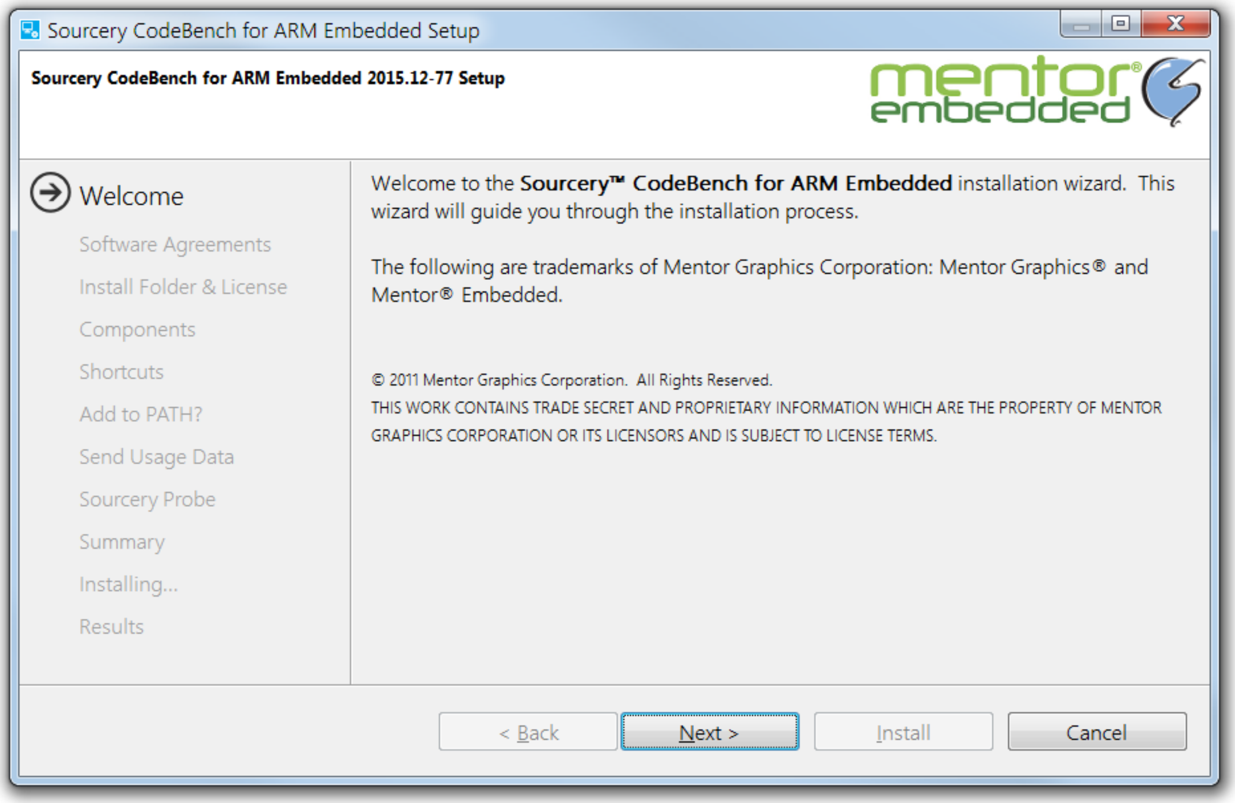
P2 Installer


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.

The P2 Installer is an Eclipse Rich Client Platform (RCP) application that can install products packaged in P2 repositories.

For additional information on P2, see:
- http://www.eclipse.org/equinox/p2/
- https://wiki.eclipse.org/Equinox/p2

The base installer supports:
- GUI, console, and silent operation
- Showing welcome, information, and license pages
- Choosing the install location
- Showing required and optional components for install along with disk usage
- Installing P2 units
- Creating product short-cuts (and Windows add/remove)
- Extending the system PATH environment variable
- Installing Windows drivers
- Launching programs/files at the end of installation

- Uninstalling
- Packaging in a self-extracting binary

The installer can be extended with additional wizard pages and install actions.

Requirements

The P2 Installer requires:

- Eclipse SDK 3.8 or greater
- Oracle Java runtime environment (JRE) SE 7 or greater (Other JRE's are not supported)

Source

The latest source for the P2 installer can be obtained from the GitHub repository, https://github.com/MentorEmbedded/p2-installer. The P2 Installer is composed of one plug-in project, one feature project, and multiple platform specific fragment projects.

com.codesourcery.installer
- P2 Installer plug-in project

feature.com.codesourcery.installer
- P2 Installer feature project

The installer platform specific fragments contain a utility (INSTMON) that is used by the installer to perform certain native operations (like writing to the Windows registry). The *instmon* C/C++ Development Toolkit (CDT) project can be used to build these binaries. It contains project build configurations for different platforms. Make files in the *instmon/src* folder can also be used.

Getting Started

In order to start using the P2 Installer, import the source projects into a new Eclipse PDE workspace. For this example, we will install the Eclipse SDK from the Eclipse download repositories, but you could use any Eclipse P2 based application. Typically, you would want to include the P2 repositories for your product and use them to create a self-contained installer.

Edit the *installer.properties* file in the *feature.com.codesourcery.installer* project. Enter the Eclipse online P2 repository into the *eclipse.p2.repos* property.

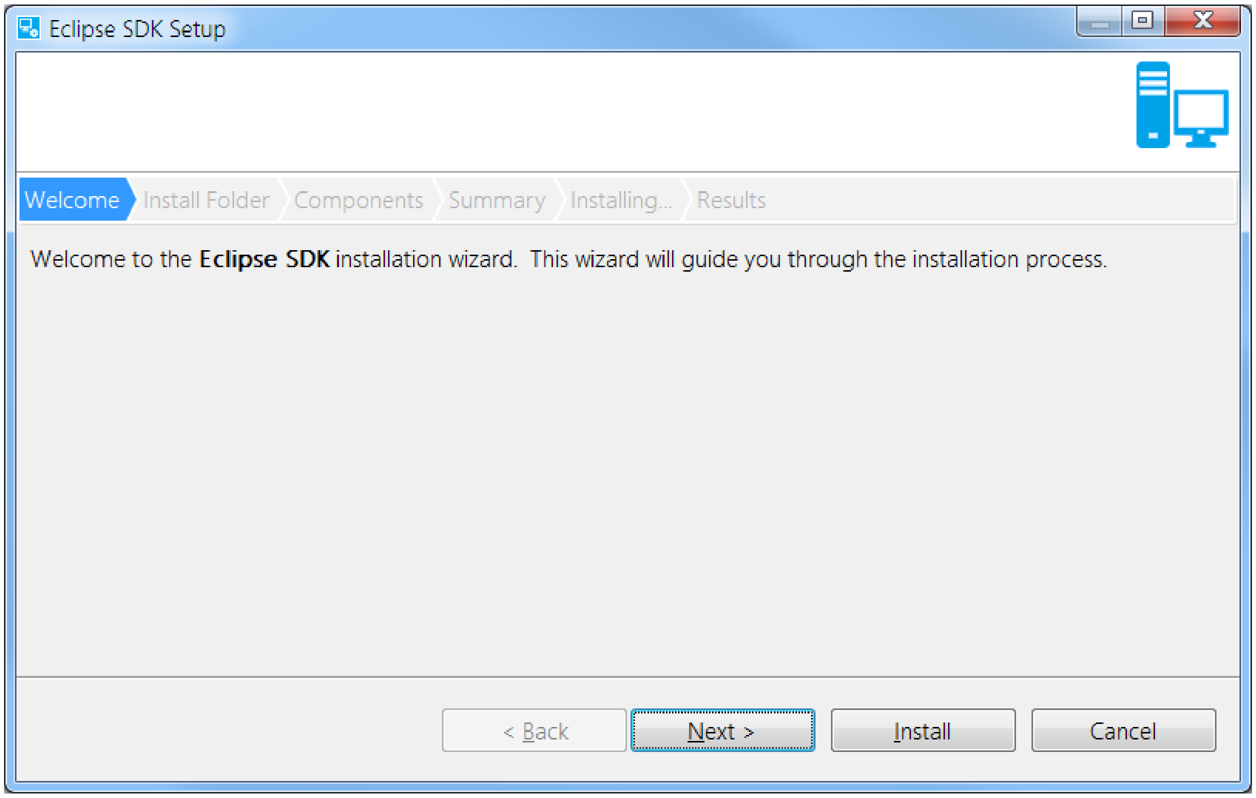
```
eclipse.p2.repos=http://download.eclipse.org/releases/helios
```

Set the *eclipse.p2.product.name* property to "Eclipse SDK".

```
eclipse.p2.product.name=Eclipse SDK
```

```
```

This name will be displayed in the install wizard. You could also set other properties that affect the installer presentation (*eclipse.p2.window.title, eclipse.p2.wizard.image, etc.*).



Welcome to the **Eclipse SDK** installation wizard. This wizard will guide you through the installation process.

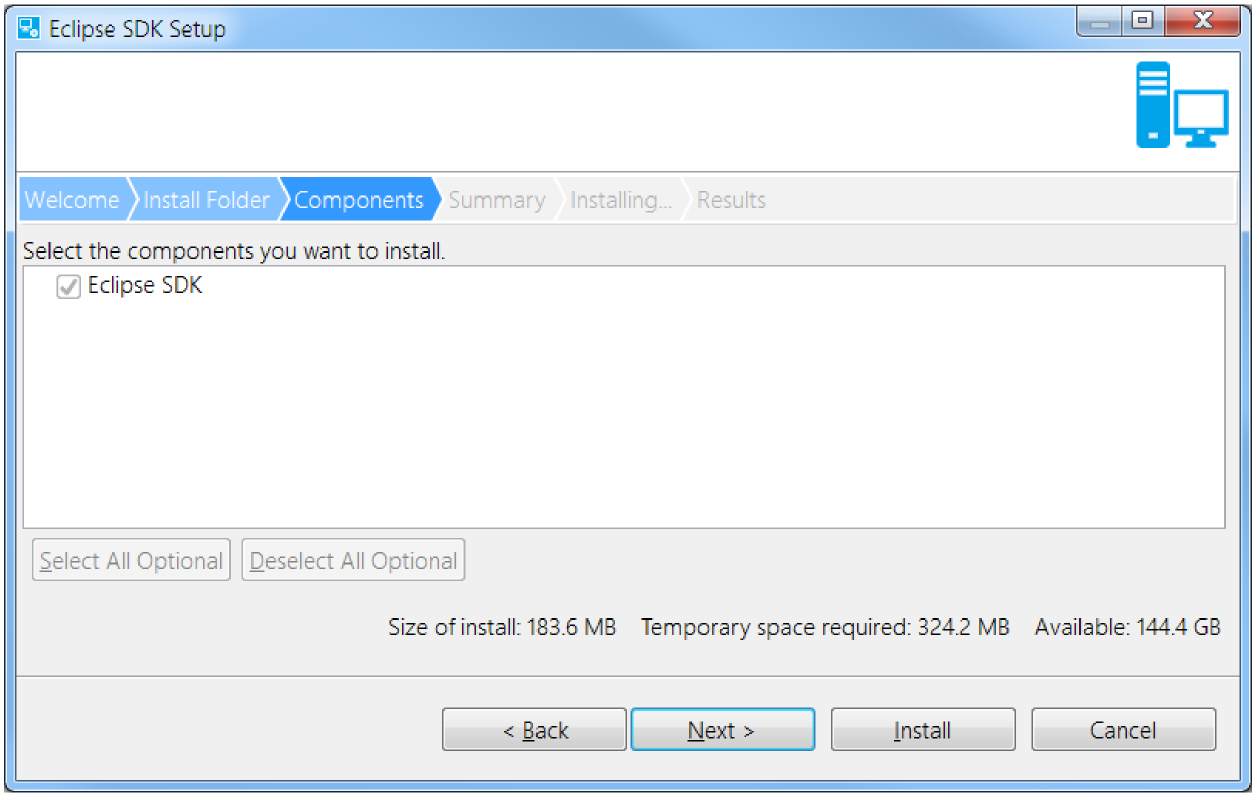
Set the *eclipse.p2.profile.name* property to the P2 profile that will be used.

```
eclipse.p2.profile.name=SDKProfile
```

Set the *eclipse.p2.roots.required* property to the identifier for the Eclipse SDK IU.

```
eclipse.p2.roots.required=org.eclipse.sdk.ide
```

This property lists all the IUs that will be installed. These IUs will be shown on the install wizard's Component page. You could also list optional components using the *eclipse.p2.roots.optional* property.



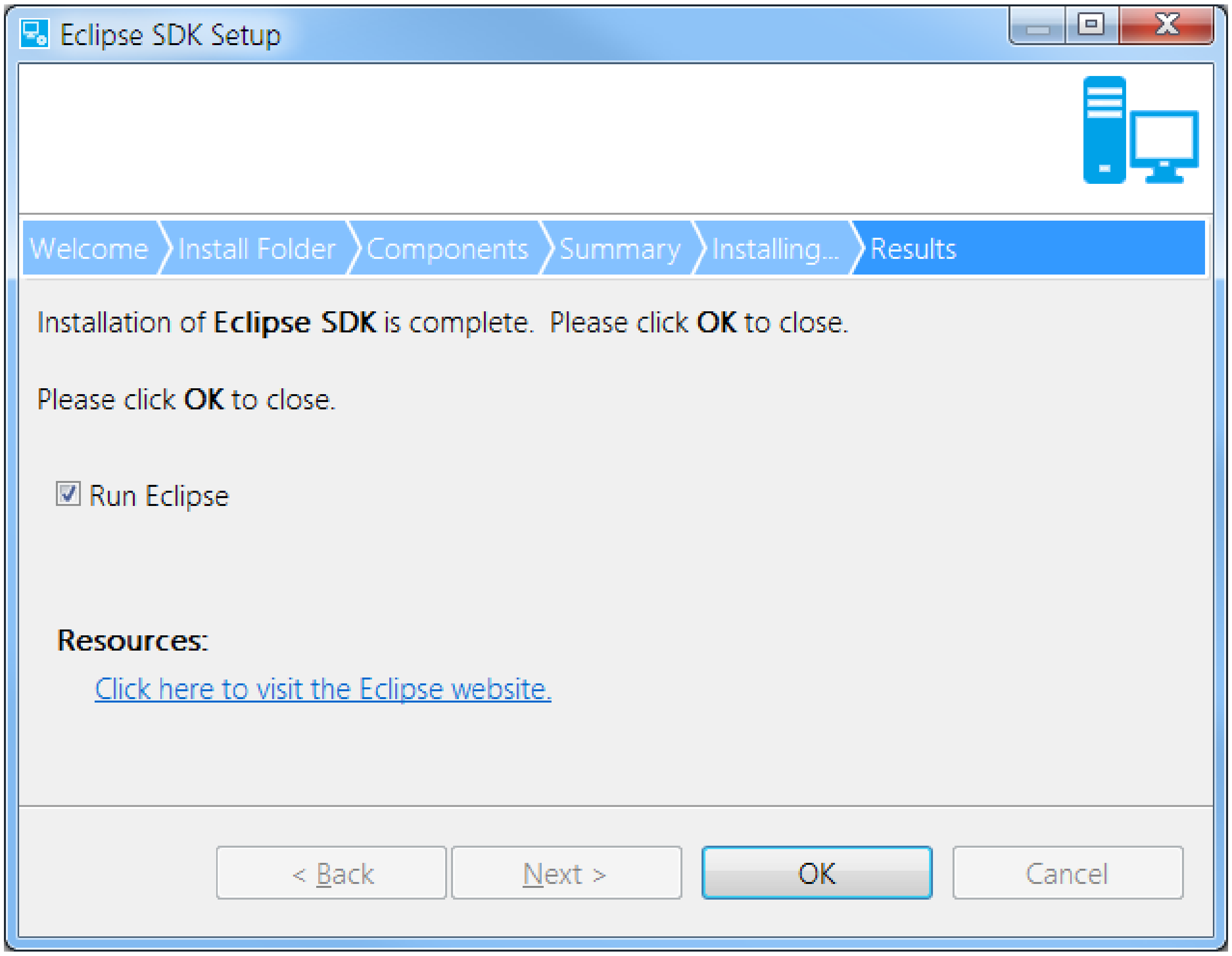
Select the components you want to install.

- [✓] Eclipse SDK

Size of install: 183.6 MB Temporary space required: 324.2 MB Available: 144.4 GB

Optionally, set the *eclipse.p2.wizard.launch* property so that it will run Eclipse after it is installed.

```
eclipse.p2.wizard.launch=\
Run Eclipse;eclipse/eclipse{exe};exe;checked,\
Click here to visit the Eclipse website.;http://www.eclipse.org;html;link
```


Eclipse SDK Setup

Welcome | Install Folder | Components | Summary | Installing... | Results

Installation of **Eclipse SDK** is complete. Please click **OK** to close.

Please click **OK** to close.

☑ Run Eclipse

Resources:

Click here to visit the Eclipse website.

Now run the installer using the **Installer – GUI** launch configuration.

Installer Properties

The P2 Installer is configured using a properties file. This file contains properties that specify what will be installed and the behavior of the installer. Normally, the installer looks for the *installer.properties* in the same directory as the installer launcher binary. If there is a localized directory (for example, "*en_US*"), the installer properties will be loaded from that directory. You can also specify a different properties file to load by using the "-install.desc" command line switch.

The values of installer properties can be expressed in terms of other installer property values using the syntax, *${<property name>}*, where *<property name>* is the name of the property to substitute. Properties can be set to use different values depending on the platform by appending the operating system and/or architecture to the property name. For example, you could use one title for Windows, but then another title for all other platforms.

```
# Title to use on Windows
```

```
eclipse.p2.window.title.win32=Windows Setup
# Title to use on all other platforms
eclipse.p2.window.title=Setup
```

For a complete list of installer properties along with their documentation, refer to the template *installer.properties* file located in the *feature.com.codesourcery.installer* project or refer to **Appendix 1**.

Exporting the P2 Installer

A stand-alone installer can be created by exporting the P2 Installer product. Before exporting, you should set up the *feature.com.codesourcery.installer* for your product.

1. Copy your products' P2 repositories into the *repos* folder.
2. Edit the *installer.properties* file for your product installation.
3. It is advisable to copy the required JRE into the *jre* folder. This will ensure that the installer has a JRE to run and will not be dependent on the system JRE.

Select *File>Export* and choose the *Eclipse product* wizard. Choose the *feature.com.codesourcery.installer/installer.product* (or your own product configuration file) for the configuration to export.

The normal PDE export will produce a launcher executable to run the installer. This will be *setup* on Linux. It will be *setup.exe* and *setupc.exe* on Windows. On Linux, the *setup* binary is used to run the GUI, console, and silent installer. On Windows, *setup.exe* is used to run the GUI installer while *setupc.exe* is used to run the console and silent installers.

On Windows, the UAC will display a warning for any executable run that has *setup* or *install* in the name if it doesn't include the proper embedded manifest. It will also attempt to run the executable using administrator privileges. For these reason, special binaries are exported from the *feature.com.codesourcery.installer/win32.win32.*x86 folder and replace the ones normally generated by the PDE export. These binaries have been embedded with a manifest. The manifest can be found in: *feature.com.codesourcery.installer/win32/installer.manifest*.

You can use the Windows SDK manifest tool (mt.exe) to embed the manifest into a new executable.

```
mt.exe -nologo -manifest "installer.manifest" -outputresource:"setup.exe;#1
```

Refer to the MSDN documentation on *mt.exe* for more information.

Running the P2 Installer

When running from the Eclipse workbench you can use the included launch configurations to run the P2 Installer.

Installer – GUI

- Runs the installer GUI wizard and uses the *feature.com.codesourcery.installer* project for installer contents (installer.properties, P2 repositories, etc.).

Installer – GUI – Browse

- Runs the installer and prompts for the location of the installer contents to use. This is useful for running an already exported installer.

Installer – Console

- Runs the installer in console mode

Installer – Console – Browse

- Runs the installer in console mode and prompts.

Installer – Silent

- Runs the installer in silent mode

Installer – Silent – Browse

- Runs the installer in silent mode and prompts.

Installer – Uninstall

- Runs the uninstaller. This assumes you have run the installer and have product(s) for uninstallation. The launch will prompt for an *install.manifest* file for the product to uninstall. This will be in the product installation *uninstall* directory.

Please note that when running from the Eclipse workbench, no uninstaller will be copied to the product installation *uninstall* directory. The installer binaries needed for copy are only available in an exported installer.

An exported P2 Installer can be run using the *setup* executable. By default the GUI installer is run.

```
Windows
      setup.exe

Linux
      ./setup
```

To run the console installer, specify the "-nosplash –install.console" command line options.

```
Windows
      setupc.exe –nosplash –install.console

Linux
```

```
      ./setup –nosplash –install.console
```

To run the silent installer, specify the "-nosplash –install.silent" command line options. Note, the silent installer will use defaults for all options.

```
Windows
      setupc.exe –nosplash –install.silent

Linux
      ./setup –nosplash –install.silent
```

The command line options supported by the P2 Installer are:

-help
When running from the console, prints usage and exits.

-install.location="<install location>"
This option specifies the directory for install. This will override the default install location specified in the *eclipse.p2.location.root* installer property.
Example:
-install.location="/home/user/software"

-install.desc="<URL of installer.properties>"
This option specifies the path of an installer properties file to use for installation.
Example:
-install.desc="file:/temp/installer.properties"

-install.manifest="<Path to the install manifest file>"
This option specifies the path of a product install manifest file to use for uninstallation.
Example:
-install.manifest="C:\users\user\product\uninstall\install.manifest"

–install.console
This option runs the installer in console mode. If large amounts of text are output, like a license agreement, the console will pause after every 25th line. This limit can be changed by passing **=<limit>** where **<limit>** is the maximum number of lines to display before pausing.
Example:
-nosplash –install.console=120

–install.silent
This option runs the installer in silent mode. In this mode, no install wizard will be displayed and the installation will be performed with only defaults.
Example:
-nosplash –install.silent –install.location="c:\install"

-install.D<property name>=”<property value>”
Sets an installer property value where *<property name>* is the property to set and *<property value>* is the value to set. This will replace any property values loaded from the installer properties file.

-install.status=”<status file path>”
This option creates a status file text file. The *<file path>* will be created after the installer completed and will contain either “OK”, “CANCELED”, OR “FAIL:” followed with an error message.

-install.data=”<directory>”
This option specifies the installer data directory. If *<directory>* does not exist, it will be created. This can also be set using the *eclipse.p2.location.data* installer property. If this option is not specified, the *.p2_installer* directory in the users’ home directory will be used.

-install.mirror=”<ID of the mirror to use>”
This option specifies an installer mirror property to use for installation. If no mirrors are defined then this option does nothing.

Product Information

There are several installer properties that set information for the product to be installed.
The unique identifier for the product is specified in the *eclipse.p2.product.id* property. The unique P2 profile to use is specified in the *eclipse.p2.profile.name* property. The version of the product is specified using the *eclipse.p2.product.version* property. The installer will not allow a version of a product to be installed into a directory already containing the same version of that product. A name to display for the product can be specified in the *eclipse.p2.product.name* property.

Install Location

The *eclipse.p2.location.root* property can be used to set the default install directory for a product. If this property uses “~”, the location will be based on the user’s operating specific home directory.

```
eclipse.p2.location.root=~/${eclipse.p2.product.name}
```

P2 Repositories

The P2 repositories used for installation are listed in the *eclipse.p2.repos* property.

```
eclipse.p2.repos=file:./repos
```

By default, this property is set to the /repos folder. This folder is located in the *feature.com.codesourcery.installer* project and is exported with the installer.

Alternatively, the *eclipse.p2.repos.metadata* property can be used to specify P2 meta-data repositories and the *eclipse.p2.repos.artifacts* property can be used to specify the P2 artifact repositories.

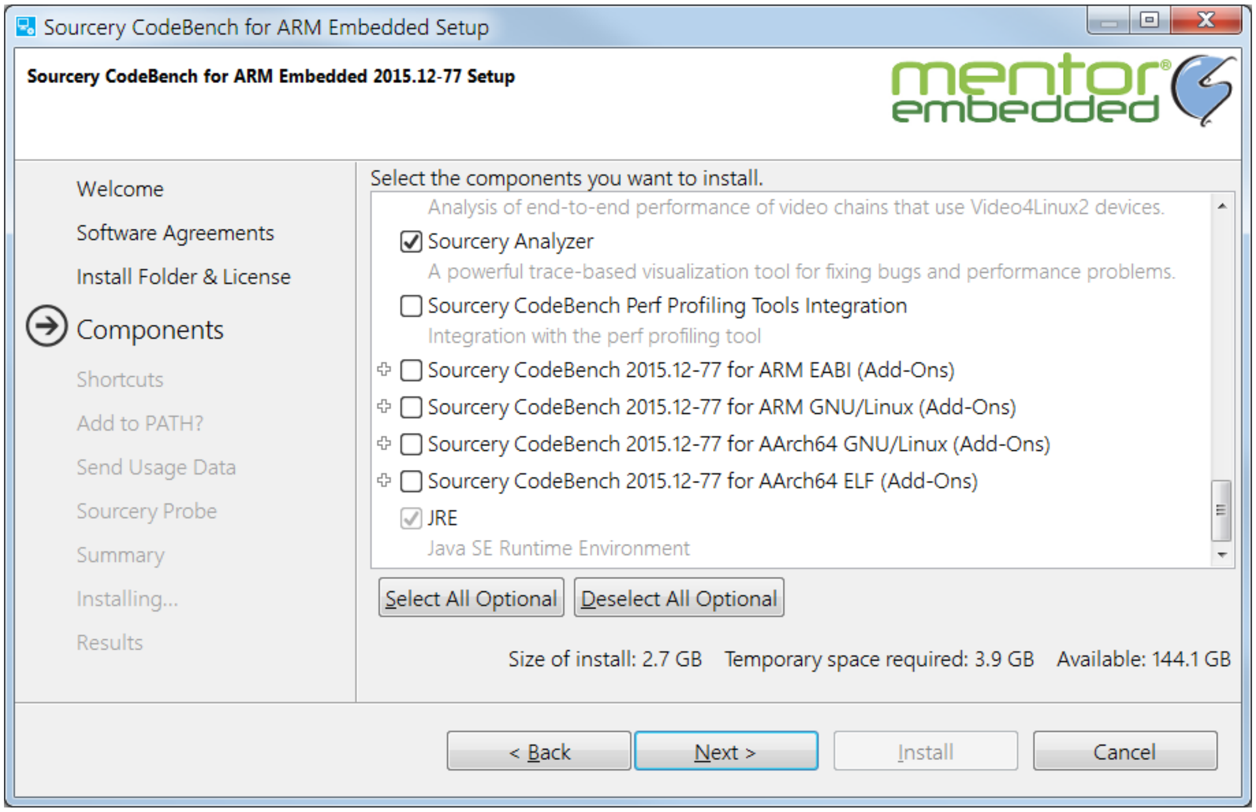
Some examples for referring to a repository location are:

```
file:./local_directory
https://www.mydomain.com/remote_directory
jar:file:./local_repository_archive.zip!/
https://www.mydomain.com/remote_repository_archive.zip!/
```

When constructing an online installer, mirrors can be specified by using the *eclipse.p2.repos.mirror* property. In this case, the repositories should be expressed in terms of the *${eclipse.p2.repos.mirror}* property and the installer will try the various mirrors that are defined until it finds one that is available.

Components

The IUs that will be installed for the product are specified in the *eclipse.p2.roots.required* and *eclipse.p2.roots.optional* properties. Any units specified in the *eclipse.p2.roots.optional* property can be selected by the user on the install wizard **Components** page. Optional roots that should be selected for installation by default can be specified in the *eclipse.p2.roots.optional.default* property. The installation size is computed from the P2 provisioning plan that uses the *artifact.size* property value set for the IUs and their dependent IUs.



Select the components you want to install.

Analysis of end-to-end performance of video chains that use Video4Linux2 devices.

[✓] Sourcery Analyzer
A powerful trace-based visualization tool for fixing bugs and performance problems.

[ ] Sourcery CodeBench Perf Profiling Tools Integration
Integration with the perf profiling tool

[ ] Sourcery CodeBench 2015.12-77 for ARM EABI (Add-Ons)

[ ] Sourcery CodeBench 2015.12-77 for ARM GNU/Linux (Add-Ons)

[ ] Sourcery CodeBench 2015.12-77 for AArch64 GNU/Linux (Add-Ons)

[ ] Sourcery CodeBench 2015.12-77 for AArch64 ELF (Add-Ons)

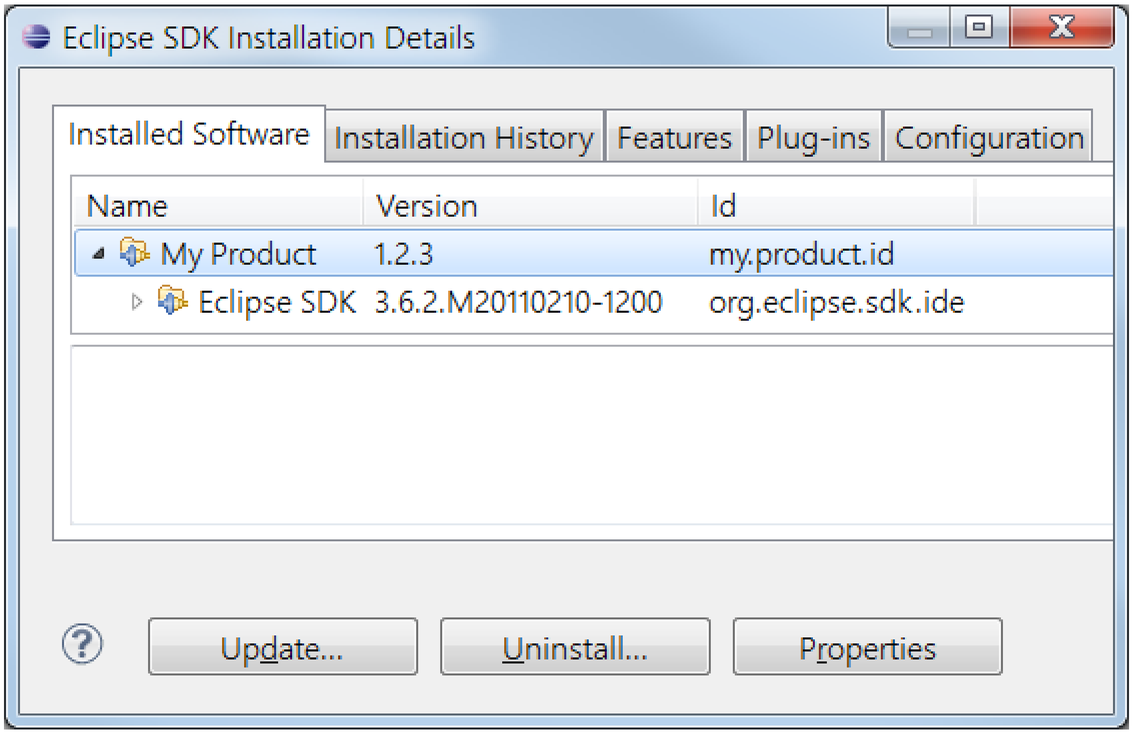
[✓] JRE
Java SE Runtime Environment

Size of install: 2.7 GB Temporary space required: 3.9 GB Available: 144.1 GB

If category IUs (*org.eclipse.equinox.p2.type.category* IU property is set to “true”) are specified then the category along with all of its member units will be displayed in a hierarchal fashion.

By default, the installer will generate a product root IU for provisioning. This IU will have an identifier based on the product identifier (*eclipse.p2.product.id*), name based on the product name (*eclipse.p2.product.name*), and version based on the product version (*eclipse.p2.product.version*). Only the major, minor, and service part of the version will be used. The qualifier part will not be used.

```
eclipse.p2.product.id=my.product.id
eclipse.p2.product.name =My Product
eclipse.p2.product.version=1.2.3.4
```


Eclipse SDK Installation Details

Name	Version	Id
My Product	1.2.3	my.product.id
Eclipse SDK	3.6.2.M20110210-1200	org.eclipse.sdk.ide

The product IU will include requirements on any component IUs selected for installation and the product IU will be provisioned as a root.

If the *eclipse.p2.product.root* property is set to *false*, this behavior will be disabled and each component IU will be provisioned as a root.

Constraints

Relationships between components can be expressed using the *eclipse.p2.roots.constraints* property. Three types of constraints are supported:

ONE_OF – Specifies that at least one out of a certain set of components must be selected for installation.

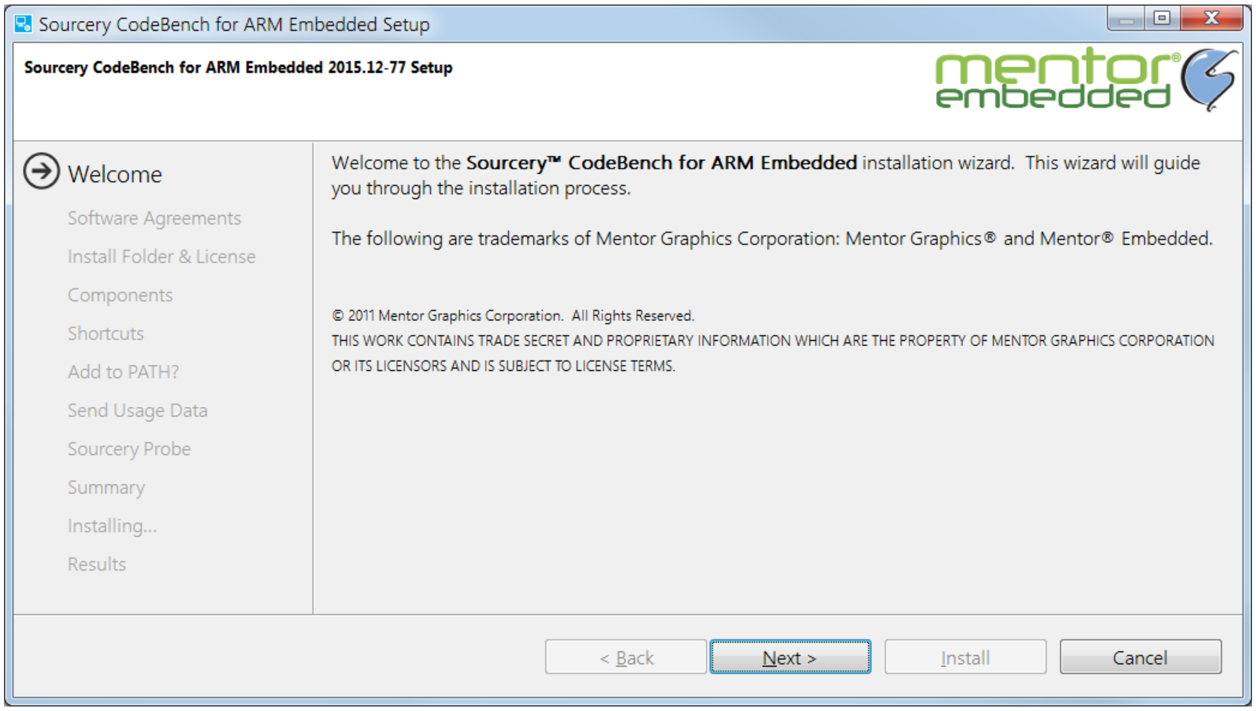
ONLY_ONE – Specifies that only one out of a certain set of components can be selected for installation. The install wizard will automatically deselect other components in the set when more than one is selected.

REQUIRES – Specifies that a particular component requires one or more additional components for installation. When the component is selected, the install wizard will automatically select the required components. If the component is selected and one of its required components is deselected, the components will also be deselected.

Additional component constraints can be specified using the *IInstallVerifier* interface.

Welcome Page

The P2 installer displays a welcome page with information about the product being installed. The information presented on this page can be customized using the *eclipse.p2.wizard.text.welcome* property.



Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

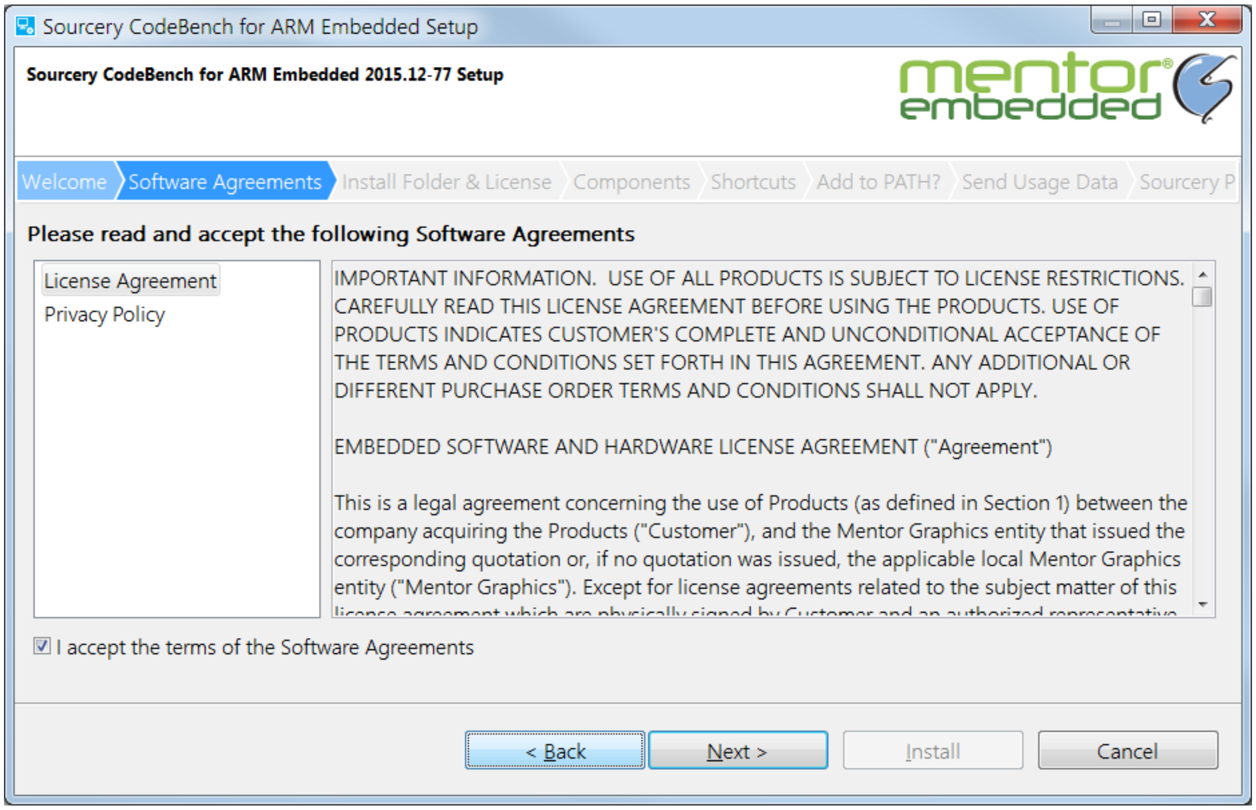
© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.

License Information

The P2 Installer wizard can display a page for the user to accept any required license agreements. License information can be included in files by using the *eclipse.p2.license* property or obtained from the IU meta-data.

```
eclipse.p2.license=license1.txt:License 1,license2.txt:License 2
```

License files must be included in the *build.properties* as root files in order to be available in the exported installer.



Please read and accept the following Software Agreements

IMPORTANT INFORMATION. USE OF ALL PRODUCTS IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF PRODUCTS INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.

EMBEDDED SOFTWARE AND HARDWARE LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Products (as defined in Section 1) between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this

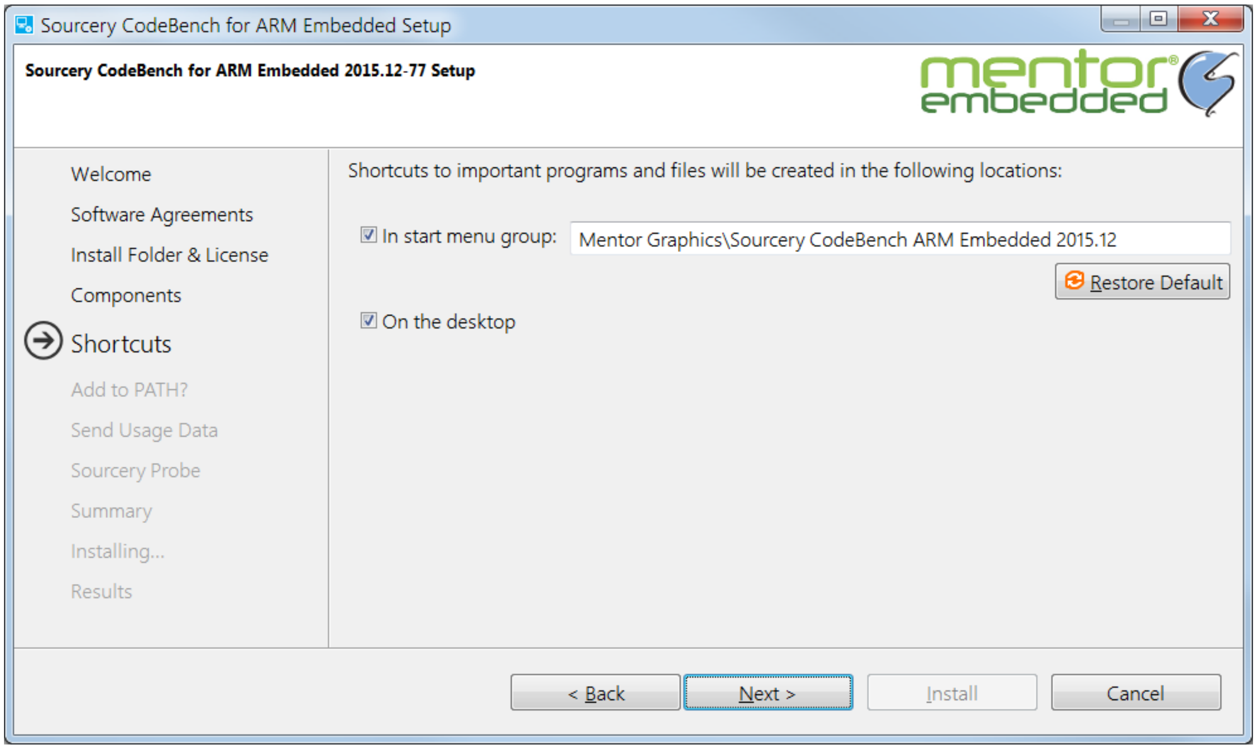
☑ I accept the terms of the Software Agreements

To display license information for IUs, specify them in the *eclipse.p2.license.iu* property.

Product Shortcuts

The P2 Installer can create shortcuts to launch product files using the *eclipse.p2.links* property. Shortcuts can be created in the programs area, desktop, or launcher (Ubuntu Unity only) and can include arguments. On Linux, short-cuts are created using symbolic links. All *program* shortcuts are created relative to the location specified in the *eclipse.p2.location.links* property. On Windows, the *eclipse.p2.location.links* location will be relative to the start menu. On Linux, it will be relative to the installation directory.

```
eclipse.p2.links.location=${eclipse.p2.product.name}
eclipse.p2.links=programs;;${eclipse.p2.product.name};eclipse/eclipse{exe};;;
,
desktop;;${eclipse.p2.product.name};eclipse/eclipse{exe};;;
```


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Shortcuts to important programs and files will be created in the following locations:

[✓] In start menu group: Mentor Graphics\Sourcery CodeBench ARM Embedded 2015.12

[✓] On the desktop

A shortcut can also be created to uninstall the product by pointing to the uninstaller executable if included (i.e. uninstall/uninstall{exe}),

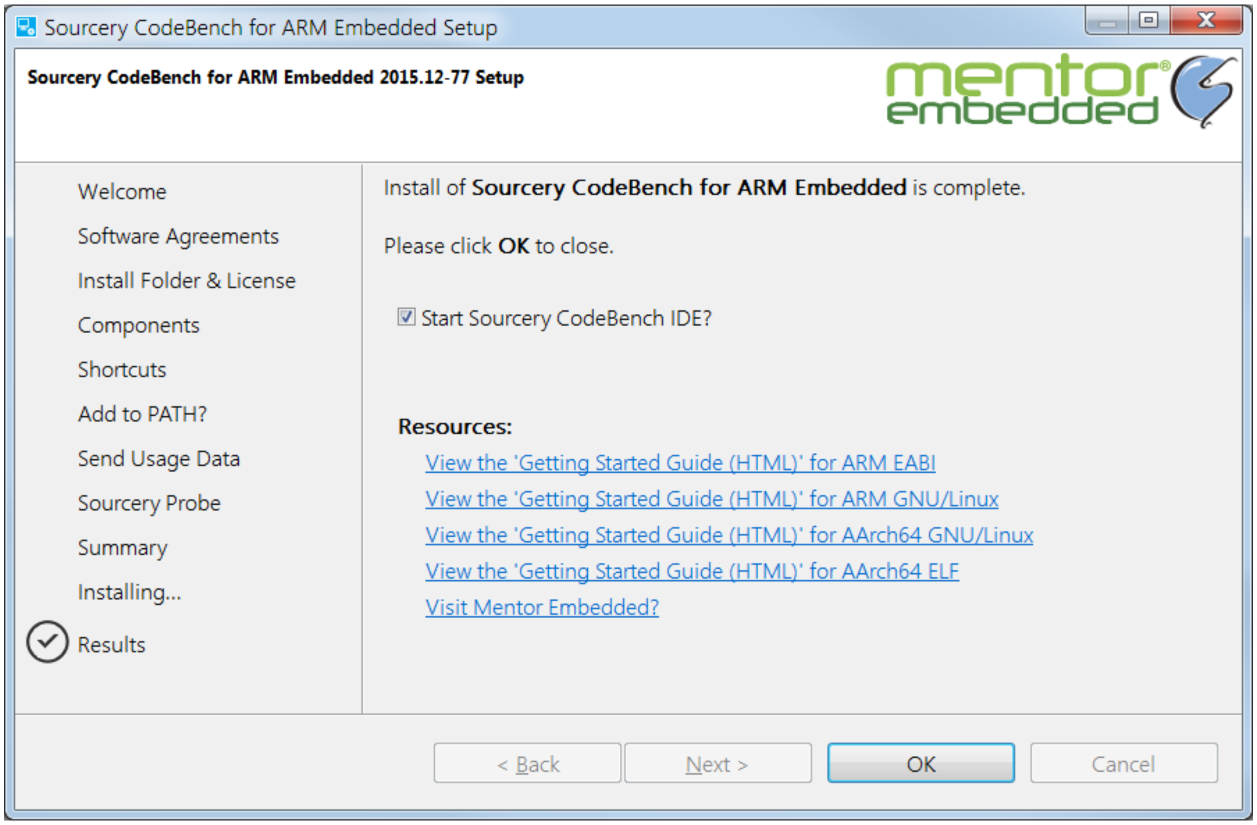
```
eclipse.p2.links=programs;;Uninstall
${eclipse.p2.product.name};uninstall/uninstall{exe};;;
```

Product Launching

The P2 Installer can be set to run executables, open web pages, or open files at the end of a successful installation by using the *eclipse.p2.wizard.launch* property.

```
eclipse.p2.wizard.launch=Run Example;eclipse/eclipse{exe};exe;true,\
View documentation;docs/doc.pdf;file;false,\
Visit www.eclipse.org;http://www.eclipse.org;html;false
```

These items can be set to launch automatically when the installer is closed or only when the user selects a link.


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Install of **Sourcery CodeBench for ARM Embedded** is complete.

Please click **OK** to close.

☑ Start Sourcery CodeBench IDE?

Resources:

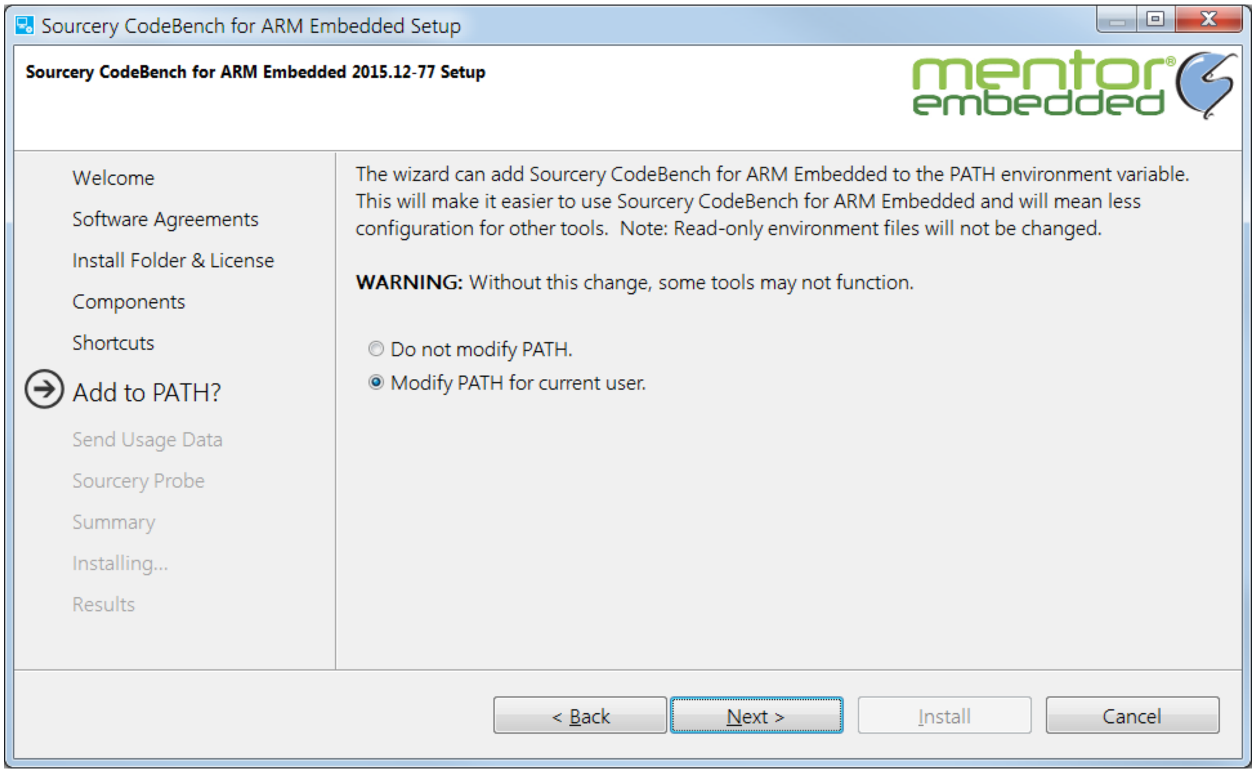
View the 'Getting Started Guide (HTML)' for ARM EABI
View the 'Getting Started Guide (HTML)' for ARM GNU/Linux
View the 'Getting Started Guide (HTML)' for AArch64 GNU/Linux
View the 'Getting Started Guide (HTML)' for AArch64 ELF
Visit Mentor Embedded?

PATH Environment

Some products may require that their binaries be available on the system path. The *eclipse.p2.location.paths* property can be used to add product directories to the system PATH environment variable.

```
eclipse.p2.env.paths=bin
```

Additional environment variables can be managed by providing the appropriate *EnvironmentAction* class from an install module.


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

The wizard can add Sourcery CodeBench for ARM Embedded to the PATH environment variable. This will make it easier to use Sourcery CodeBench for ARM Embedded and will mean less configuration for other tools. Note: Read-only environment files will not be changed.

WARNING: Without this change, some tools may not function.

Do not modify PATH. [✓ Modify PATH for current user.]

Uninstaller

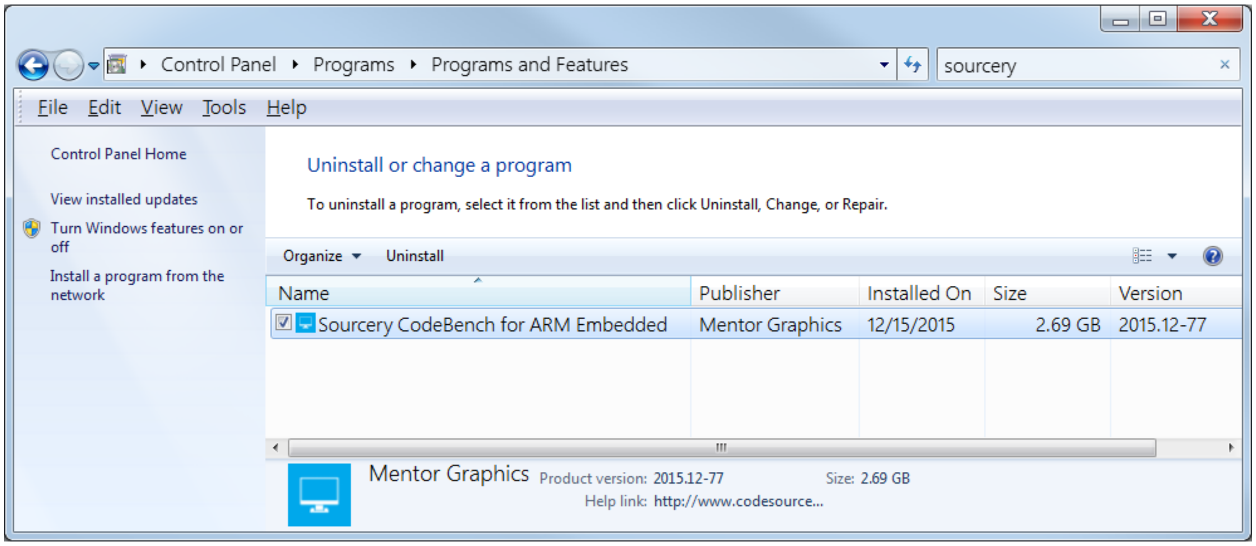
The P2 Installer can also act as an uninstaller for the product. The *eclipse.p2.uninstall.files* property must be filled in with the uninstaller files to copy into the installation uninstall directory,

```
eclipse.p2.uninstall.files=setup{exe}:uninstall{exe},setup.ini:uninstall.ini,
configuration,features,plugins,jre,.eclipseproduct
```

Typically, this will include all directories and files except those that are used for installation (P2 repositories, etc.). Note, these files will only be available in an exported installer so running the uninstall launch configuration from a workbench will not be able to copy the files.

The installer writes a manifest (*install.manifest)* file in the directory that records information about what was installed. The P2 installer uses this during uninstallation.

The *eclipse.p2.uninstall* property can be used to specify flags that control certain behaviors of the uninstaller. For example, including the CREATE_ADD_REMOVE flag will create an entry in the *Add/Remove Programs* entry on Windows. This entry will use several properties to provide product information including *eclipse.p2.product.name, eclipse.p2.product.vendor, eclipse.p2.product.version,* and *eclipse.p2.product.help*.


Control Panel ▸ Programs ▸ Programs and Features

sourcery

File Edit View Tools Help

Control Panel Home

View installed updates

Turn Windows features on or off

Install a program from the network

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

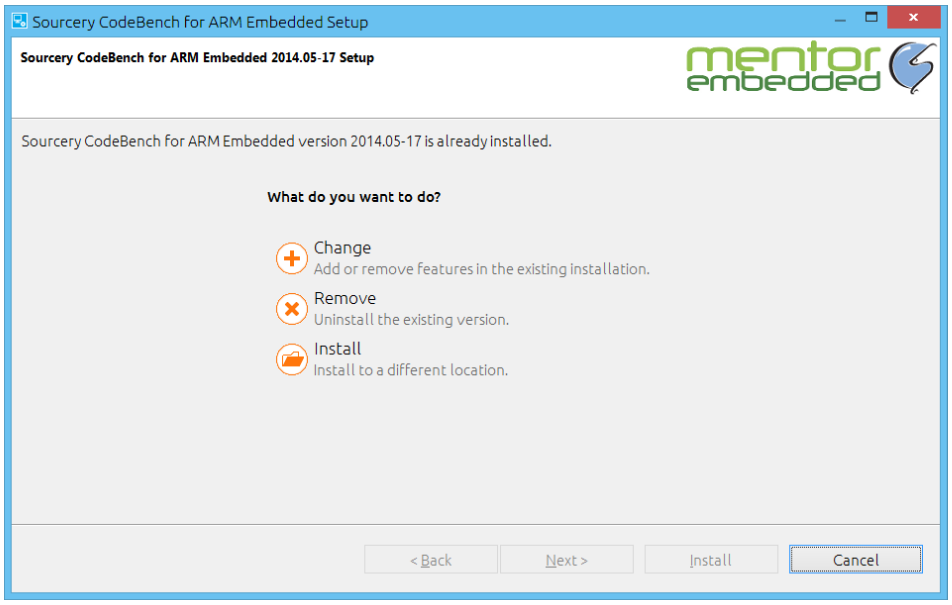
Organize ▾ Uninstall

Name	Publisher	Installed On	Size	Version
Sourcery CodeBench for ARM Embedded	Mentor Graphics	12/15/2015	2.69 GB	2015.12-77

Mentor Graphics Product version: 2015.12-77 Size: 2.69 GB
Help link: http://www.codesource...

Product Update and Upgrade

The installer records all product installations and uninstallations to an install registry (in the installer data area). If an installer for the same version of an installed product is run again, the existing installation will be updated. The user will be prompted to *Change*, *Remove*, or *Install* to a different location.


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2014.05-17 Setup

Sourcery CodeBench for ARM Embedded version 2014.05-17 is already installed.

What do you want to do?

Change
Add or remove features in the existing installation.

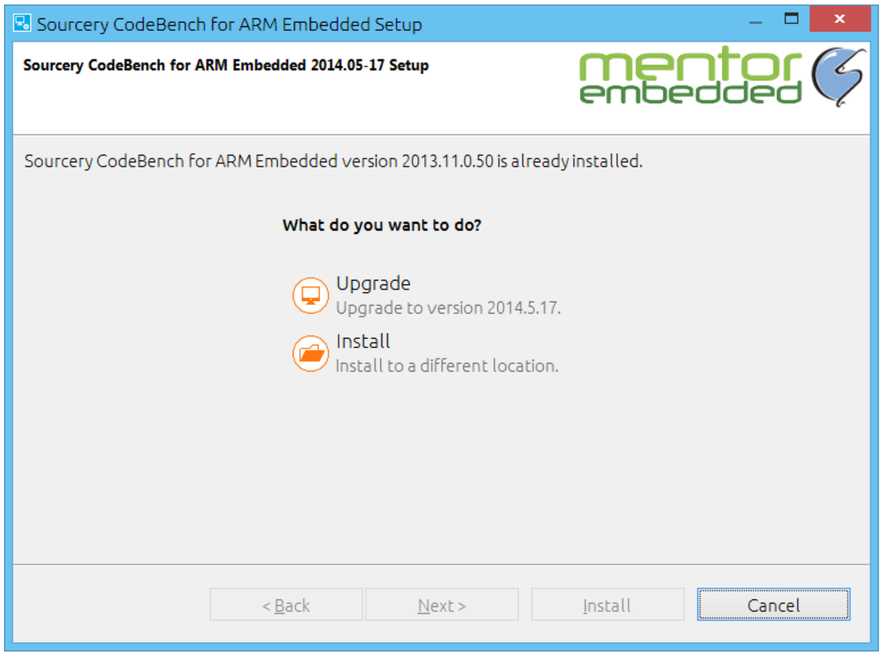
Remove
Uninstall the existing version.

Install
Install to a different location.

< Back Next > Install Cancel

When changing an installation, only the Components and License pages will be displayed in the wizard. Components that are not currently installed can be added to the installation. Components that are currently installed can be removed from the installation. If the installer is set to generate a product root IU (*eclipse.p2.product.root=true*) then the qualifier of the product IU version will be updated with each change.

When installing a newer version of an existing installed product, the user will be prompted to *Upgrade* or *Install* to a different location.


Sourcery CodeBench for ARM Embedded Setup

Sourcery CodeBench for ARM Embedded 2014.05-17 Setup

Sourcery CodeBench for ARM Embedded version 2013.11.0.50 is already installed.

What do you want to do?

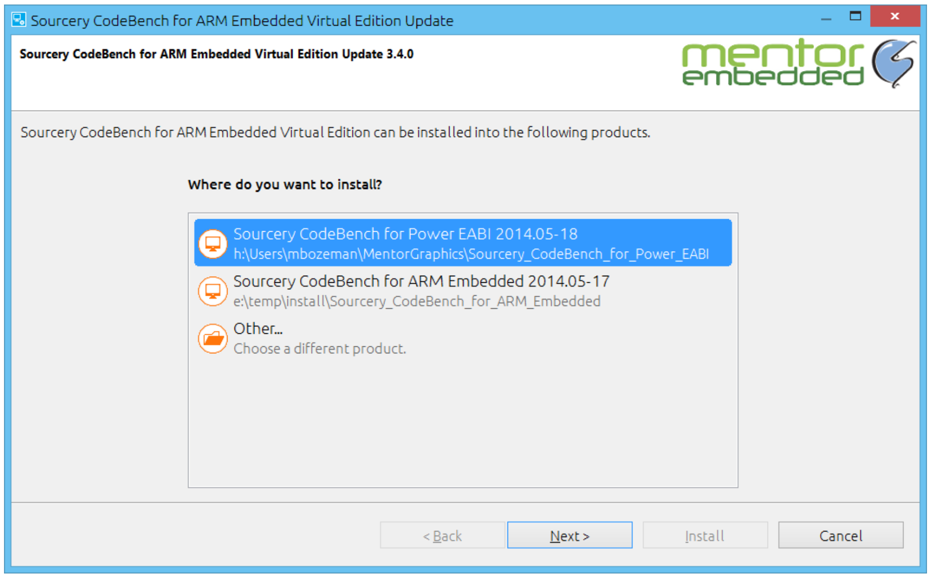
Upgrade
Upgrade to version 2014.5.17.

Install
Install to a different location.

The use of the install registry can be controlled using the *eclipse.p2.useInstallRegistry* property.

Product Patch

A patch installer can be created by setting the *eclipse.p2.patch* property to *true*. An optional, *org.eclipse.p2.requires*, property can specify the products (and optional product version ranges) that can be updated. If the patch is valid for multiple installed products, the installer will prompt for which one to update.


Sourcery CodeBench for ARM Embedded Virtual Edition Update

Sourcery CodeBench for ARM Embedded Virtual Edition Update 3.4.0

Sourcery CodeBench for ARM Embedded Virtual Edition can be installed into the following products.

Where do you want to install?

Sourcery CodeBench for Power EABI 2014.05-18
h:\Users\mbozeman\MentorGraphics\Sourcery_CodeBench_for_Power_EABI

Sourcery CodeBench for ARM Embedded 2014.05-17
e:\temp\install\Sourcery_CodeBench_for_ARM_Embedded

Other...
Choose a different product.

Similar to upgrade, the user will not have to specify the location of an installed product for patching. If the product is not installed (or its location is not found in the install registry) then the user will be required to browse for its location.

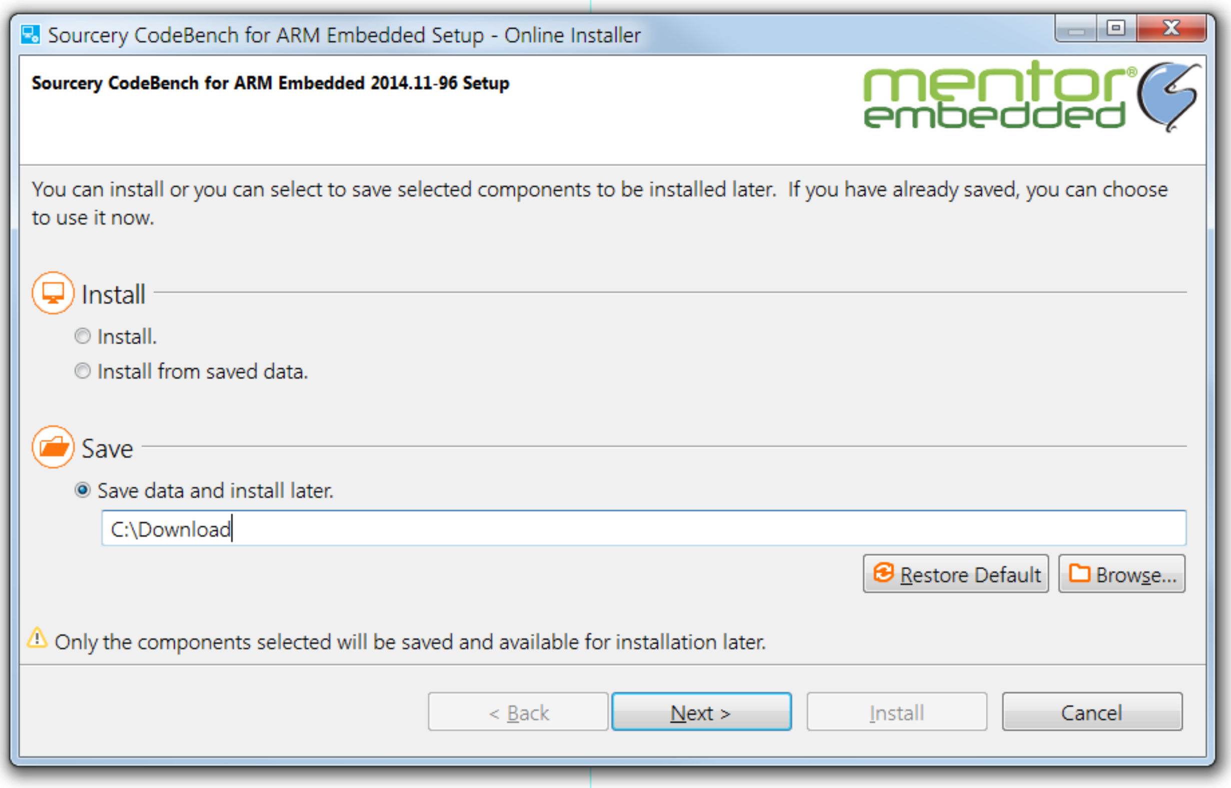
Product Categories

Categories can be defined for products using the *eclipse.p2.product.category* property. A category is used when a product can be installed into other products. If any products with the same category are already installed, their locations will be offered as destinations for a new product installation.

Mirror

The P2 installer can be set to mirror the installation repositories. This can be useful when building an "online" installer to allow the user to download the installation data (mirror the P2 repositories) so multiple local installation can be performed quicker.

If the *MIRROR* or *MIRROR_REMOTE* flags are specified in the *eclipse.p2.install* property, then the install wizard will show a page providing the option to mirror installation data.



Sourcery CodeBench for ARM Embedded Setup - Online Installer

Sourcery CodeBench for ARM Embedded 2014.11-96 Setup

You can install or you can select to save selected components to be installed later. If you have already saved, you can choose to use it now.

Install

( ) Install.
( ) Install from saved data.

Save

(●) Save data and install later.

C:\Download

⚠ Only the components selected will be saved and available for installation later.

If the *MIRROR_REMOTE* flag is specified then the page will only be displayed if there is at least one remote P2 installation repository specified.

The user can select to:

Install – The wizard will proceed, show all install pages, and perform an installation as normal.

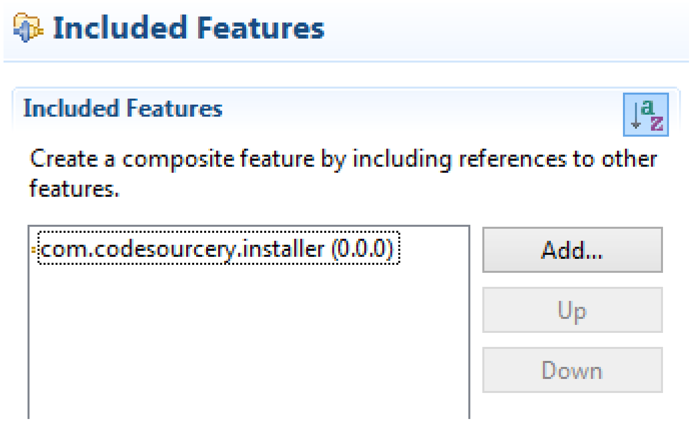
Install from saved data – The wizard will prompt for the location of saved install data. This data is the mirrored P2 installation repositories and will be used instead of the normal installation repositories specified in the installer properties.

Save data and install later – The wizard will prompt for a location to save installation data (mirror the installation repositories).

Creating a Custom Installer

Although the installer can be used by itself, you will probably want to create your own feature and plug-ins in order to add additional install wizard pages and custom install actions.

Create an additional feature and plug-in project for your installer. In the feature project, edit the *feature.xml* and add the installer *com.codesourcery.installer* feature.



Create a composite feature by including references to other features.

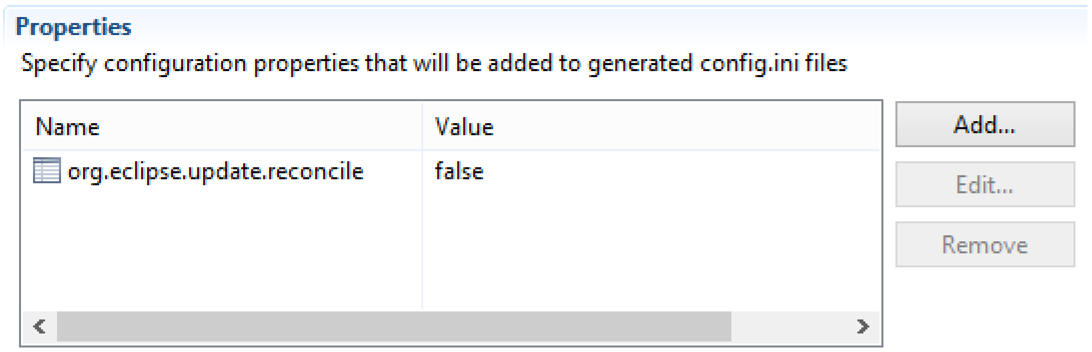
com.codesourcery.installer (0.0.0)

If you will be including any files with the installer like information or license, add these as root files in the *build.properties*.

```
root=file:license.txt
root=file:information.txt
```

Add a project configuration to your feature project (*File>New>Other* and choose *Plug-in Development>Product configuration*). This will be used to export your installer application. Edit the product file. Select the **Dependencies** tab and add your feature along with the *com.codesourcery.installer* feature. On the **Splash** tab, you can select your plug-in for the splash screen or you can use the splash screen included with the *com.codesourcery.installer* plug-in.

In order to avoid "Bundle ... has already been installed" warnings in the Eclipse log file when running, the *"org.eclipse.update.reconcile=false"* property should be added in the **Configuration** tab.


Properties

Specify configuration properties that will be added to generated config.ini files

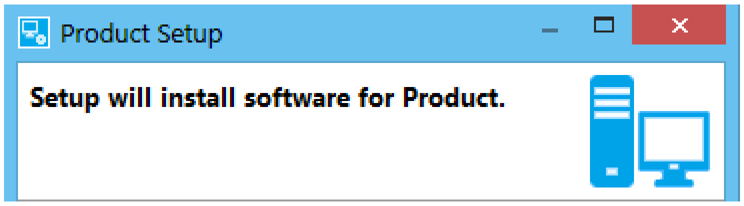
Name	Value
org.eclipse.update.reconcile	false

See: http://www.eclipse.org/forums/index.php/t/158631/ for more information.

Branding

Branding for the P2 installer is done using a combination of installer properties and extension points. The title for the installer window can be set using the *eclipse.p2.window.title* property. A message can be displayed in the title area using the *eclipse.p2.wizard.title* property.

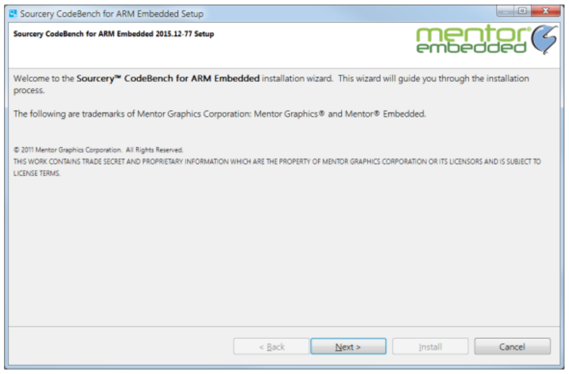
```
eclipse.p2.window.title=${eclipse.p2.product.name} Setup
eclipse.p2.wizard.title=Setup will install software for
${eclipse.p2.product.name}.
```


Product Setup

Setup will install software for Product.

The page navigation bar in the install wizard can be changed using the *org.eclipse.p2.wizard.navigation* property. The bar can be set to not show ("none"), show at the top ("top"), show at the left side ("left"), or show a minimal version on the left side ("left_minimal").

```
org.eclipse.p2.wizard.navigation=top
```

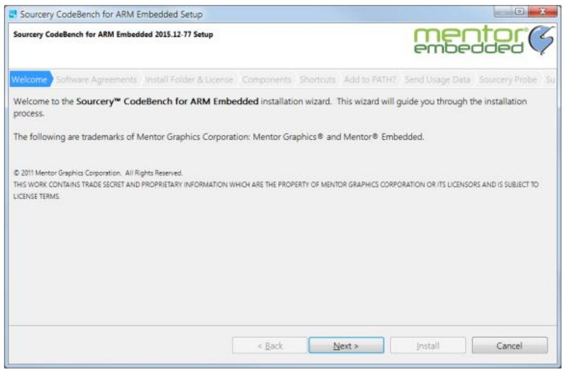


Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.

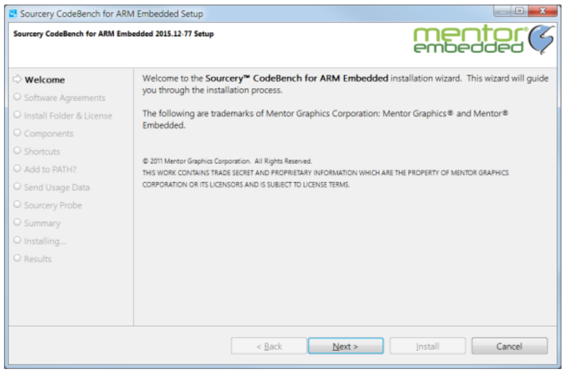


Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.

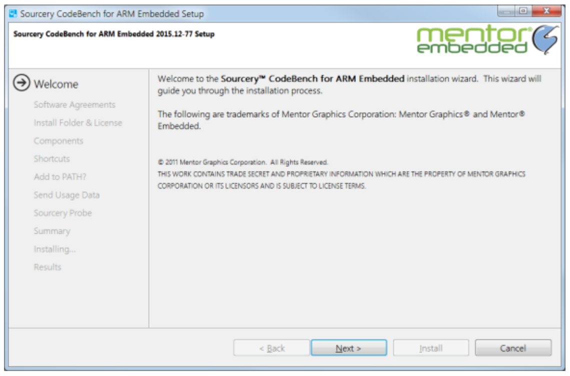


Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.



Sourcery CodeBench for ARM Embedded 2015.12-77 Setup

Welcome to the **Sourcery™ CodeBench for ARM Embedded** installation wizard. This wizard will guide you through the installation process.

The following are trademarks of Mentor Graphics Corporation: Mentor Graphics® and Mentor® Embedded.

© 2011 Mentor Graphics Corporation. All Rights Reserved.
THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION WHICH ARE THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.

To change the splash screen, you can replace *splash.bmp* in the *com.codesourcery.installer* project, but it is recommended that you create your own feature and plug-in projects (see **Creating a Custom Installer**).

The *eclipse.p2.wizard.image* property can be used to specify the image that is displayed in the title area of the wizard. The *com.codesourcery.installer.icon* extension point can also be used to set the image. For example, you could replace it with a *logo.png* image located in your installer plug-in's *icons* folder with the following declaration in the plug-in's *plugin.xml*,

```
<extension point="com.codesourcery.installer.icon">
      <icon image="icons/logo.png" />
</extension>
```

Debugging an Exported P2 Installer

You can debug an exported installer by including its contents (installer.properties, P2 repositories) in the *feature.com.codesourcery.installer* project and running from the workbench. You can also run the workbench using one of the "*- Browse" launch configurations and pointing to the exported installer location when prompted.

You can also attach to running exported installer by doing the following:

1. Add the following additional options after the –*vmargs* option in the *setup.ini* file located in the installer directory:

```
-Xdebug
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000
```

2. Create a new *Remote Java Application* launch configuration. Specify the *com.codesourcery.installer* project.

3. Run the exported installer and then run the launch configuration to attach and debug.

Extending the P2 Installer

The P2 Installer is composed of modules. These modules provide pages that are displayed in the install wizard along with actions that get performed during installation. The installer includes one module, *com.codesourcery.installer.module.default*, that provides the standard wizard pages and install actions.

A new install module is contributed using the *com.codesourcery.installer.modules* extension point. The Java class for a module must implement the *IInstallModule* interface or can extend *AbstractInstallModule*.

```
<extension
  point="com.codesourcery.installer.modules">
    <module
      class="org.eclipse.example.InstallModule"
      id="org.eclipse.example.InstallModule">
    </module>
</extension>
```

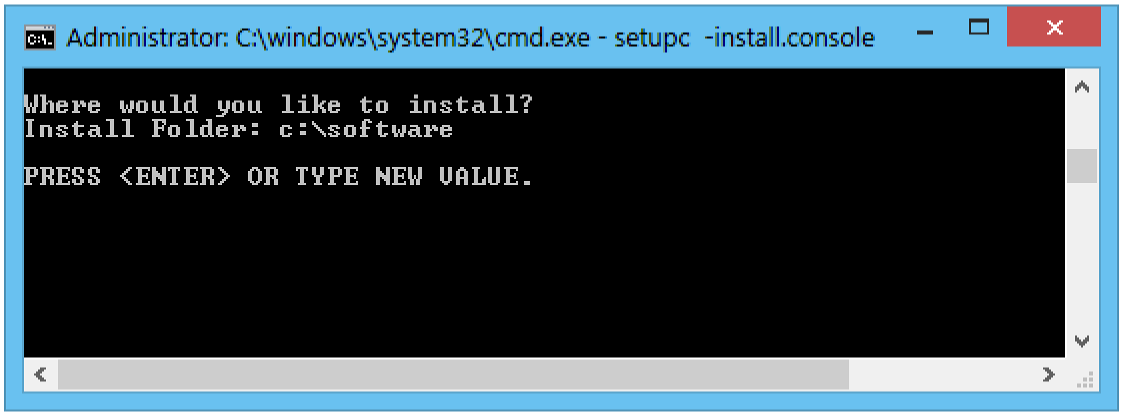
By default, the installer will load all registered modules. You can configure the installer to only use certain modules by listing them in the *eclipse.p2.modules* installer property.

Install Wizard Pages

An install module contributes wizard pages through the *getInstallPages()* method.

The wizard pages must extend the *InstallWizardPage* class. A wizard page is passed an *IInstallData* to save its data and read data set by other pages. Initial install data can be set using the *eclipse.p2.default.** properties.

If a wizard page will be included when the installer is run in console mode, it should implement *IInstallConsoleProvider*. There are three classes that can aid in console presentation. The *ConsoleInputPrompter* can be used to query for simple input. The *ConsoleListPrompter* class can be used to display a list of items to select. The *ConsoleYesNoPrompter* can be used to display a standard yes/no prompt.


Administrator: C:\windows\system32\cmd.exe - setupc -install.console

```
Where would you like to install?
Install Folder: c:\software

PRESS <ENTER> OR TYPE NEW VALUE.
```

If a wizard page will contribute text to the **Summary** page, it should implement the *IInstallSummaryProvider* interface.

Install Actions

An install module can contribute actions to perform during installation through the *getInstallActions()* method. This method is passed an *IInstallData* that can be used to read data set by the wizard pages. An action must implement the *IInstallAction* interface.

An action performs its operation when the *IInstallAction.run()* method is called. The method is passed an *IInstallMode* that specifies the installation mode. The action can perform different operations for installing and uninstalling.

The installer calls the *IInstallAction.save()* method to store action data in the install manifest. When a product is uninstalled, the installer calls the *IInstallAction.load()* method to load the action data.

An action can call *Installer.getInstallPlatform()* to get an *IInstallPlatform* interface that has methods to perform certain platform specific operations (like writing/reading the Window registry, installing Windows drivers, etc.).

Install Verification

An install module can perform additional validation for an installation by registering an install verifier. This can validate the install location, selected components, and log-in credentials.

An install verifier must implement the *IInstallVerifier* interface or extend *AbstractInstallVerifier.* A module should register the install verifier in its *init()* method by calling the *IInstallManager.addInstallVerifier()* method.

Packaging the P2 Installer (SFX)

After the installer has been exported, it can be packaged into small self-extracting binary, or a single executable file, which the user executes to launch the installer. The self-extracting binary is supported on Linux and Windows hosts. Support for SFX on Mac is not included at this time.

User Host Requirements

Linux

The Linux self-extraction script is based on a concept described in the Linux Journal by Jeff Parent (http://www.linuxjournal.com/node/1005818). The p2 installer SFX script itself can be found in the source:

sfx/scripts/src/linux/p2_sfx.h

This script will extract and launch the installer on any modern Linux distribution which includes standard core utilities such as tar with gzip compression (-z argument), df, readlink (-e argument), awk, tail, and others.

The extraction script has been tested using bash and dash shells.

Windows

The Windows self-extracting program will extract and launch the installer on any recent Windows version such as Windows 7 or 8. The SFX for Windows uses unzip.exe from http://www.info-zip.org/. By default, this is automatically bundled with SFX for Windows with no additional software needed to be installed by the user.

The algorithm used by SFX for Windows is a rather simplistic usage of the *incbin* assembly directive to link binary blobs to an executable and then extract them on the user's machine. It uses zip and unzip to create and extract the binary blobs. The source code for Windows SFX can be found here:

*sfx/src/windows/p2_sfx.**

SFX Command Line Arguments

The following command line arguments are supported by sfx for the user on both Windows and Linux hosts:

- -silent
  - Performs a silent installation
- -console
  - Performs a text console based installation

- -location=[path]
  - Specifies the path for the installation
- -noplash
  - Suppresses the splash screen
- -x [path]
  - Extracts the installer to the path specified but does not start the installer.

All other arguments are passed directly to the installer.

By default, the extractor will extract the installer to the system temporary directory. A different directory can be used by setting its path in the *P2_INSTALLER_TEMP_PATH* environment variable.

The SFX will put any logs in the user's home location in a directory named *.p2_installer*. On Linux SFX typically won't add a log unless the JRE crashes on start up. On Windows however, SFX is a Microsoft GUI application and not a console application. Therefore it is not capable of printing any output (including help) to the command line. This means on Windows, all status and errors are printed directly to the log file which by default is named *p2_installer_extraction.log*. If the SFX fails on ar Windows machine, check the log file.

Creating the SFX

The creation of the SFX binary for both Linux and Windows must be performed on a Linux host and requires a bash shell. In addition, the creation of the Windows SFX requires the Minimalist GNU tools for Windows to be installed.

The scripts that can be used to build the SFX executable binary can be found here:

```
sfx/scripts/build
```

Building SFX

The script that can be used to build the Linux SFX executable is

```
sfx/scripts/build/build_linux_p2_sfx.sh.
```

The script that can be used to build the Windows SFX executable is

```
sfx/scripts/build/build_windows_p2_sfx.sh.
```

To brand SFX for your product, you can modify these scripts directly or use them as a template. The arguments for the SFX build scripts are as follows:

> --installer-dir <directory where the exported installer to be packaged by SFX can be found. This argument is required>

--scratch-dir <directory where temporary objects and files can be written during SFX generation. This argument is optional. The script will use the current directory if not specified>

--data-dir <name of directory for the installer data. This argument is optional. The script will use ".p2_installer" if not specified>

--output-file <path and file name of generated SFX binary. This argument is required>

--silent <Turn off messages from this script. This argument is optional>

The Linux extraction script (p2_sfx.h) contains the following tags:

#@PREPROCESS@

and

#@POSTPROCESS@.

These tags can be replaced with any script code you deem necessary by writing your own SED scripts.

The windows extraction program behavior and splash screen may be changed by modifying the code in the functions found in build_windows_p2_sfx.sh.

Windows SFX build

The script that can be used to build the Windows SFX executable is *sfx/scripts/build/build_windows_p2_sfx.sh*. You can modify this file directly or use it as a template. It takes similar input arguments as the build script for Linux SFX, but it also adds the following:

--compiler <path to mingw compiler. Required>

--resource-compiler <path to mingw resource compiler. Required>

SFX JRE Requirement

The SFX for Linux and Windows requires a JRE to be present in the following directory before the installer will successfully launch on the user's computer:

```
$installer_dir/jre/bin
```

The easiest way to accomplish this is to add a JRE to the installer exported by eclipse prior to building SFX.

Repositories

SFX requires that all P2 repositories for the product be in zip files. Directory repositories are not supported.

Windows Firewall

On Windows, P2 can sometimes trigger a firewall warning during installation. This is due to the *BackupStore* class in the *org.eclipse.equinox.p2.touchpoint.natives* bundle. The *genUnique()* method uses *InetAddress* methods to generate a unique string. This can be avoided by patching the *org.eclipse.equinox.p2.touchpoint.natives* plug-in and changing the *genUnique()* method to use Java 1.6 UUID. The *genUnique()* method implementation can be replaced with the following:

```
private String genUnique() {
        long timePart = (System.currentTimeMillis() << 5) | (msCounter++ & 31);
        return Long.toHexString(timePart) + "_" + UUID.randomUUID().toString();
}
```

Signing

You may want to digitally sign all of your installer executables. This can prevent Antivirus software reporting the binaries as suspicious. You can check various Antivirus reports against files using a tool like: https://www.virustotal.com/

Appendix 1: Installer Properties

```
################################################################################
#  Copyright (c) 2015 Mentor Graphics and others.
#  All rights reserved. This program and the accompanying materials
#  are made available under the terms of the Eclipse Public License v1.0
#  which accompanies this distribution, and is available at
#  http://www.eclipse.org/legal/epl-v10.html
#
#  Contributors:
#     Mentor Graphics - initial API and implementation
################################################################################

################################################################################
# This file specifies properties used by the installer.  The file is first
# searched for in a locale directory in the installer directory (i.e. /en_US).
# If a locale directory is not found, it will be looked for in the installer
# directory.
#
# Property values can be expressed in terms of other property values or system
# environment variable values using the following syntax:
#   ${<property name>}
# where
#   <property name> is the name of the other property.
# or
#   ${<environment variable name}
# where
#   <environment variable name> is the name of the system environment variable.
#
# The property name can optionally be prefixed with a tag to indicate special
# processing should be performed on the property value,
#   ${<tag>:<property name> or <environment variable name>}
# Supported tags are:
#   filename:  - Converts the property value to a file safe format.
#   nospace:   - Replaces any spaces in the property value with underscores.
#   nonum:     - Removes any numbers and periods from the property value.
#   uppercase: - Converts the property value to uppercase.
#   lowercase: - Converts the property value to lowercase.
#
# A platform specific property can be specified by appending the operating
# system and/or architecture to the property name,
#   <property>.<os>.<arch>
# where
#   <property> is the property name.
#   <os> is the operating system, linux, win32, macosx, etc.
#   <arch> is the architecture, ie. x86, x86_64, etc.
#
# The order that a property will be read is:
#   <property>.<os>.<arch> if present
#   <property>.<os> if present
#   <property> if present
################################################################################

################################################################################
# P2 Repositories (Required)
#   Specify the URL for P2 repositories.  Separate multiple repositories with
#   a comma.
#   By default, the installer expects repositories to be located in the /repos
#   folder of the feature.com.codesourcery.installer feature.
#
#   To specify an archived repository use the following syntax,
#     "jar:file:<full or relative path to archive file>!/"
#   For Example:
#     jar:file:./repos/repo-file.zip!/
################################################################################
eclipse.p2.repos=file:./repos

################################################################################
# P2 Meta-data Repositories (Required)
#   Specify the URL for the P2 meta-data repositories.  Separate multiple
#   repositories with a comma.
################################################################################
eclipse.p2.repos.metadata=${eclipse.p2.repos}

################################################################################
# P2 Artifact Repositories (Required)
#   Specify the URL for the P2 artifact repositories.  Separate multiple
#   repositories with a comma.
################################################################################
eclipse.p2.repos.artifacts=${eclipse.p2.repos}

################################################################################
# P2 Repository Mirrors (Optional)
#   This property can be used to specify mirror sites for the P2 repositories.
#   A default location is specified in the eclipse.p2.repos.mirror property and
#   additional locations can be specified in properties according to the
#   syntax:
#     eclipse.p2.repos.mirror.<id>=
#   where <id> is some identifier.
#
#   If the eclipse.p2.repos.metadata and eclipse.p2.repos.artifacts property
#   values are expressed in terms of ${eclipse.p2.repos.mirror}, the P2 installer
#   will first attempt to load the repositories using the default
#   ${eclipse.p2.repos.mirror} value and if that fails, proceed to try the
#   various mirrors defined, substituting ${eclipse.p2.repos.mirror} with
#   their values.
#
#   Example:
#     eclipse.p2.repos.mirror=http://mydomain.com/repos
#     eclipse.p2.repos.mirror.alt1=http://mydomain2.com/repos
#     eclipse.p2.repos.mirror.local=file:./local/repos
#
#     eclipse.p2.repos=${eclipse.p2.repos.mirror}/myrepo
```

```
#
#   The -install.mirror=<id> command line option can be used to specify that a
#   a particular mirror or "default" should be used.  If specified, the
#   installer will only contact that site.
#
#   Example:
#     -install.mirror=default
#     -install.mirror=alt1
#     -install.mirror=local
################################################################################
#eclipse.p2.repos.mirror=

################################################################################
# Update Site (Optional)
#   Optional update sites for the product.
#   Each update site specified will be added and available in the 'Install New
#   Software' dialog.
#   Separate multiple update sites with a comma.  An optional name for the update
#   site can be specified using a semicolon.
#     <update site 1 URL>;<update site 1 name>,<update site 2 URL>;<update site 2
#     name>,...
#   Example:
#     http://www.mysite1.com;Update Site 1,http://www.mysite2.com;Update Site 2
################################################################################
#eclipse.p2.update=

################################################################################
# Product Install Location (Required)
#   Specifies the default root install location.  Use '~' to specify a location
#   based on the operating system specific home directory.  To specify a
#   a location based off the Windows system drive, use ${SystemDrive}.
#
#   Note: Do not use this property in the values for other properties as it can
#   be changed by the user in the install wizard.
#
#   Example:
#     eclipse.p2.location.root=~/${eclipse.p2.product.name}
#     eclipse.p2.location.root.win32.x86=~/${eclipse.p2.product.name}/win32_x86
#     eclipse.p2.location.root.win32=~/${eclipse.p2.product.name}/win32
#     eclipse.p2.location.root.linux=~/${eclipse.p2.product.name}/linux
################################################################################
eclipse.p2.location.root=~/${eclipse.p2.product.name}

################################################################################
# P2 Install Location (Required)
#   Specifies the location for the P2 (i.e. Eclipse) installation directory
#   relative to the product location (eclipse.p2.location.root).
#   The installer will append a "P2" directory to the location specified.
################################################################################
eclipse.p2.location.install=eclipse

################################################################################
# Short-cuts Location (Optional)
#   Specifies the short-cut/links location.  This location is relative to the
#   Start Menu on Windows and installation directory on Linux.
#   This property is required if the "eclipse.p2.links" property is set.
################################################################################
#eclipse.p2.location.links=.

################################################################################
# PATH (Optional)
#   Specifies paths to append to the system PATH environment variable.  The
#   paths are relative to the product installation directory
#   (eclipse.p2.location.root).  Separate multiple paths with a comma.
#   If this property is specified, the install wizard will display a page to
#   choose if the path should be modified.
#
#   Example:
#     eclipse.p2.env.paths=bin
################################################################################
#eclipse.p2.location.paths=
################################################################################
# Installer (Optional)
#   This property specifies certain behaviors for the installer.  You can
#   specify a combination of the following modes (separated with '|').
#
#     NO_UPGRADE - By default the installer prompts to upgrade older versions
#     of installed products.  If this mode is specified, the installer will not
#     prompt to upgrade previous installations and will not allow installing
#     to an existing product location.
#
#     NO_UPDATE - By default the installer prompts to update existing installed
#     products of the same version.  If this mode is specified, the installer
#     will not prompt to update.
#
#     EMPTY_DIRECTORY - By default the installer does not require the
#     installation directory to be empty.  If this mode is specified, the
#     installer will not allow installation into a directory that is not empty
#     unless that directory contains existing installed products.
#
#     MIRROR - Show options in the install wizard to 1)install, 2) save the
#     install to a directory, or 3) install from a saved directory. The saved
#     directory will contain a P2 repository with only the components selected
#     for installation.
#
#     MIRROR_REMOTE - Show mirror options in the install wizard (see MIRROR),
#     only if the installation contains remote repositories.
#
################################################################################
#eclipse.p2.install=

################################################################################
# Size Format (Optional)
#  The installer can display size information on the Components page.  This
#  property specifies the format to display the information.  The property value
#  can contain the following variables:
#    {0} - Replaced with installation size as computed from IU artifact sizes.
#    {1} - Replaced with the temporary size required for installation. This
#          includes the IU artifact and download sizes.
#    {2} - Replaced with the available space on the install location.
#
#  If this property is not specified, default formatting will be used.  If this
#  property is set to an empty value, no size information will be displayed.
#
#  Example:
#    eclipse.p2.wizard.sizeFormat=\
#    Size of install: {0}    Total space required: {1}    Available: {2}
################################################################################
#eclipse.p2.install.size.format=Size of install: {0}    Temporary space required: {1}
Available: {2}

################################################################################
# Minimum Upgrade Version (Optional)
#   This property specifies a minimum version of the product that this
#   installer can upgrade.  If this property is included, the installer will
#   not attempt to upgrade an existing installed product unless its version is
#   greater than or equal to the version specified.
#   This property is only applicable if eclipse.p2.install does not specify
#   NO_UPGRADE.
################################################################################
#eclipse.p2.install.upgrade.minVersion=

################################################################################
# Uninstaller (Optional)
#   This property specifies the behavior of the uninstaller.  You can specify
#   a combination of the following modes.
#
#     CREATE_ADD_REMOVE - If this mode is specified, a control panel add/remove
#     entry will be created when installing on the Windows platform.  This will
#     only be done if the eclipse.p2.uninstall.files is specified.
#
```

```
#     REMOVE_DIRS - If this mode is specified, all directories created during
#     installation will be removed on uninstallation when there are no more
#     products installed.
#
#     SHOW_UNINSTALL - If this mode is specified, the product will be shown in
#     the uninstaller.  If it is not specified, the product is internal and will
#     be uninstalled when all other products are uninstalled.
#
#   Specify one or modes separated with a '|'.
#
#   If this property is not specified, no uninstaller will be included in the
#   product installation directory. Regardless of the modes specified, the
#   uninstall directory will always be removed when the last product is
#   uninstalled.
#
#   Example: Full RCP installation:
#     eclipse.p2.uninstall=CREATE_ADD_REMOVE|REMOVE_DIRS
#   Example: Installing plug-ins into existing Eclipse
#     eclipse.p2.uninstall=CREATE_ADD_REMOVE
################################################################################
#eclipse.p2.uninstall=CREATE_ADD_REMOVE|REMOVE_DIRS

################################################################################
# Uninstaller Files (Optional)
#   This property specifies the installer directories and files to copy into
#   the uninstall directory for the uninstaller. If a file is an executable
#   binary, append {exe} that  will be replaced with ".exe" on Windows and
#   nothing on Linux.  A file or directory can be copied to a different name by
#   specifying a ':' and the new name.
#   If this property is specified and the eclipse.p2.uninstall property is not
#   specified, a default mode of CREATE_ADD_REMOVE|REMOVE_DIRS will be used.
################################################################################
#eclipse.p2.uninstall.files=setup{exe}:uninstall{exe},setup.ini:uninstall.ini,configuration,featu
res,jre,plugins,.eclipseproduct

################################################################################
# P2 Flavor (Required)
#   As part of a product build, PDE/Build automatically generates default
#   configuration meta-data to set start levels and config.ini property. This
#   meta-data is commonly referred to as Configuration Units (CUs). In
#   particular, start levels are set using CU fragments on the IU for the bundle
#   being started. The flavor is used as a qualifier when generating the CU's
#   name based on the IU.  For example, with "p2.flavor = tooling",
#   'toolingwin32.win32.x86org.eclipse.core.runtime' will be the name of the CU
#   that configures the org.eclipse.core.runtime bundle on windows. It may be a
#   good idea to use a flavor based on your product id to avoid conflicts with
#   other meta-data, particularly if your product has particular needs with
#   respect to start levels.
################################################################################
eclipse.p2.flavor=tooling

################################################################################
# Product Identifier (Required)
#   Specifies a unique identifier for the product.  This identifier must be
#   different than any IU identifier planned for installation.
################################################################################
eclipse.p2.product.id=${nospace:eclipse.p2.product.name}

################################################################################
# Product Name (Required)
#  Specifies a name for the product.
################################################################################
eclipse.p2.product.name=Software

################################################################################
# Uninstall Product Name (Optional)
#  Specifies a name for the product to be used for uninstallion. This property
#  is only applicable for Windows.
################################################################################
#eclipse.p2.product.uninstall.name=Software
```

```
################################################################################
# Category (Optional)
#  Specifies a category for the product.  A category is used when a product can
#  be installed into other products.  If any products with the same category are
#  already installed, their locations will be offered as destinations for this
#  product installation when the installer is run.
################################################################################
#eclipse.p2.product.category=

################################################################################
# Product Vendor Name (Optional)
#  Specifies the vendor name for the product.  This will appear in the
#  add/remove entry on Windows host.
################################################################################
#eclipse.p2.product.vendor=Eclipse.org

################################################################################
# Product Version (Required)
#   Specifies the version of the product
################################################################################
eclipse.p2.product.version=1.0.0

################################################################################
# Product Help URL (Optional)
#   Specifies the URL for product help.  This will appear in the add/remove
#   entry on Window host.
################################################################################
#eclipse.p2.product.help=http://eclipse.org/

################################################################################
# Create Product Root IU (Optional)
#   If this property is set to 'true', the installer will create a root group
#   IU for the product.  The IU identifier will be set to the value of
#   eclipse.p2.product.id.  The IU name will be set to the value of
#   eclipse.p2.product.name.  The IU will contain requirements on every
#   component IU selected for installation (from eclipse.p2.roots.required and
#   eclipse.p2.roots.optional).
#
#   If this property is set to 'false', the installer will provision all
#   component IUs directly as roots.
#
#   The default for this property is 'true' if not specified.
################################################################################
#eclipse.p2.product.root=false

################################################################################
# P2 Profile (Optional)
#   Specifies the P2 Profile to use for installation.
#   If no profile is specified, the first available profile will be used if
#   available.
################################################################################
eclipse.p2.profile.name=SoftwareProfile

################################################################################
# Remove Profile IUs (Optional)
#   This property specifies if all installable units included in the profile
#   will be removed when the product is uninstalled or upgraded.  This will
#   include any units that are installed into the profile after the initial
#   installation.
#   If this property is set to "false" or not specified, on the root IUs
#   recorded during installation and any root IUs dependent on those IUs
#   will be removed.
################################################################################
#eclipse.p2.profile.remove=true

################################################################################
# Short-cuts (Optional)
#   Specifies short-cuts/links to create.
#   If this property is specified, the installer will display a page to choose
#   short-cuts.
#
```

```
#   Separate multiple short-cuts with a comma.
#   The format for links is:
#     <folder>;<link path>;<link name>;<link target>;<icon path>;<arguments>
#   where,
#     <folder> is
#       programs - Programs folder.  This folder is relative to the
#         eclipse.p2.location.links folder.
#       desktop - The desktop folder
#       launcher - The Unity launcher panel (Ubuntu Linux only)
#     <link path> - The folder/directory to create the short-cut in.  This path
#       is relative to the <folder> location and can be omitted.
#     <link name> - The name for the short-cut.
#     <link target> - The path to the target file for the short-cut relative
#       to the root install location.  If the link target is an executable
#       binary, append {exe} that will be replaced with ".exe" on Windows and
#       nothing on other platforms. The special variable {cmd} can be specified
#       as a link target and will be replaced with the path to the Command Shell
#       (Windows only).
#     <icon path> - The path to the icon file that will be used for the
#       short-cut.  This path is relative to the <folder> location and can be
#       omitted.  If this parameter is not specified, the <link target> will
#       be used for the icon if needed.
#       This parameter is only used on Windows and Unity launcher short-cuts.
#       The icon file name can be appended with {icon} that will be replaced
#       with ".ico" on Windows and ".png" on other platforms.
#     <arguments> - A space-separated list of command-line arguments that will
#       be passed to the executable when the link is activated. Arguments may
#       contain the variable {installFolder} to refer to the absolute install
#       location as chosen by the user in the install folder page (Windows only).
#
#   Example:
#     programs;Mentor Graphics/CodeBench#CodeBench;bin/eclipse{exe};;;
#     desktop;;CodeBench;bin/eclipse{exe};bin/icon.ico;;
#
#   Created Short-cut directories will be removed on uninstallation if they are
#   empty.
###############################################################################
#eclipse.p2.links=

###############################################################################
# Short-cut Defaults (Optional)
#   Specifies the short-cuts that will be selected by default.  Separate
#   short-cut folders to be selected by  a comma.
#   If this property is not specified, all short-cuts will be created by
#   default.
#   This property is only applicable if the "eclipse.p2.links" property is set.
#
#   Example:
#     eclipse.p2.links=desktop,programs
###############################################################################
#eclipse.p2.links.default=desktop,programs,launcher

###############################################################################
# Required Root Installable Units (Required)
#   Specifies the identifiers of the root installable units (IU) that will be
#   installed.  IUs that are specified will be listed on the Components page of
#   the install wizard.  If the IU is a category IU, it's members will also be
#   listed.  The user will not be allowed to select what required components
#   get installed.
#   Any IUs listed must be present in the P2 repositories specified in the
#   "eclipse.p2.repos" property.
#   Separate multiple IUs with a comma.
#
#   Example:
#     eclipse.p2.roots.required=org.eclipse.example,org.eclipse.extras
###############################################################################
#eclipse.p2.roots.required=

###############################################################################
# Optional Root Installable Units (Optional)
#   Specifies the identifiers of the root installable units (IU) that can be
#   optionally installed.  IUs that are specified will be listed on the
#   Components page of the install wizard.  If the IU is a category IU, it's
#   members will also be listed.  The user will be allowed to select what
#   optional components get installed.
#   Any IUs listed must be present in the P2 repositories specified in the
#   "eclipse.p2.repos" property.
#   Separate multiple IUs with a comma.
################################################################################
#eclipse.p2.roots.optional=

################################################################################
# Default Optional Root Installable Units (Optional)
#   Specifies the optional root installable units that should be selected by
#   default.  Any IUs listed will initially be checked on the install wizard
#   Components page.
#   Separate multiple IUs with a comma.
#   Note: Any IUs found in an existing installation that is being upgraded,
#   will be selected for install automatically regardless if they are not listed
#   in this property.
################################################################################
#eclipse.p2.roots.optional.default=

################################################################################
# Root Installable Unit Constraints (Optional)
#   Specifies constraints for the combination of IUs that can be installed.
#   A constraint has the following format:
#     <constraint>(<IU>,<IU>,...)
#   where <constraint> is the type of constraint and <IU> is a root identifier.
#
#   The type of constraints supported are:
#     ONE_OF   - At least one of the roots specified must be selected for a new
#                installation.
#     ONLY_ONE - Only one of the specified roots can be selected.
#     REQUIRES - The first root specified requires the other roots specified.
#
#   Separate multiple constraints with a semicolon.
#
#   Example:
#     eclipse.p2.roots.constraints=\
#     ONE_OF(org.eclipse.iu1,org.eclipse.iu2,org.eclipse.iu3);\
#     REQUIRES(org.eclipse.iu4,org.eclipse.iu5,org.eclipse.iu6);\
#     ONLY_ONE(org.eclipse.iu7,org.eclipse.iu8)
#
#     - Either org.eclipse.iu1, org.eclipse.iu2, or org.eclipse.iu3 must be
#       selected.
#     - org.eclipse.iu4 requires org.eclipse.iu5 and org.eclipse.iu6
#     - Only org.eclipse.iu7 or org.eclipse.iu8 can be selected.
################################################################################
#eclipse.p2.roots.constraints=

################################################################################
# License Agreement (Optional)
#   Specifies files containing license agreements for the product.  If this
#   property is specified, the install wizard will display a page for the user
#   to accept the license agreements.
#   Multiple files should be separated with a comma.  An optional name to
#   display for the license agreement can be specified separated with a ':'.
#   The format is:
#     <path to license file 1>:<optional name>,
#     <path to license file 2>:<optional name>,
#     ...
#     <path to license file 3>:<optional name>
#
#   Examples:
#     eclipse.p2.license=./license.txt
#     eclipse.p2.license=./license1.txt, ./license2.txt
#     eclipse.p2.license=./license1.txt:License Agreement A,\
#     ./license2.txt:License Agreement B
#
#   The file paths are relative to the installer.properties file.
#   If no name is provided for the license agreements, the licenses will be
#   named "License Agreement 1", "License Agreement 2", etc.
#
#   To include a license file, place it in the
#   feature.com.codesourcery.installer project and modify the build.properties
#   to include it as an exported root file.
#   For example, if the license file is license.txt, the following would need
#   to be added in the build.properties,
#     root=file:installer.properties
#   The property would be set to:
#     eclipse.p2.license=./license.txt
###############################################################################
#eclipse.p2.license=

###############################################################################
# IU License Agreement (Optional)
#  Specifies if license information from selected component IUs should be
#  displayed on the wizard License page.  If this property is specified and
#  any listed IUs are selected for installation, the license agreement for
#  those IUs will be displayed.
#  An optional name to display for the license agreement can be specified
#  separated with a ':'.  If no name is entered then the name of the IU will be
#  used.  Separate multiple IUs with a comma.
#  The format is:
#    <IU #1>:<optional name>,<IU #2>:<optional name>,...,<IU #n>:<optional name>
#
#  Examples:
#    eclipse.p2.license.iu=com.example.iu1
#    eclipse.p2.license.iu=com.example.iu1:IU1 License Agreement
#    eclipse.p2.license.iu=com.example.iu1:IU1 License Agreement,com.example.iu2
###############################################################################
#eclipse.p2.license.iu=

###############################################################################
# P2 Progress Filter (Optional)
#   Specifies a set of regular expressions to filter the progress messages
#   reported from P2 during installation.  Multiple properties can be specified
#   and the first match will be applied.
#
#   The format of the property names is:
#     eclipse.p2.progress.find.0
#     eclipse.p2.progress.find.1
#     ...
#
#   For each eclipse.p2.progress.find property, there should be a corresponding
#   eclipse.p2.progress.replace property that specifies the expression used to
#   replace the progress message.  The eclipse.p2.progress.find expressions can
#   contain groups that are referenced by the eclipse.p2.progress.replace
#   expressions.
#
#   Example:
#    eclipse.p2.progress.find.0=\
#    (.*) (([_A-Za-z0-9-]+)\\.([_A-Za-z0-9-]+)\\.([._A-Za-z0-9-]+))(.*)
###############################################################################
eclipse.p2.progress.find.0=(.*) (([_A-Za-z0-9-]+)\\.([_A-Za-z0-9-]+)\\.([._A-Za-z0-9-]+).jar)(.*)
eclipse.p2.progress.find.1=(.*) (([_A-Za-z0-9-]+)\\.([_A-Za-z0-9-]+)\\.([._A-Za-z0-9-]+))(.*)

###############################################################################
# P2 Progress Filter Replacement (Optional)
#   Specifies a set of regular expressions to replace the progress messages
#   from P2.
#
#   The format of the property names is:
#     eclipse.p2.progress.replace.0
#     eclipse.p2.progress.replace.1
#     ...
#
#   For each eclipse.p2.progress.find property, there should be a corresponding
#   eclipse.p2.progress.replace property that specifies the expression used to
#   replace the progress message.  The eclipse.p2.progress.find expressions can
#   contain groups that are referenced by the eclipse.p2.progress.replace
#   expressions.
#   The special macro, $IU_NAME(), can be used to replace an IU identifier
#   with the IU name.
#
#   Example:
#     eclipse.p2.progress.replace.0=Installing $2
#     eclipse.p2.progress.replace.1=Configuring $IU_NAME($1)
################################################################################
eclipse.p2.progress.replace.0=Loading $2
eclipse.p2.progress.replace.1=Installing $2

################################################################################
# P2 Properties (Optional)
#   Specifies additional properties used by P2.
#   The property, "org.eclipse.update.install.features", set to "true" will
#   cause the P2 update manager features to be installed.
################################################################################
org.eclipse.update.install.features=true

################################################################################
# Component Versions (Optional)
#   By default, the installer will not display the versions of components on the
#   Components page.  If this property is set to 'true' then component versions
#   will be displayed on components page.
################################################################################
#eclipse.p2.wizard.components.showVersion=true

################################################################################
# Optional Components First (Optional)
#   By default, the installer will display required components first followed
#   by the optional components on the wizard Components page.  If this property
#   is set to 'true' then the optional components will be displayed first.
################################################################################
#eclipse.p2.wizard.components.showOptionalFirst=true

################################################################################
# Component Expansion (Optional)
#   By default, the installer will display any category roots in a collapsed
#   state on the wizard Components page.  This property can be used to specify
#   category roots that should be expanded by default.  The specified category
#   root and all of its children will be expanded.
#   Separate multiple roots with a comma.  This property has no effect if the
#   specified root is not a category IU.
################################################################################
#eclipse.p2.wizard.components.expand=

################################################################################
# Installer Modules (Optional)
#   The installer can be extended with modules that implement the
#   com.codesourcery.installer.IInstallModule interface or extend the
#   com.codesourcery.installer.AbstractInstallModule class.  Modules can
#   contribute additional install wizard pages and install actions.
#   Installer modules are registered using the
#   "com.codesourcery.installer.modules" extension point.  The installer
#   includes one module, "com.codesourcery.installer.module.default", that
#   contributes the common wizard pages and install actions.  By default, all
#   registered modules are loaded.  This property can be used to only include
#   certain modules.  The identifiers for modules that should be loaded should
#   be separated with a comma.
################################################################################
#eclipse.p2.modules=com.codesourcery.installer.module.default

################################################################################
# Exclude Actions (Optional)
#   Specifies install actions that should be excluded.  By default, all actions
#   provided by install modules are run during install.  If this property is
#   specified, the listed actions will be excluded.
#   Specify the identifiers for any install actions to exclude separated by a
#   comma.
################################################################################
#eclipse.p2.actions.exclude=
################################################################################
# Patch Installer (Optional)
#   Optional property to specify this installer is a patch for an existing
#   product.
################################################################################
#eclipse.p2.patch=true

################################################################################
# Product Requirements (Optional)
#   Optional property to specify this installer requires other products be
#   installed.  This property specifies the identifiers of the products (and
#   optional version ranges) to be updated the installer.  Separate multiple
#   products with a semicolon. If the install can only be applied to specific
#   versions of a product, specify the version range separated by a colon.
#
#   <1st Product ID>:<Version Range>;<2nd Product ID>:<Version Range>;...
#
#   Specify the minimum and maximum versions of the product separated by a comma
#   and between either '['/']' to include the version in range or '('/')' to
#   exclude the version from the range.
#   If the patch can be installed to any version of the product, don't include
#   a version range for it.
#
#   The Setup page in the install wizard will display all installed products
#   that are valid for this installation.  The product will be installed into
#   the existing product chosen from the list.  If no installed product is found,
#   the user can browse for the location of the product to update.
#
#   Example:
#     To allow a patch to install into a version of a product PRODUCT_1 that
#     is greater than or equal to 2.0, but less than 3.0 and also any versions
#     of PRODUCT_2:
#
#     eclipse.p2.requires=PRODUCT_1:[2.0,3.0);PRODUCT_2
#
################################################################################
#eclipse.p2.requires=

################################################################################
# Missing Requirement Expressions (Optional)
#   This property can be used to specify a series of find/replace regular
#   expressions that are used to format P2 error messages regarding missing
#   requirements.  If these properties are not specified, or no expressions
#   match then the original P2 information will be displayed.
#
#   The normal format for a P2 missing requirement messages is similar to:
#
#     Cannot complete the install because one or more required items could not
#     be found.
#     Software being installed: My Feature Name 1.0.0
#     (com.company.feature.group 1.0.0)
#     Missing requirement: Other Feature Name 1.0.0 (com.company.feature 1.0.0)
#     requires 'bundle com.company.missing.feature 0.0.0' but it could not be
#     found
#     Cannot satisfy dependency:
#     From: My Feature Name 1.0.0 (com.company.feature.group 1.0.0)
#     To: com.company.bundle [1.0.0]
#     Cannot satisfy dependency:
#     From: Bundle Name 1.0.0 (com.company.bundle 1.0.0)
#     To: bundle com.company.other.bundle 1.0.0
#
#   Regular expression find/replace property pairs can be specified using the
#   following format:
#
#   eclipse.p2.missingRequirement.find.0=
#   eclipse.p2.missingRequirement.replace.0=
#   eclipse.p2.missingRequirement.find.1=
#   eclipse.p2.missingRequirement.replace.1=
#   ...
#
#   If you want to display a message when a particular requirement (bundle,
#   package, etc.) is not found, you can use the following find expression:
#
#     eclipse.p2.missingRequirement.find.0=.*Software being installed:([_A-Za-z0\
#     -9-\\. ]+) ([_A-Za-z0-9-\\. ]+)\\(.*requires '(?:bundle |package |)\
#     <ID OF MISSING REQUIREMENT>.*
#
#   Where <ID OF MISSING REQUIREMNT> is replaced with the identifier of the
#   required feature/bundle.
#
#   Example:
#     eclipse.p2.missingRequirement.find.0=.*Software being installed:([_A-Za-z0\
#     -9-\\. ]+) ([_A-Za-z0-9-\\. ]+)\\(.*requires '(?:bundle |package |)\
#     com.product.bundle.*
#     eclipse.p2.missingRequirement.replace.0=$1 requires Some Product.  Please\
#      install it first.
#
#     The following will display a message that 'Some Product' is required when
#     any bundle starting with 'com.product.bundle' is found missing.
#
#     eclipse.p2.missingRequirement.find.0=.*
#     eclipse.p2.missingRequirement.replace.0=Requirements not found.\
#     Installation can't continue.
#
#     This will display the same message or any missing requirement.
################################################################################
#eclipse.p2.missingRequirement.find.0=.*Software being installed:([_A-Za-z0-9-\\. ]+) ([_A-Za-z0-
9-\\. ]+)\\(.*requires '(?:bundle |package |)INSERT ID OF REQUIRED BUNDLE HERE.*
#eclipse.p2.missingRequirement.replace.0=$1 requires INSERT MISSING PRODUCT NAME HERE.  Please
install it first.

################################################################################
# Include Remote Repositories (Optional)
#   By default, the installer will only use local repositories during
#   installation.  To include remote repositories, set this property to 'true'.
#   Note, this can increase delays significantly as remote repositories are
#   contacted.
################################################################################
#eclipse.p2.includeAllRepositories=true

################################################################################
# Ordered Planner (Optional)
#   By default, the installer will use an ordered planner instead of the
#   default P2 planner.  This planner ensures that any IUs that have
#   dependencies are installed after the IUs they depend on.
#   The default P2 planner will be used if this property is set to 'false'.
################################################################################
#eclipse.p2.orderPlanner=false

################################################################################
# Installer Data Location (Optional)
#   This property can be used to specify the directory to use for installer
#   data.  If this property is not specified, the installer will use the default
#   ~/.p2_installer or the path specified using the -install.data command line
#   option.
#   Use '~' to specify a location based on the operating system specific home
#   directory.
################################################################################
#eclipse.p2.location.data=~/.p2_installer

################################################################################
# Window Title (Optional)
#   Specify the window title of the installer.  If this property is not
#   specified, the product name will be used.
################################################################################
eclipse.p2.window.title=${eclipse.p2.product.name} Setup

################################################################################
# Installer Title Image (Optional)
#   Optional image to display in the install wizard title area.  The image file
#   should be included with the installer.
#   If this property is not specified then the image provided by a
#   com.codesourcery.installer.icon extension point will be used if available.
#   If no extension is registered, a default title image will be used.
#
#   Example:
#     eclipse.p2.wizard.image=./title.png
################################################################################
#eclipse.p2.wizard.image=

################################################################################
# Installer Title (Optional)
#   Optional property to specify the title area text that is displayed for all
#   wizard pages in the installer.  If this property is not specified, no title
#   will be displayed.
################################################################################
#eclipse.p2.wizard.title=Software Setup

################################################################################
# Product Information (Optional)
#   Optional file containing additional information to display for the product.
#   If this property is specified, the installer will display an information page
#   with the contents of the file.  The path to the file is relative to the
#   installer properties file.
#   Text in the file can be made bold using the <b> tag.  For example,
#     "This is <b>bold</b> text."
#   Text in the file can be made italic using the <i> tag.  For example,
#     "This is <i>italic</i> text."
################################################################################
#eclipse.p2.wizard.text.information=./information.txt

################################################################################
# Welcome Text (Optional)
#   Optional property to set text to display on the wizard Welcome page.
#   The text can be made bold by enclosing in the <b> and </b> tags.
#   The text can be made italic by enclosing in the <i> and </i> tags.
#   The text can be made smaller by enclosing in the <small> and </small> tags.
#   The text can be made bigger by enclosing in the <big> and </big> tags.
#   If this property is not specified, default text will be displayed.
#   Example:
#     Welcome to the <b>${eclipse.p2.product.name}</b> installation wizard.
################################################################################
#eclipse.p2.wizard.text.welcome=

################################################################################
# Install Folder Text (Optional)
#   Optional property to set text to display on the wizard Install Folder page.
#   If this property is not specified, the default text will be displayed.
################################################################################
#eclipse.p2.wizard.text.installFolder=Where would you like to install?

################################################################################
# Update/Upgrade text (Optional)
#   Optional property to set the text that is displayed on the wizard Setup
#   page when the product is already installed.  If this property is not
#   specified, the default text will be displayed.  The variable,
#   ${eclipse.p2.product.existingVersion}, can be used and will be substituted
#   with the existing version of the  product found.
################################################################################
#eclipse.p2.wizard.text.existingInstall=${eclipse.p2.product.name} version
${eclipse.p2.product.existingVersion} is already installed.

################################################################################
# Category install text (Optional)
#   Optional property to set the text that is displayed on the wizard Setup
#   page when choosing to install into existing products with the same
#   category.  If this property is not specified, the default text will be
#   displayed.
################################################################################
#eclipse.p2.wizard.text.categoryInstall=You can choose to install ${eclipse.p2.product.name} into
one of the existing installations found or a different location.

################################################################################
```

```
# Install Summary Information Text (Optional)
#   Optional property to set additional text that is displayed in the wizard
#   after the product is installed.  This text will be displayed on the wizard
#   Results page.
################################################################################
#eclipse.p2.wizard.text.install.addendum=

################################################################################
# Uninstall Summary Information Text (Optional)
#   Optional property to set additional text that is displayed in the uninstall
#   wizard when the product is uninstalled.  This text will be displayed on the
#   wizard Results page.
################################################################################
#eclipse.p2.wizard.text.uninstall.addendum=

################################################################################
# Mirror Update Progress Text (Optional)
#   Optional property to set the progress text when the install mirror
#   repository is being updated.
################################################################################
#eclipse.p2.wizard.text.progress.mirroring=Saving...

################################################################################
# Mirror Wizard Page Message Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.message=You can select to also save the install so that it can
be performed later.  If you have already saved an installation, you can choose to use it now.

################################################################################
# Mirror Wizard Page Install Section Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.section.install=Install

################################################################################
# Mirror Wizard Page Install Option Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.install=Install

################################################################################
# Mirror Wizard Page Save Section Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.section.save=Save

################################################################################
# Mirror Wizard Page Save Option Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.save=Save data and install later.

################################################################################
# Mirror Wizard Page Load Option Text (Optional)
################################################################################
#eclipse.p2.wizard.text.mirrorPage.load=Install from saved data.

################################################################################
# Mirror Wizard Page Default Directory (Optional)
#   Specifies the default directory for saving and loading installation mirror
#   repositories.  Use '~' to specify a location based on the operating system
#   specific home directory.
################################################################################
#eclipse.p2.wizard.text.mirrorPage.defaultDirectory=~/Downloads/${eclipse.p2.product.id}_${eclips
e.p2.product.version}

################################################################################
# Name of progress page when installing (Optional)
################################################################################
#eclipse.p2.wizard.text.progressPage.name.installing=Installing...

################################################################################
# Name of progress page when mirroring (Optional)
################################################################################
#eclipse.p2.wizard.text.progressPage.name.mirroring=Saving...
```

```
################################################################################
# Progress page message when installing (Optional)
################################################################################
#eclipse.p2.wizard.text.progressPage.message.installing=Please wait while
${eclipse.p2.product.name} is installed.

################################################################################
# Progress page message when mirroring (Optional)
################################################################################
#eclipse.p2.wizard.text.progressPage.message.mirroring=Please wait while
${eclipse.p2.product.name} is saved.

################################################################################
# Result message displayed when installation is complete (Optional)
################################################################################
#eclipse.p2.wizard.text.result.instal=Download of <b>${eclipse.p2.product.name}</b> is complete.

################################################################################
# Result message displayed when mirror is complete (Optional)
################################################################################
#eclipse.p2.wizard.text.result.mirror=Save of <b>${eclipse.p2.product.name}</b> is complete.

################################################################################
# Text displayed on the "Finish" button for an install. (Optional)
################################################################################
#eclipse.p2.wizard.text.finish.install=Install

################################################################################
# Text displayed on the "Finish" button for a mirror operation. (Optional)
################################################################################
#eclipse.p2.wizard.text.finish.mirror=Save

################################################################################
# Text displayed for the message on the Shortcuts page. (Optional)
################################################################################
#eclipse.p2.wizard.text.shortcutsPage.message=Shortcuts to important programs and files will be
created in the following locations:

################################################################################
# Text displayed for the message on the Paths page. (Optional)
################################################################################
#eclipse.p2.wizard.text.pathsPage.message=The wizard can add ${eclipse.p2.product.name} to the
PATH environment variable.  This will make it easier to use {0} and will mean less\nconfiguration
for other tools.  Note: Read-only environment files will not be changed.\n\n<b>WARNING:</b>
Without this change, some tools may not function.

################################################################################
# Launch (Optional)
#   Specifies items to launch after installation is complete.  If this property
#   is specified, the installer will display a page to choose items to launch
#   after installation.
#   Separate multiple items with a comma.
#   The format for an item is:
#     <name>;<path>;<type>;<default>
#   where
#     <name> is the name to display
#     <path> is the path to the file that should be launched.  This path can be
#       a file relative to the product installation directory
#       (eclipse.p2.location.root), or an HTML address.
#       If the item is an executable binary, append {exe} that will be replaced
#       with ".exe" on Windows and nothing on Linux.
#       If the item type is "restart" or "logout" then this parameter is
#       ignored.
#     <type> is
#       exe     - Specifies an executable to run.
#       file    - Specifies a file to open.
#       html    - Specifies a web page to open the browser.
#       restart - Restarts the computer and disable all other items if selected.
#                 This option is only displayed if any action is run that
#                 requires it.
#                  Only supported on Windows.
#       logout  - Logs out the user and disables all other items if selected.
#                  This option is only displayed if any action is run that
#                  requires it.
#     <presentation> is optional, and may be one of the following:
#       checked (or equivalently may also be "true") - displays the item as a
#                                                      checkbox which is checked
#                                                      by default. This is the
#                                                      default value if the
#                                                      presentation is omitted
#                                                      from the property
#       unchecked (or equivalently may also be "false") - displays the item as a
#                                                         checkbox which is unchecked
#                                                         by default.
#       link - displays the item as a HTML style link. Note that a link cannot
#              be used with the restart or logout types.
#  Only one 'restart' or 'logout' item can be specified.  If more than one is
#  specified, only the first will be used.  If an action runs that requires a
#  restart and restart or logout item is not specified, a default item will be
#  displayed.
#
#  If the launch item refers to an actual file, and if the file does not exist, the
#  launch item will not be displayed in the GUI.
#
#   Examples:
#     Run ${eclipse.p2.product.name}?;bin/program{exe};true
#     Visit Eclipse.org?;http://www.eclipse.org/;html;false
#     Restart computer so changes can take effect?;;restart;true
#
###############################################################################
#eclipse.p2.wizard.launch=
#
###############################################################################
# launch presentation (Optional)
#   Specifies launch types to display with a specific presentation.  If this
#   property is used, the installer will display all specified launch types
#   using the presentation.
#
#   Normally if the presentation of the launch is not specified it defaults
#   to a presentation of checked. These properties allow to specify default
#   for different launch types.
#
#   The list of types must be comma separated.
#
#   Note that restart and logout types may not be specified as links.
#
#   Examples:
#     eclipse.p2.wizard.launch.presentation=html:link,exe:checked
###############################################################################
#eclipse.p2.wizard.launch.presentation=html:link,file:link
#
###############################################################################
# Wizard Navigation (Optional)
#   Specifies the type of wizard page navigation bar.  This property can be one
#   of the following values:
#   none - Don't show a page navigation bar
#   top  - Show a stepped page navigation bar at the top of the wizard
#   left - Show a bulleted page navigation bar on the left of the wizard
#   left_minimal - Show a simple page navigation bar on left of the wizard (2nd scheme)
###############################################################################
org.eclipse.p2.wizard.navigation=top

###############################################################################
# Exclude Wizard Pages (Optional)
#  Specifies pages that should be excluded in the wizard.  By default, wizard
#  pages from all included install modules are displayed.  If this property is
#  specified, any pages listed will not be shown.
#  Separate page names with a comma.
#  The names of common pages are:
#    welcomePage, licensePage, informationPage, installFolderPage,
#    componentsPage, shortcutsPage, addonsPage, and pathsPage.
```

```
###############################################################################
#eclipse.p2.wizard.pages.exclude=

###############################################################################
# Wizard Page Order (Optional)
#  Specifies the order that install wizard pages should be displayed.  Separate
#  page names with a comma.  The pages will be displayed in the order that they
#  are listed in this property.  Any page not listed will be displayed after the
#  pages specified.  Any page listed that does not exist will be skipped.
#  The names of common pages are:
#    welcomePage, licensePage, informationPage, installFolderPage,
#    componentsPage, shortcutsPage, addonsPage, and pathsPage.
#  Note: Some wizards pages are designed to be shown after certain other pages.
###############################################################################
#eclipse.p2.wizard.pages.order=

###############################################################################
# Page Titles (Optional)
#  Specifies titles for wizard pages.  Separate multiple pages with a comma.
#  If titles for pages are not specified, the default titles will be used.
#  Specify the page name followed by the title to display separated by a colon.
#  <page name>:<page title>,...
#
#  Example:
#   licensePage:Software Agreements
###############################################################################
#eclipse.p2.wizard.pages.titles=

###############################################################################
# Data Defaults (Optional)
#  This property can be used to specify default values for install data.
#  Install data is used by wizard pages and install actions.
#  The format for an install data default is:
#
#    eclipse.p2.default.<property name> = <property value>
#
#   where
#     <property name> is the name of the install data property used by
#     a wizard page and/or install action.
#     <property value> is the value for the property.
#
#   Example: To set the Short-cuts page to not create desktop short-cuts by
#   default, the following property could be set:
#     eclipse.p2.default.createDesktopShortcuts = false
###############################################################################
#eclipse.p2.default.setPATH=false
#eclipse.p2.default.createDesktopShortcuts=false
#eclipse.p2.default.createProgramShortcuts=false

###############################################################################
# Install Registry (Optional)
#  By default, the installer tracks installations using an install registry
#  located in the installer data area.  This property can be used to enable or
#  disable the use of the registry.
###############################################################################
#eclipse.p2.useInstallRegistry=false

###############################################################################
# Network Time-out (Optional)
#  Specifies a time-out for network operations (in millisconds).
###############################################################################
#eclipse.p2.network.timeout=120000

###############################################################################
# Network Retry Attempts (Optional)
#  Specifies number of times to retry network operations.
###############################################################################
#eclipse.p2.network.retry=10
```