a file requires you to either run the rc file before the changes can take effect or exit and restart the shell, thereby running the rc file again automatically at initialization.

**To set JAVA_HOME in a Bash shell,** set JAVA_HOME to the root directory of your Java JDK and be sure to export it. To set the JAVA_HOME environment variable to `opt/sun-jdk-1.5.0.06`, use `JAVA_HOME=opt/sun-jdk-1.5.0.06`. To export the newly set JAVA_HOME environment variable, use `export JAVA_HOME`. To be sure that JAVA_HOME is set correctly, view it from the command line.
To view JAVA_HOME with a Bash shell, type `echo $JAVA_HOME` at the command-line. If set correctly, you should see the new value of JAVA_HOME output to the screen. In our example, you would see `opt/sun-jdk-1.5.0.06`.

To set JAVA_HOME in a C shell set JAVA_HOME to the root directory of your Java JDK. To set the JAVA_HOME environment variable to point to `opt/sun-jdk-1.5.0.06`, use `setenv JAVA_HOME opt/sun-jdk-1.5.0.06`. To be sure that JAVA_HOME is set correctly, view it from the command line.

To view JAVA_HOME with a C shell, type `echo $JAVA_HOME` at the command line. If set correctly, you should see the new value of JAVA_HOME output to the screen. In our example, you would see `opt/sun-jdk-1.5.0.06`.

Similar approaches should be taken for other shells.

In Windows:

To edit the environment variable graphically, right click on "My Computer" and select: Properties . from the drop down menu. Then select the Advanced tab and click the Environment Variables button. There you will be able to Edit or create a New JAVA_HOME environment variable. Set JAVA_HOME to the root directory of your Java JDK. From the command line, set it thusly: `set JAVA_HOME=C:/java/sun-jdk-1.5.0.06`. To check that it is set correctly, use: `set JAVA_HOME`.
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cadena</td>
<td>An Eclipse-based extensible integrated modeling and development framework for component-based systems.</td>
</tr>
<tr>
<td>TinyOS</td>
<td>An open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks.</td>
</tr>
<tr>
<td>nesC</td>
<td>An extension to the C programming language designed to embody the structuring concepts and execution model of TinyOS.</td>
</tr>
<tr>
<td>Eclipse</td>
<td>An open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. When we refer to Eclipse it is usually as an IDE or platform and not the project or community.</td>
</tr>
<tr>
<td>workspace</td>
<td>An Eclipse term that refers to the central hub for all user data. This is a specific folder/directory. A good quote from the Eclipse website is &quot;you can think of the platform workbench as a tool that allows the user to navigate and manipulate the workspace&quot;.</td>
</tr>
<tr>
<td>project</td>
<td>An Eclipse term that refers to a specific type of resource in the workspace. To be more specific, a workspace contains a collection of projects. Projects contain files and folders.</td>
</tr>
<tr>
<td>Module File</td>
<td>A Cadena term that refers to a file that contains a Cadena Module.</td>
</tr>
<tr>
<td>Scenario File</td>
<td>A Cadena term that refers to a file that contains a Cadena Scenario.</td>
</tr>
<tr>
<td>Scenario</td>
<td>A Cadena term that refers to a collection of instances (component, scenario, and connector) that define a modeled application.</td>
</tr>
<tr>
<td>Module</td>
<td>A Cadena term that refers to the description of the types available in the model which will be used at the Scenario tier. Modules contain definitions of Types that are used to define Scenario instances.</td>
</tr>
<tr>
<td>Style</td>
<td>A Cadena term that refers to the description of the platform that will be modeled at the other tiers of Cadena (module and scenario tiers). In other words, the style helps define a language to use in the Module tier. Styles contain definitions of Kinds (and Meta-Kinds) that are used to define Module Types.</td>
</tr>
<tr>
<td>nesC Interface</td>
<td>A TinyOS/nesC term that refers to a collection of methods (or method signatures) with a name. In nesC, components (modules and configurations) provide and use interfaces.</td>
</tr>
<tr>
<td>nesC Module</td>
<td>A TinyOS/nesC term that refers to a component that holds logic. This uses and provides interfaces, commands, and events. It also holds the logic that maps to the defined interfaces, commands, and events.</td>
</tr>
<tr>
<td>nesC Configuration</td>
<td>A TinyOS/nesC term that refers to a component that does not hold logic. A configuration defines a collection of components (modules and configurations)</td>
</tr>
</tbody>
</table>
and connectors as well as an optional collection of interfaces, commands, and events that it uses and provides. This holds no logic.

<p>| Nature | An Eclipse term that refers to flags set on Eclipse projects. These flags help Eclipse behave in a prescribed way. For example, certain actions, features, and builders are only available in projects with certain natures. For example, the Cadena Specification Path can only be defined in a project with a Cadena nature. |
| Specification Path | A Cadena term that refers to the path Cadena uses to find the model specifications available in a project. This includes three distinct paths for styles, modules, and scenarios. |
| Interface Type | ... |
| Component Type | ... |
| Component Instance | ... |
| Scenario Instance | ... |
| TinyOS Module | A Cadena/nesC term that refers to a Cadena Module that is set to use the nesC style. |
| TinyOS Scenario | A Cadena/nesC term that refers to a Cadena Scenario that is set to use the nesC style. |
| Architectural Definition Language (ADL) | ... |
| Product-Line Development | ... |
| Software Product Lines (SPL) | ... |
| Middleware | ... |
| Type | ... |
| Service | ... |
| Meta Model | ... |
| Component | ... |
| Interface | ... |
| Connector | ... |
| Meta Kind | ... |
| Kind | ... |
| Platform | ... |
| Port Option | ... |
| Role | ... |
| Interface Kind | ... |</p>
<table>
<thead>
<tr>
<th>Glossary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Component Kind</td>
<td>...</td>
</tr>
<tr>
<td>Connector Kind</td>
<td>...</td>
</tr>
<tr>
<td>Instance</td>
<td>...</td>
</tr>
<tr>
<td>Level</td>
<td>...</td>
</tr>
<tr>
<td>Layer</td>
<td>...</td>
</tr>
<tr>
<td>Assembly</td>
<td>...</td>
</tr>
</tbody>
</table>
Bibliography


[nesC:URL] “nesC Web Site”.

[TinyOS:URL] “TinyOS Web Site”.

[Cadena:URL] “\sc Cadena Web Site”.