Getting Started with

nWire for Java

nWire for PHP

Version 1.3
May 2010

Preface

About this document

Welcome to nWire. This document outlines the main functionality and features of nWire. We welcome your feedback for improving nWire and this document. If you have any questions or comments, please let us know by sending an email to support@nwiresoftware.com or by using the feedback option under the help menu.

This document covers both **nWire for Java** and **nWire for PHP**. Both products are very similar, however, some sections in this document might be relevant to a specific product. The screenshots in this document were taken on Mac OS X. nWire supports Windows and Linux as well.

Terminology

The following terms are used throughout this document:

- **Resource** - a file which contains code.
- **Component** - a recognized entity in the code, e.g. a class, interface, method or field. This is sometimes referred to as an **artifact**.
- **Association** - a connection, implicit or explicit, between two components, e.g. class extends another class, method contained in a class, methods invokes another method, etc.
- **Model** - the collection of components and associations.
- **Metamodel** - the definition of component types and association types that nWire can recognize. The nWire metamodel is composed of component types (e.g. Java class), association types (e.g. Java extension) and resource types (e.g. Java source file).
- **Repository** - a data store which holds the model and metamodel.
- **Eclipse** - nWire is an Eclipse plug-in. nWire can be installed on Eclipse based products such as Zend Studio and MyEclipse. Throughout this document the name “Eclipse” is used as a general term for the base IDE.

Getting More Information

For more information, please use the following sources:

- Our Support knowledge base as: http://www.nwiresoftware.com/support/kb
- Our discussion forums at: http://forum.nwiresoftware.com/

Installing nWire

Before You Begin

Please make sure you have an Eclipse IDE which is properly installed and configured. Eclipse can be downloaded from the Eclipse.org web site at http://www.eclipse.org/.

You may also choose to install nWire through Pulse. Pulse is a free Eclipse distribution with simplified installation and configuration capabilities. This is especially recommended for first-time Eclipse users. For mode details visit http://www.poweredbypulse.com/.

Requirements

nWire is **tested** on the following operating systems:

- Windows XP Professional, Windows Vista
- Mac OS X
- Linux Ubuntu

Since nWire is based on Java and Eclipse, it is designed to run on any operating system that can run Eclipse and has an installed web browser. If you have any questions regarding a specific operating system, please contact our support team.

nWire for Java requires:

- Eclipse 3.4 (Ganymede) or Eclipse 3.5 (Galileo, recommended).
- Java Development Tools (JDT) for Eclipse.

nWire for PHP requires:

- Eclipse 3.5 (Galileo) or Zend Studio 7.x
- PHP Development Tools (PDT) for Eclipse, version 2.1 or above.

Both products can be installed on the same Eclipse IDE, provided the above requirements are met.

nWire adds some memory and CPU overhead to your Eclipse IDE.
A strong machine with at least 2 GB of RAM is recommended.
It is not recommended to use nWire on a Netbook.

The Installation & Setup Process

nWire installation is done in 3 main steps:

1. Installing the nWire plugin in the Eclipse IDE.
2. Activating the nWire plugin by entering a license key or initiating a free 30-day trial.
3. Initial analysis: nWire will analyze your entire code base. See the section Understanding the nWire Code Analysis for more details.

The initial analysis may take up to several hours in a large workspace or on a slow machine. We recommend installing nWire at the end of your workday, leaving the analysis overnight. See the analysis chapter for more details.

Choosing an Installation Technique

There are 3 main options for installing plugins in Eclipse:

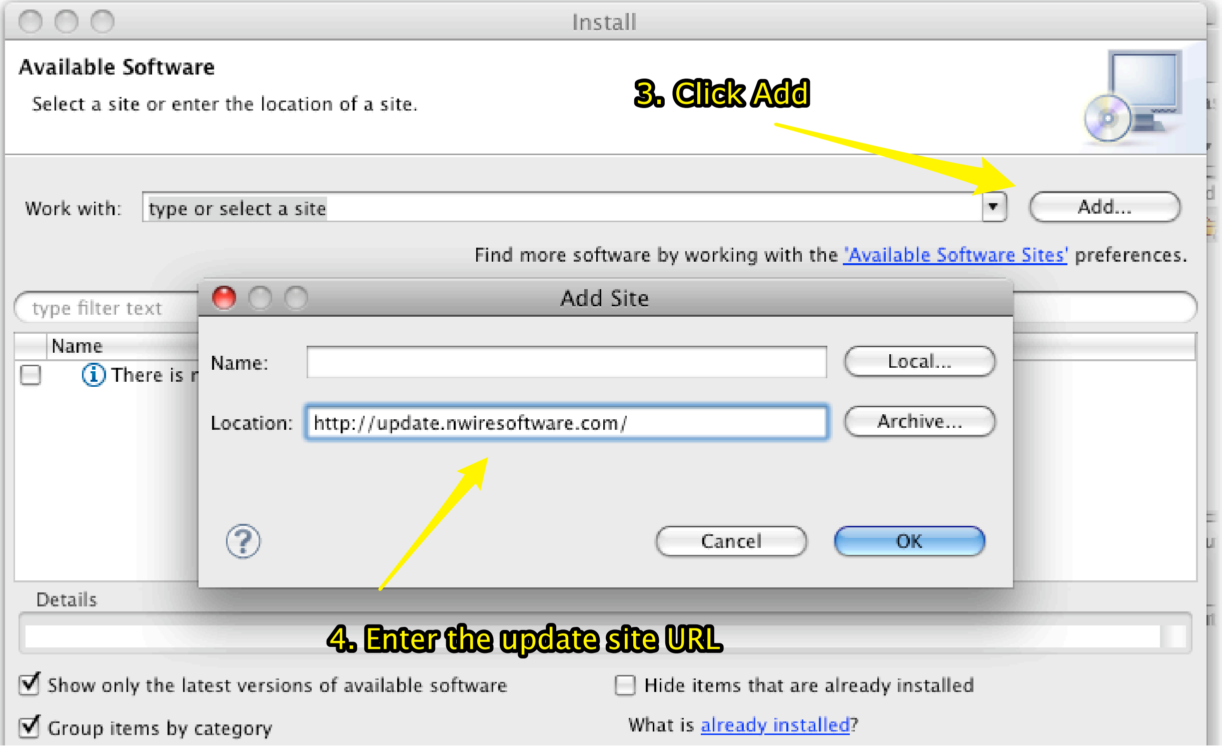
1. **Using the “Dropins Folder”:** this folder is located with the Eclipse executable. Upon launch, Eclipse will scan this folder for new plugin and install them automatically.
2. **Using the Update Manager from an online site:** Eclipse has a built-in mechanism for installing plugins, which connects to a remote site and installs the plugin online.
3. **Using the Update Managed from an archived site:** this is essentially the same as the first method, but here, the site is archived and downloaded to the client machine before the installation.

From the above methods, the update manager method is recommended from several reasons:

- Installing from an online site means getting the most updated software.
- If a prerequisite is missing, Eclipse will install it automatically.
- It is easier to troubleshoot in case of installation errors.

Installing using the Update Manager on Eclipse 3.5 (Galileo) or Zend Studio 7.x

1. Start your Eclipse IDE.
2. In Eclipse, select **Help → Install New Software**.
3. In the **Available Software** dialog box, select **Add Site...**
4. In the **Location** field, type the URL: http://update.nwiresoftware.com/ and click **OK**.



Install
Available Software
Select a site or enter the location of a site.
Work with:
type or select a site
Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text
Name
There is r
Add Site
Name:
Local...
Archive...
Cancel
OK
Details
What is already installed?

Location: http://update.nwiresoftware.com/

☑ Show only the latest versions of available software

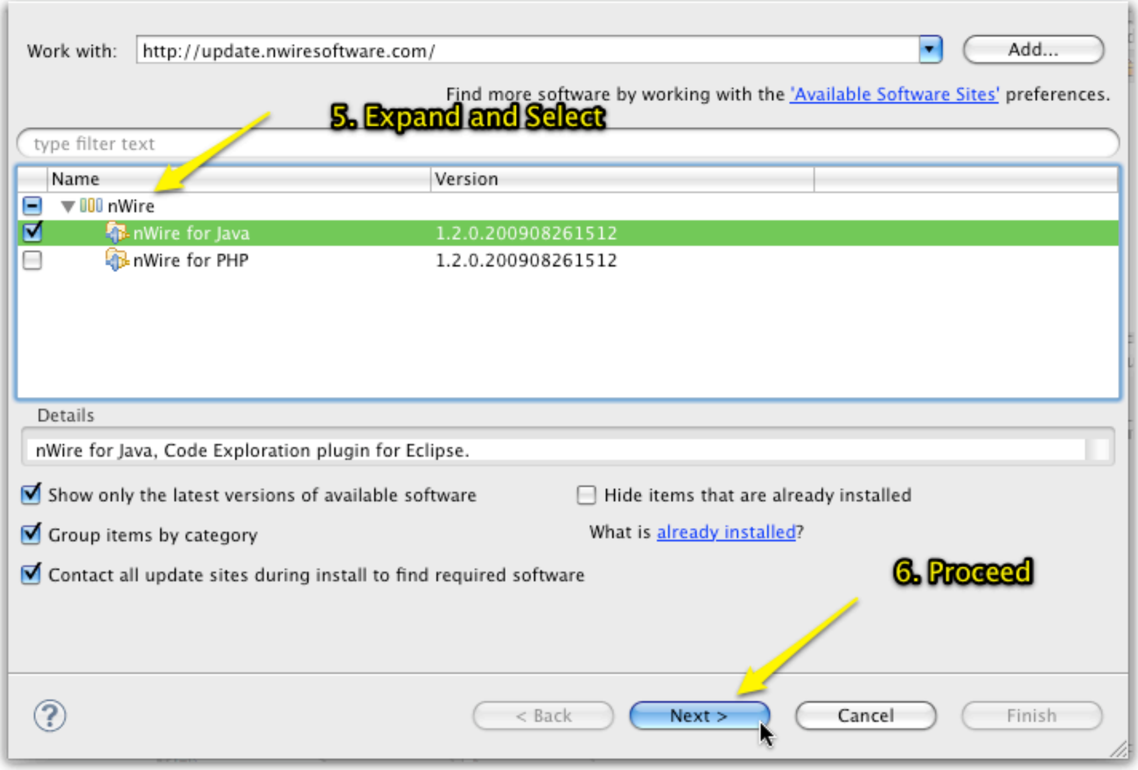
☐ Hide items that are already installed

☑ Group items by category

3. Click Add

4. Enter the update site URL

5. Expand and select the check box for the desired features:
   nWire For PHP and/or **nWire For Java**.
6. Click the **Next >** button, and follow the instructions of the installation wizard.



Work with: http://update.nwiresoftware.com/

Find more software by working with the 'Available Software Sites' preferences.

type filter text

	Name	Version
⊟	nWire	
☑	nWire for Java	1.2.0.200908261512
☐	nWire for PHP	1.2.0.200908261512

Details

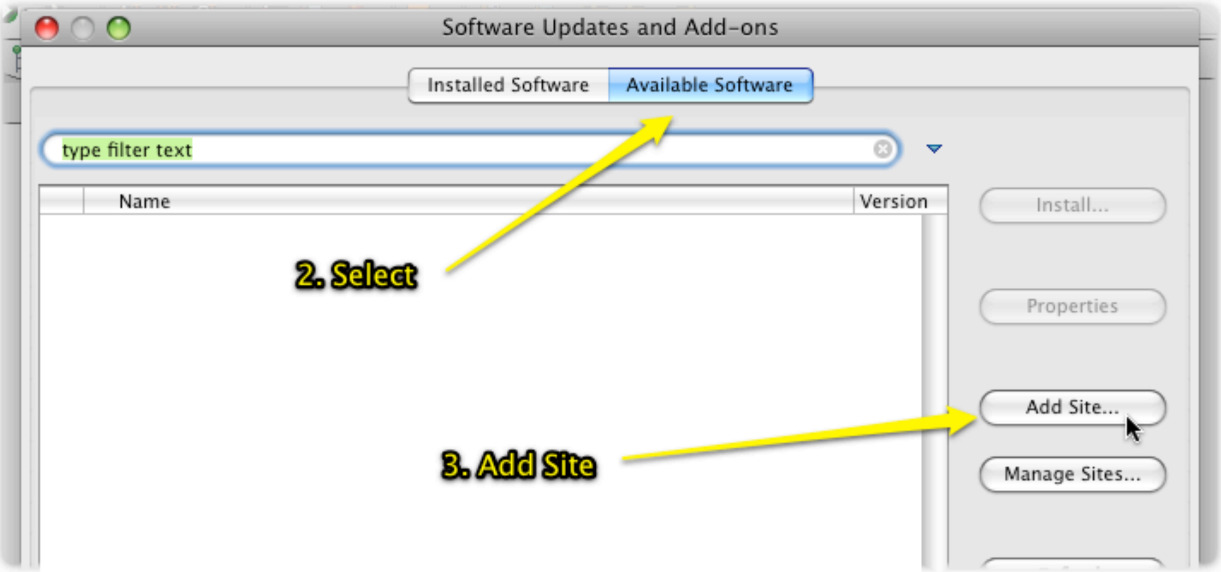
nWire for Java, Code Exploration plugin for Eclipse.

☑ Show only the latest versions of available software
☑ Group items by category
☑ Contact all update sites during install to find required software
☐ Hide items that are already installed
What is already installed?

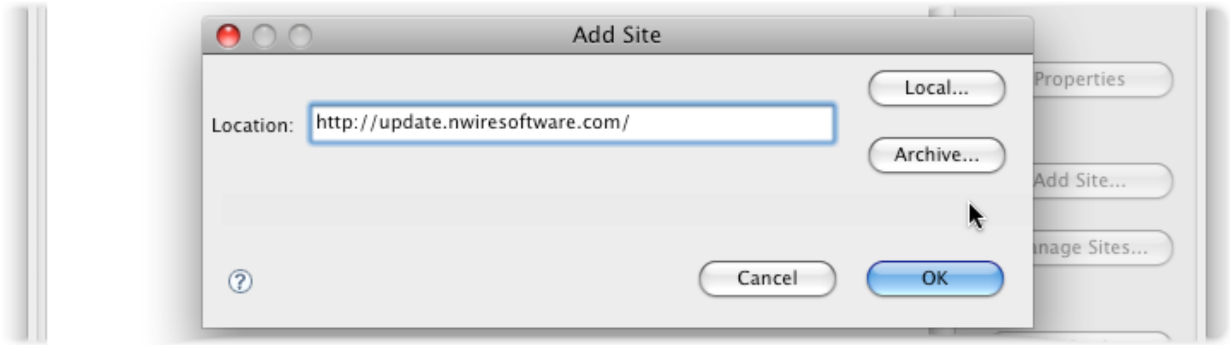
7. Once the process is complete, you will be asked to restart your Eclipse IDE. After Eclipse restarts, continue to the activation section.

Installing using the Update Manager on Eclipse Ganymede (3.4)

1. In Eclipse, select **Help → Software Updates**.
2. On the Available Software tab, select **Add Site...**

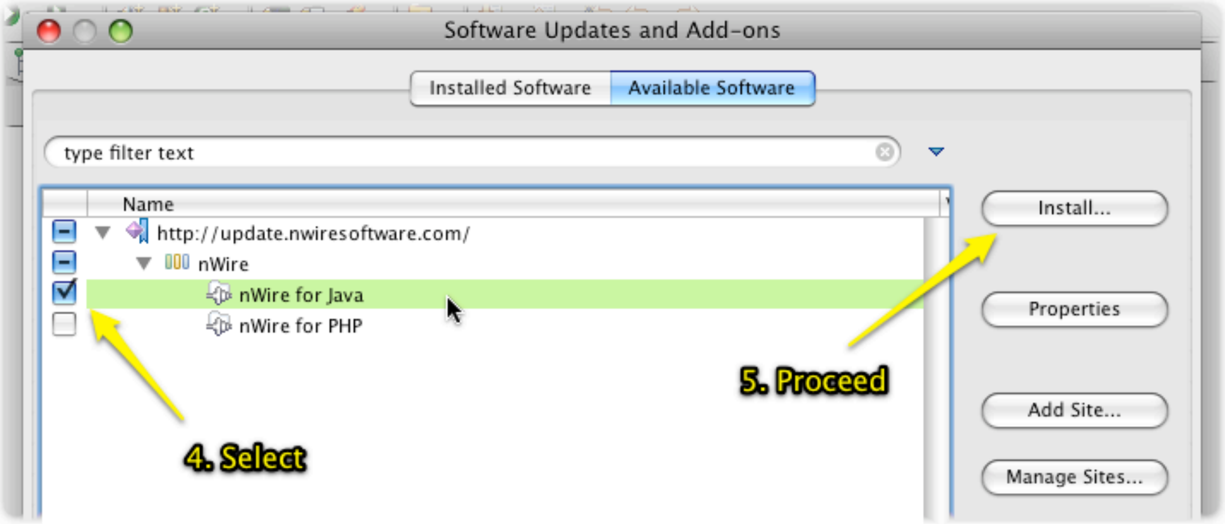


3. Enter the URL: http://update.nwiresoftware.com/.



Location: http://update.nwiresoftware.com/

4. Expand and select the check box for the desired features: **nWire For Java**. nWire for PHP does not support Eclipse 3.4.
5. Click the Install button, this may take a some time.



- http://update.nwiresoftware.com/
  - nWire
    - ☑ nWire for Java
    - ☐ nWire for PHP

6. Follow the instructions of the installation wizard.
7. Once the process is complete, you will be asked to restart your Eclipse IDE. After Eclipse restarts, continue to the activation section.

Installing using an Update Site Archive

1. Download the update site archive (JAR file) from the following URL: http://www.nwiresoftware.com/download/offline-installation
2. Follow the installation instructions above for an online update site. Instead of entering the URL, click on **Archive** and select the downloaded archive.

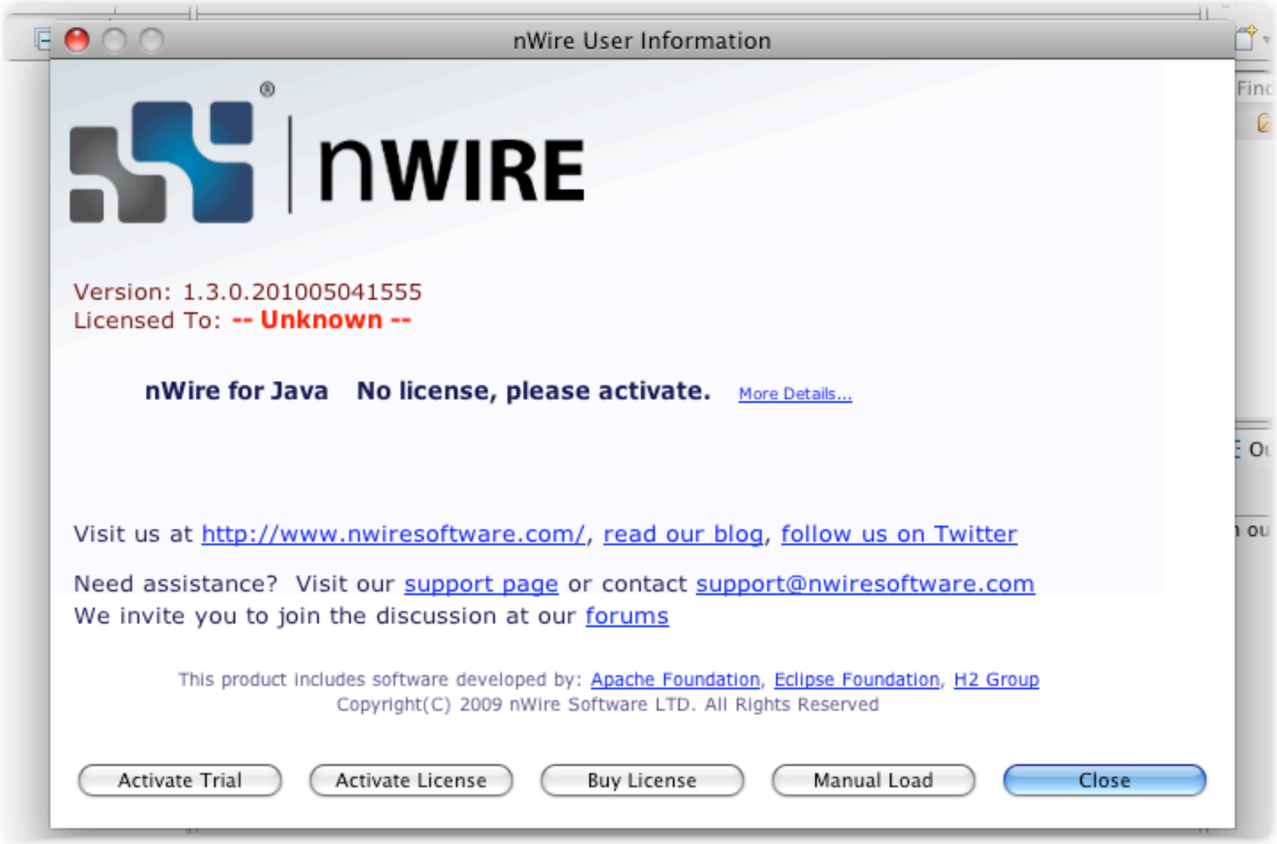
Installing using the “Dropins Folder”

1. Download the nWire dropins archive (ZIP file) from the following URL: http://www.nwiresoftware.com/download/download-now
2. Close your Eclipse IDE.
3. Locate the dropins folder: it is usually found in the same folder as your Eclipse executable.

4. Copy the nWire zip file to the dropins folder.
5. Restart your Eclipse IDE.
6. After the restart, the nWire activation process will begin. If not, please try starting your eclipse from the command line with the **-clean** argument.

Activating nWire

Following the installation process, you will be presented with a dialog box asking you to activate nWire. This is the **nWire User Information** dialog box. It will be opened automatically if nWire is not activated and it can be reached by selecting **Help → nWire → User Info...**



nWire User Information

Version: 1.3.0.201005041555
Licensed To: **-- Unknown --**

nWire for Java **No license, please activate.** More Details...

Visit us at http://www.nwiresoftware.com/, read our blog, follow us on Twitter

Need assistance? Visit our support page or contact support@nwiresoftware.com
We invite you to join the discussion at our forums

This product includes software developed by: Apache Foundation, Eclipse Foundation, H2 Group
Copyright(C) 2009 nWire Software LTD. All Rights Reserved

The following options are available:

- Click on **Activate Trial** to start your **30 day free trial**.
- Click on **Buy License** to visit our online shop.
- Click on **Activate License** to enter a valid license key.
- Click on Manual Load to load a license file (see hereunder).

The activation process will connect with the **nWire Activation Server** to obtain a license file. The information sent includes the machine physical network address and the nWire version. Entering your personal details is optional. nWire maintains a very strict privacy policy. For more details visit http://www.nwiresoftware.com/legal/privacy-policy.

Activating nWire Offline

The activation process requires a working internet connection. nWire can be activated without at active connection, by following this procedure:

1. Click on **Activate Trial** or **Activate License** and proceed like a regular activation.
2. nWire will fail to connect and a dialog box containing an encoded URL will be shown.
3. Copy the URL and save it to a text file.
4. Open a browser on **any other internet-enabled machine**, open the text file and enter the URL in any web browser.
5. Save the resulting page: this is your license file.
6. Use the **Manual Load** feature to load the license file.

Keeping up to Date

nWire is updated regularly with new features and updates. To keep track of new versions, we recommend subscribing to our product blog at: http://www.nwiresoftware.com/blogs/nwire.

Another important resource for news regarding nWire is our mailing list. To join, simply enter your details when activating nWire.

Updating nWire can be done automatically or manually with the Eclipse built-in update feature.

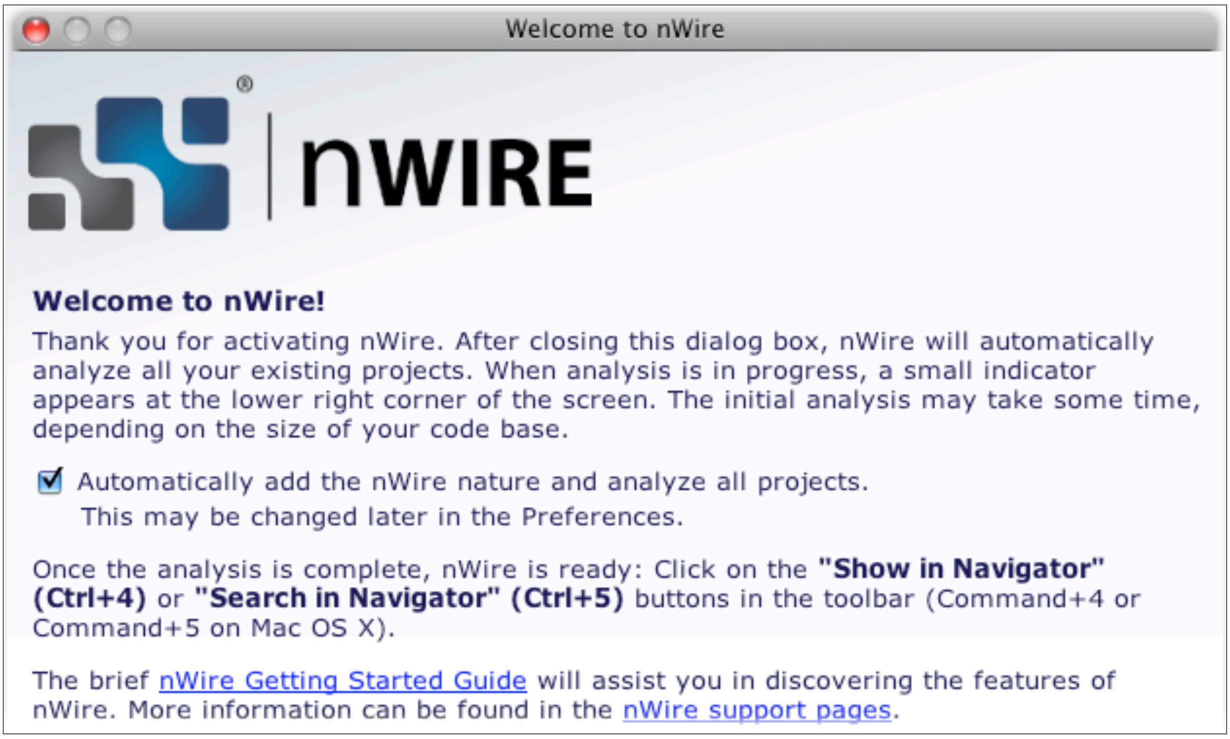
*Checking for updates in Eclipse 3.5: Select **Help → Check for Updates***
In Eclipse 3.4: follow the installation procedure described earlier.
*Update check can also be automated, open the **Preferences** and look for **Install/Update → Automatic Updates***

After updating nWire to a new version a **release notes** dialog box will open, highlighting the changes in the new version. This dialog box appears only once and can also be accessed by selecting: **Help → nWire → Release Notes...**

Getting Started

Welcome to nWire

After completing the activation process, the “Welcome to nWire” dialog will be shown:



Welcome to nWire!

Thank you for activating nWire. After closing this dialog box, nWire will automatically analyze all your existing projects. When analysis is in progress, a small indicator appears at the lower right corner of the screen. The initial analysis may take some time, depending on the size of your code base.

☑ Automatically add the nWire nature and analyze all projects.
This may be changed later in the Preferences.

Once the analysis is complete, nWire is ready: Click on the **"Show in Navigator" (Ctrl+4)** or **"Search in Navigator" (Ctrl+5)** buttons in the toolbar (Command+4 or Command+5 on Mac OS X).

The brief nWire Getting Started Guide will assist you in discovering the features of nWire. More information can be found in the nWire support pages.

Press the link to the **nWire Getting Started Guide** to open this document. The guide is available as part of the nWire installation and no internet connection is required to open the link.

This dialog has a checkbox for enabling the **Auto Analyze** feature. When this feature is on, all project will be automatically analyzed and available through nWire. It is recommended that you keep this option on.

The **Welcome to nWire** dialog box containing this guide can be accessed at any time by selecting: **Help → nWire → Getting Started...**

nWire in Action

nWire works by analyzing your code. The first analysis will scan the entire code base and create an initial repository. This process can take up to several hours depending on the amount of code and the machine. For more details on the nWire analysis process, see the section Understanding the nWire Code Analysis.

After the analysis is complete, nWire is ready for use. The nWire repository is accessed through one main tool: **the navigator**. To open the navigator, open any source file and click on the **Show in Navigator** button in the toolbar. For more details on the nWire navigator, see the section The nWire Navigator.

Understanding the nWire Code Analysis

The Analysis Process

nWire operates by analyzing your code. It keeps all the information in a local repository. The repository is then being used to present all the relevant associations and components through the nWire tools: the **navigator** and **visualizer**.

Following the nWire installation, your code will undergo an **initial analysis** process. This process may take some time. Our tests show that the analysis rate is usually about 2-3 files per second, but your actual pace may vary, depending on the various parameters.

The nWire tools will not be available for source files which were not analyzed.

The nWire Repository

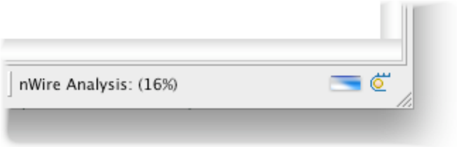
The repository files are located inside your workspace folder, under `.metadata/.plugins/com.nwiresoftware.nwire.core`. Depending on the size of your code base, the repository size may reach a few Gigabytes.

Some virus scanners scan files every time they are accessed. **It is very important for performance that the repository files are not scanned for viruses**. nWire uses the H2 database engine which is designed not to interpret the data stored in the database and, thus, cannot spread viruses. It is advised to exclude the repository directory from being scanned or exclude files with the extension `.db`.

There is no need to backup the repository: you can always reanalyze your code if the database is erased. If you are using an incremental backup solution it is recommended to exclude the repository folder from the backup set.

Continuous Analysis

The repository is updated every time a file is saved. The analysis does not block any operations, other than nWire itself. Changes in a single file may trigger changes in files which depend upon it, causing further analysis. After the analysis is complete the information in the nWire tools will be refreshed to reflect possible changes in the underlying model.



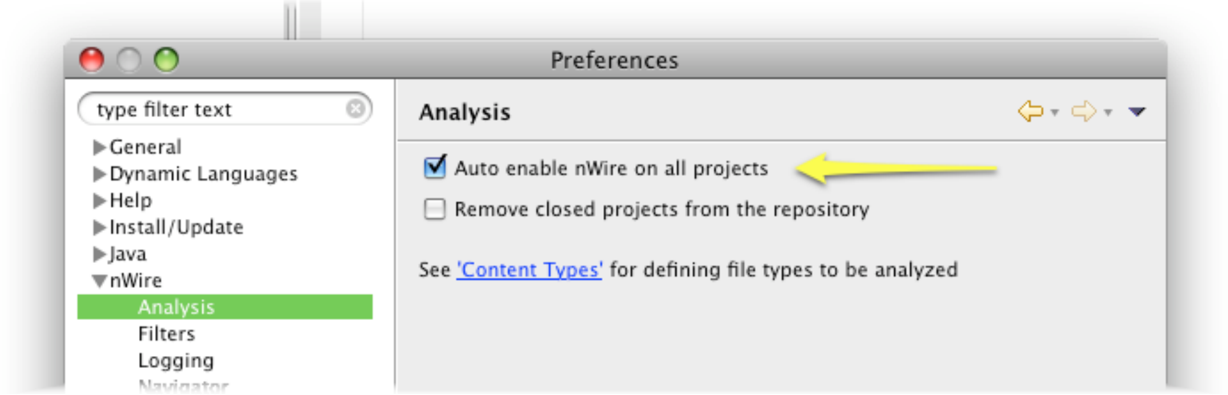
nWire Analysis: (16%)

The analysis process can be cancelled like any other Eclipse process. It will resume the next time a file is saved, including all the files which were left from the previous process.

It is recommended not to stop the analysis process.
This will result in an incomplete model.

What Sources are Analyzed?

nWire analyzes any source file in a relevant project which is **nWire Enabled**. The enablement can be automatic or manual. By default, relevant projects in your Eclipse IDE are automatically analyzed by nWire. This behavior can be changed from the nWire Analysis preferences pane.



Analysis

☑ Auto enable nWire on all projects

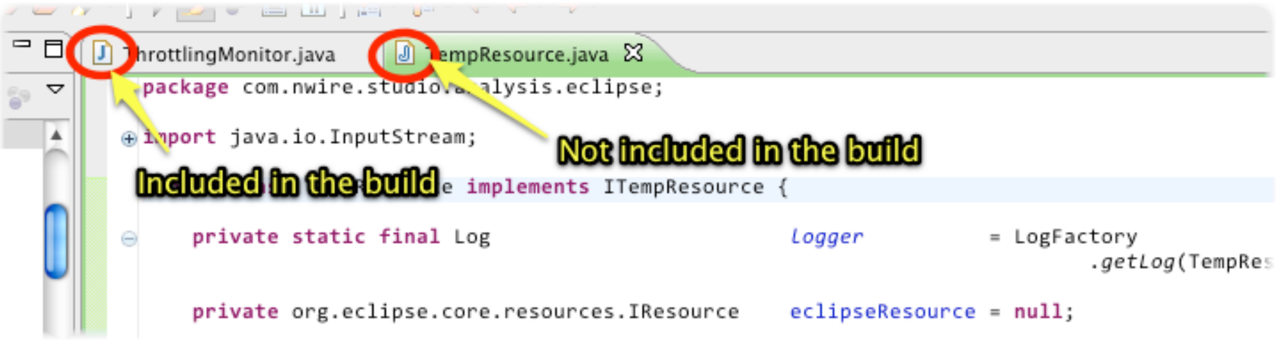
☐ Remove closed projects from the repository

See 'Content Types' for defining file types to be analyzed

Keep in mind that:

- nWire does not analyze sources in referenced libraries.
- If a source file fails to compile or validate, it will not be analyzed.
- nWire only analyzes files inside a designated source folders, which are part of the build. The source folders for a project can be determined in the project properties.

In Java, files which are not part of the build will get a hollow J icon as shown in the image below.



```
package com.nwire.studio.a alysis.eclipse;

import java.io.InputStream;

                    e implements ITempResource {

    private static final Log                          Logger          = LogFactory
                                                                          .getLog(TempRes

    private org.eclipse.core.resources.IResource      eclipseResource = null;
```

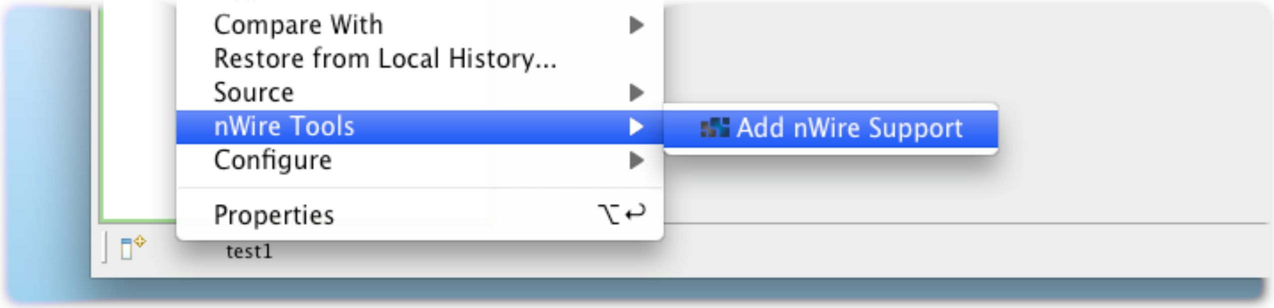
Not included in the build

Included in the build

Manually Enabling/Disabling nWire Analysis

You can manually remove or add nWire support for a given project:

1. Right click on the project in the **Package Explorer** or **PHP Explorer**.
2. Select **nWire Tools → Add nWire Support** or **Remove nWire Support**.

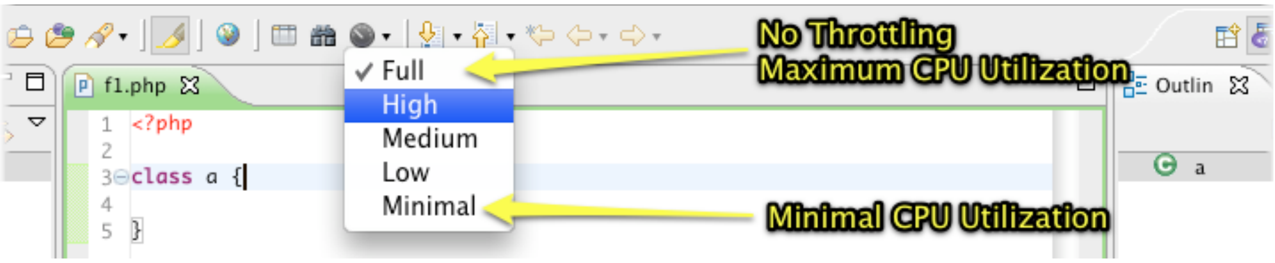


Once you manually remove the nWire support, it will not be added automatically, even if the auto-enable feature is on.

The above manual modification will add a nature to your Eclipse project indicating your selection. This means it will modify your `.project` file which contains your project settings.

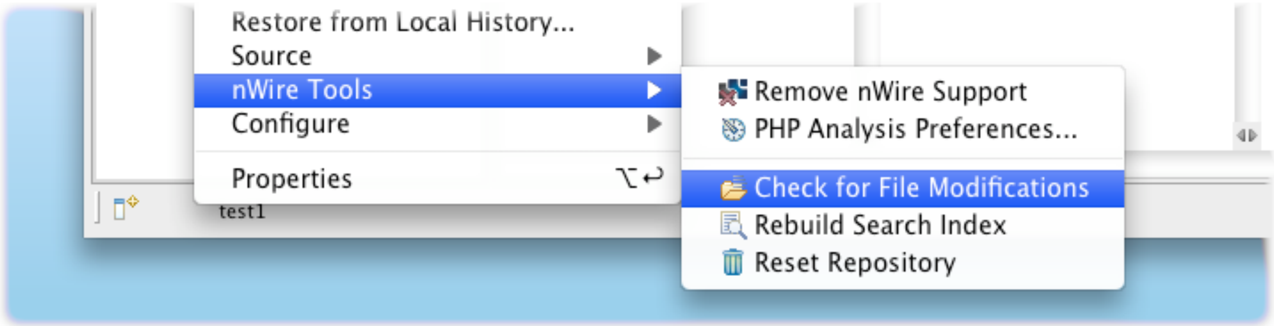
Throttling the CPU Utilization of the Analysis Process

It is possible to control the amount of CPU which is dedicated to the analysis process. This is useful if you want the analysis to run in the background without affecting your machines' performance. Using a lower throttle will result in longer analysis. The throttle can be set at any time, even during the analysis process, using the **nWire Analysis CPU Throttle** button in the toolbar.



Checking for File Modifications

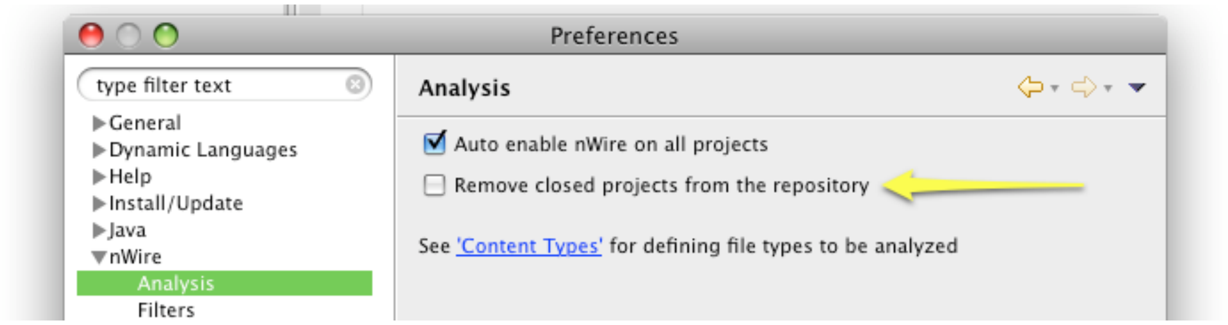
Although nWire is set to detect file changes within Eclipse, there may be cases where the model is not up to date. In this case, select **Check for File Modifications** from the **nWire Tools** menu. This menu appears when right-clicking a project.



When launching the Eclipse IDE, nWire will initiate such a check. This process usually takes just a few seconds if there are no modified files.

Closing Projects

Closing a project in Eclipse means that the project resources are not available within the IDE. In nWire, you may choose to remove the information of closed projects. This process may take up to a couple of minutes, depending on the size of the project. You may also choose to keep the closed project information. The default behavior is to keep the information.



Analysis

☑ Auto enable nWire on all projects

☐ Remove closed projects from the repository

See 'Content Types' for defining file types to be analyzed

Once the project is opened, nWire will check for file modifications within the project and re-analyze any files that might have been modified.

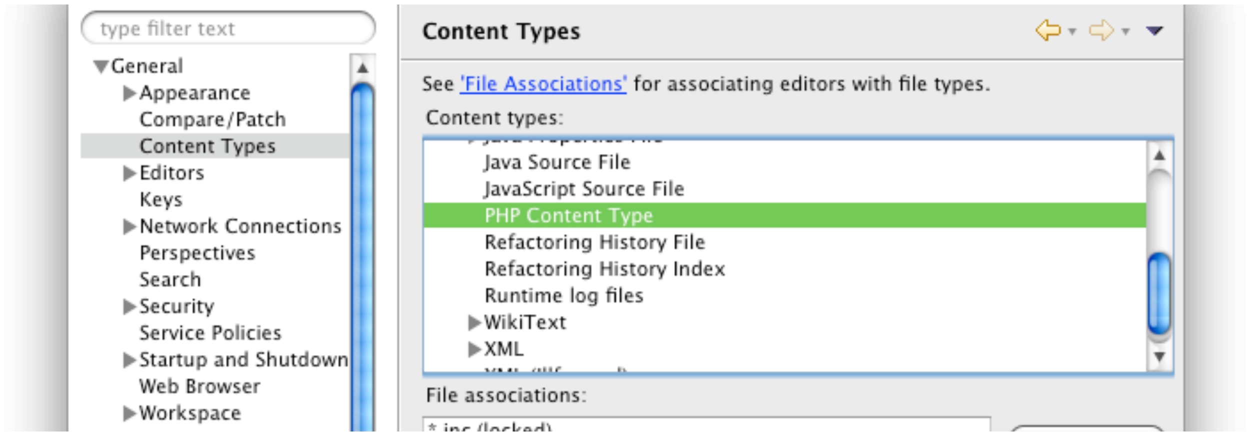
Setting Content Types for Analysis

By default, nWire recognizes source files according to their file extension.

- nWire for Java analyzes files with `.java` extension.
- nWire for PHP analyzes files with the following extensions: `php, php3, php4, php5, inc, phtml`.

The content types can be customized, by editing the Eclipse standard content types.

1. Open the Eclipse preferences.
2. Navigate to **General → Content Types**.
3. Locate the designated type:
   For Java, select **Text → Java Source File**.
   For PHP, select **Text → PHP Content Type**.
4. Click on **Add...**
5. Type an exact file name or use a wild-card (*) to identify files of a given type.
6. Click **OK**.



Content Types

See 'File Associations' for associating editors with file types.

Content types:

Java Source File
JavaScript Source File
PHP Content Type
Refactoring History File
Refactoring History Index
Runtime log files
WikiText
XML

File associations:

*.inc (locked)

The nWire Navigator

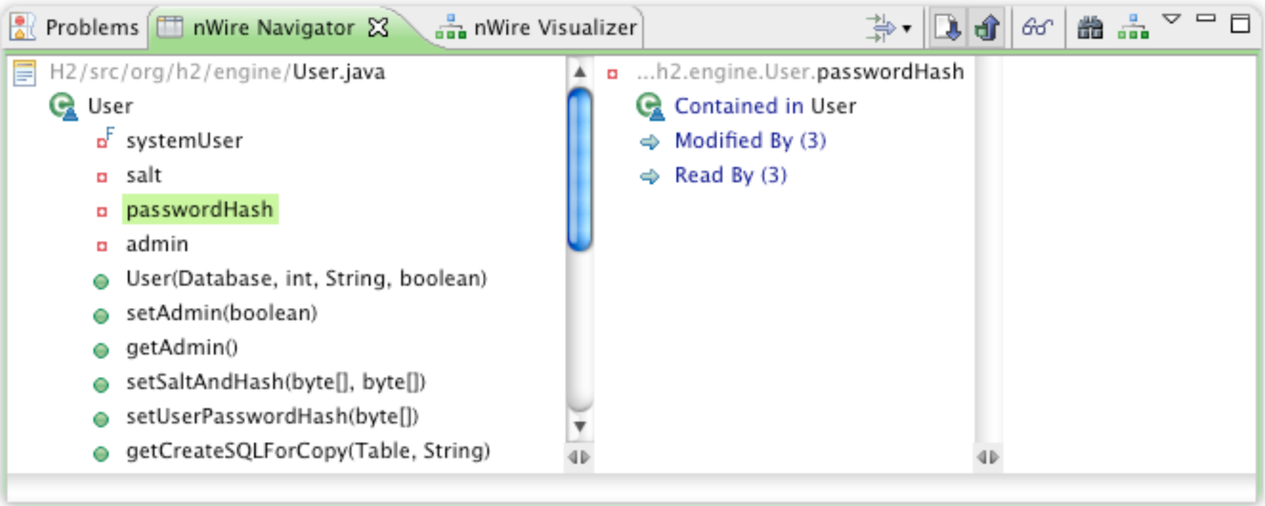
The navigator is the main nWire view. It shows component and associations in a dynamic columns display. Working with the navigator is simple: click on a component to see its' associated components in the next column.

The navigator has two modes: **resource mode** and **search mode**. The resource mode is the default mode when opening the navigator. The search is discussed in the next section.

Navigator Resource Mode

The navigator is composed of tabs. In its' default mode, the first tab shows the components in the current resource and it is very similar to the standard Eclipse Outline view. It is possible to see just the most relevant components when using the Smart Focus mode (see later in this section).

Clicking on a component shows all associated components in the next columns. More columns will be allocated as required.


Problems
nWire Navigator
nWire Visualizer
H2/src/org/h2/engine/User.java
User
systemUser
salt
passwordHash
admin
User(Database, int, String, boolean)
setAdmin(boolean)
getAdmin()
setSaltAndHash(byte[], byte[])
setUserPasswordHash(byte[])
getCreateSQLForCopy(Table, String)
...h2.engine.User.passwordHash
Contained in User
Modified By (3)
Read By (3)

The navigator toolbar has the following buttons:

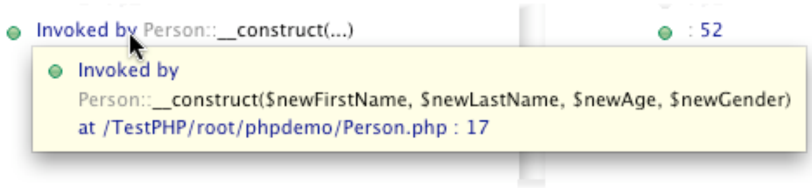
- **Filter** - See filters chapter.
- **Link Editor to Navigator** - toggles the synchronization between the editor and the navigator. When it is on, each movement in the editor will focus the navigator on the most relevant component.
- **Link Navigator to Editor** - toggles the synchronization between the navigator and the editor. When it is on, each selection of a component or association in the navigator will open up the relevant editor and focus on the most relevant position.
- **Smart Focus** - toggles Smart Focus mode (explained later).
- **Search** - turns the navigator to search mode (or returns to resource mode if already in search mode).
- **Open Visualizer** - Opens the visualizer and focuses the visualizer on the current components.

The Navigator Columns

The titles in the columns will wrap according to the column width. Less important parts of the title, e.g. method arguments, may be replaced with ellipsis (...) when the column is too narrow. The columns may be resized using a sash below the vertical scrollbar of each column. Hold and drag the sash to resize the column.

Hovers

Hover above an element in the navigator to show more details on the specific component or associations. Note that the hover delay is determined by the operating system.



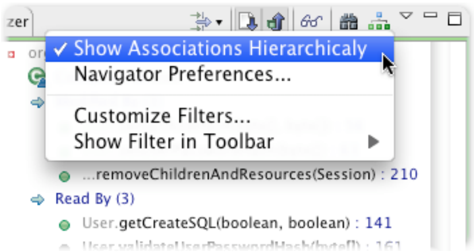
Invoked by Person::__construct(...)

: 52

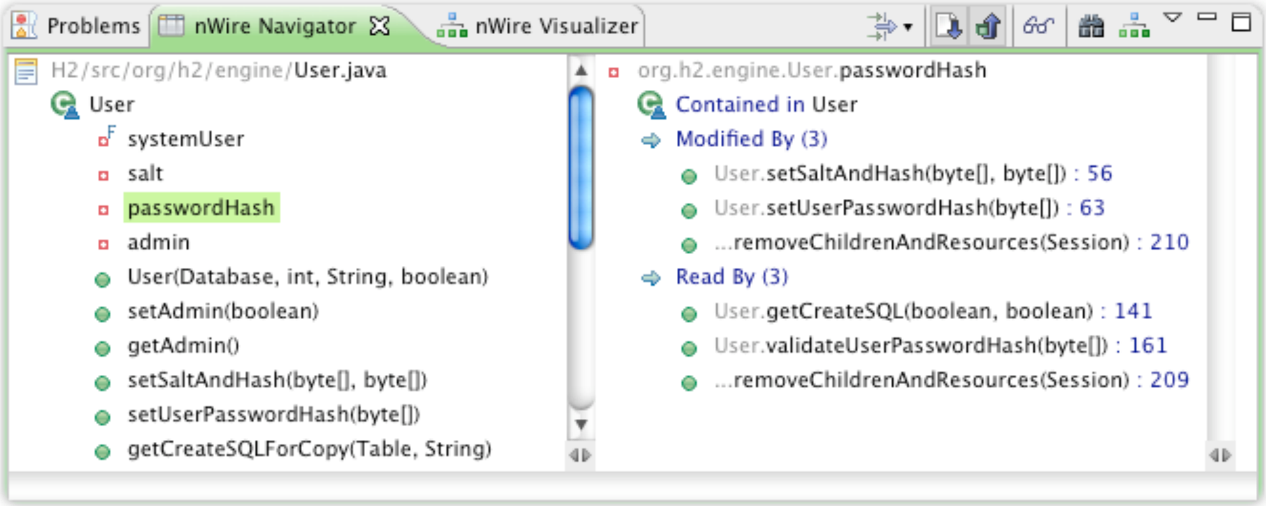
Invoked by
Person::__construct($newFirstName, $newLastName, $newAge, $newGender)
at /TestPHP/root/phpdemo/Person.php : 17

Show Associations Hierarchically

The navigator can show the associations hierarchically under the association type or in the next column. This can be toggled in the drop-down menu.


zer
✓ Show Associations Hierarchicaly
Navigator Preferences...
Customize Filters...
Show Filter in Toolbar
...removeChildrenAndResources(Session) : 210
Read By (3)
User.getCreateSQL(boolean, boolean) : 141

The navigator as shown on the previous page is in the default mode, which does not show the associations hierarchically. Checking this option would result in the following:



H2/src/org/h2/engine/User.java
User
systemUser
salt
passwordHash
admin
User(Database, int, String, boolean)
setAdmin(boolean)
getAdmin()
setSaltAndHash(byte[], byte[])
setUserPasswordHash(byte[])
getCreateSQLForCopy(Table, String)

org.h2.engine.User.passwordHash
Contained in User
Modified By (3)
User.setSaltAndHash(byte[], byte[]) : 56
User.setUserPasswordHash(byte[]) : 63
...removeChildrenAndResources(Session) : 210
Read By (3)
User.getCreateSQL(boolean, boolean) : 141
User.validateUserPasswordHash(byte[]) : 161
...removeChildrenAndResources(Session) : 209

The second column groups are expanded inside the column itself. The expansion of groups in the same column also depends on a predefined threshold, so larger groups will not be expanded in both modes.

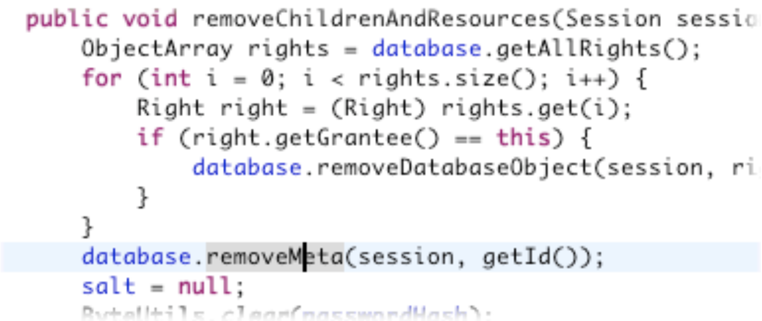
This option creates a more condensed view with less columns, but may contain too much information for some users.

Smart Focus Mode

Starting with version 1.2, the navigator supports a new feature called **Smart Focus**. The purpose of this feature is to quickly provide only the most relevant information on the current element in the editor.

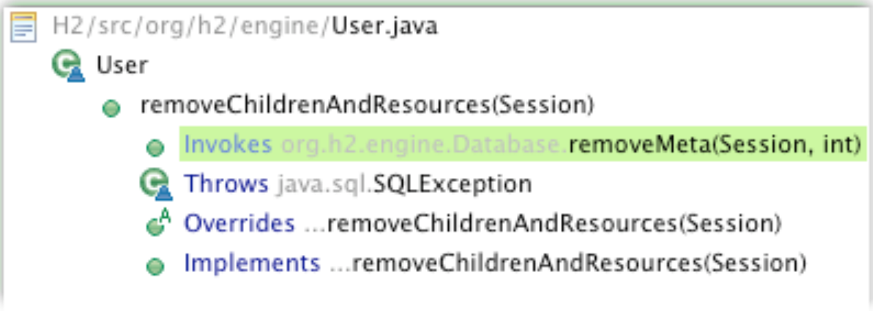
Smart Focus may be toggled at any time by using the glasses icon in the navigator toolbar.

For example, assume you are working on the following method:



```
public void removeChildrenAndResources(Session sessio
    ObjectArray rights = database.getAllRights();
    for (int i = 0; i < rights.size(); i++) {
        Right right = (Right) rights.get(i);
        if (right.getGrantee() == this) {
            database.removeDatabaseObject(session, ri
        }
    }
    database.removeMeta(session, getId());
    salt = null;
    ByteUtils.clear(passwordHash);
```

The navigator in Smart Focus will show the following components:



H2/src/org/h2/engine/User.java
- User
  - removeChildrenAndResources(Session)
    - Invokes org.h2.engine.Database.removeMeta(Session, int)
    - Throws java.sql.SQLException
    - Overrides ...removeChildrenAndResources(Session)
    - Implements ...removeChildrenAndResources(Session)

First, only the relevant components hierarchy are shown: the method, its' containing class and the file. Note that other components declared in the file are not shown in this mode, unlike the default navigator resource mode.

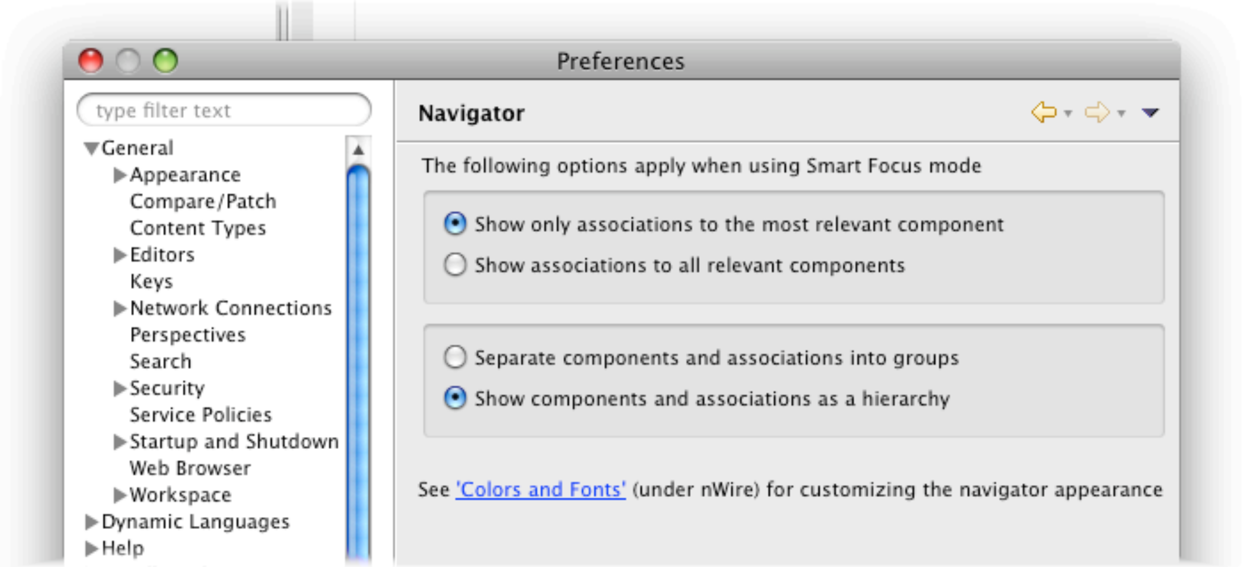
Next, the relevant associations at the given cursor position are shown under the declaring component (the bottom-most component). In this case, the invocation of the method `removeMeta` was the most relevant, so it was also selected. On the second column, you would see all the relevant associations of this method. You can jump to the method declaration with a single click on the method element in the second column (first element).

The Smart Focus mode can essentially replace the need to invoke the **Search References** feature of Eclipse.

*It is recommended to try turning off the **Link Editor to Navigator** option. For some people, it can be distracting. Instead, have nWire focus on the current selection by invoking it using the hot key.*

Customizing the Smart Focus Behavior

There are several optional settings for the Smart Focus, available from the preferences, under **nWire → Navigator**.



Navigator

The following options apply when using Smart Focus mode

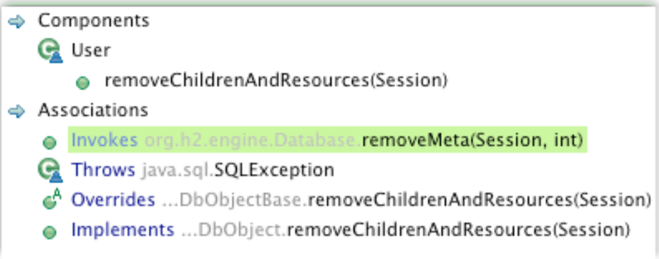
● Show only associations to the most relevant component
○ Show associations to all relevant components

○ Separate components and associations into groups
● Show components and associations as a hierarchy

See 'Colors and Fonts' (under nWire) for customizing the navigator appearance

The first option controls which associations are shown. By default, only the relevant associations for the bottom most declaration will appear. In the previous example, the only the method associations were shown, but not the class associations. Changing this setting will show all the associations relevant at the given cursor position.

The second option controls the separation to groups. This causes the Smart Focus to show the elements in two groups, separating components and associations. The following example shows the navigator in Smart Focus mode, with separation to groups.

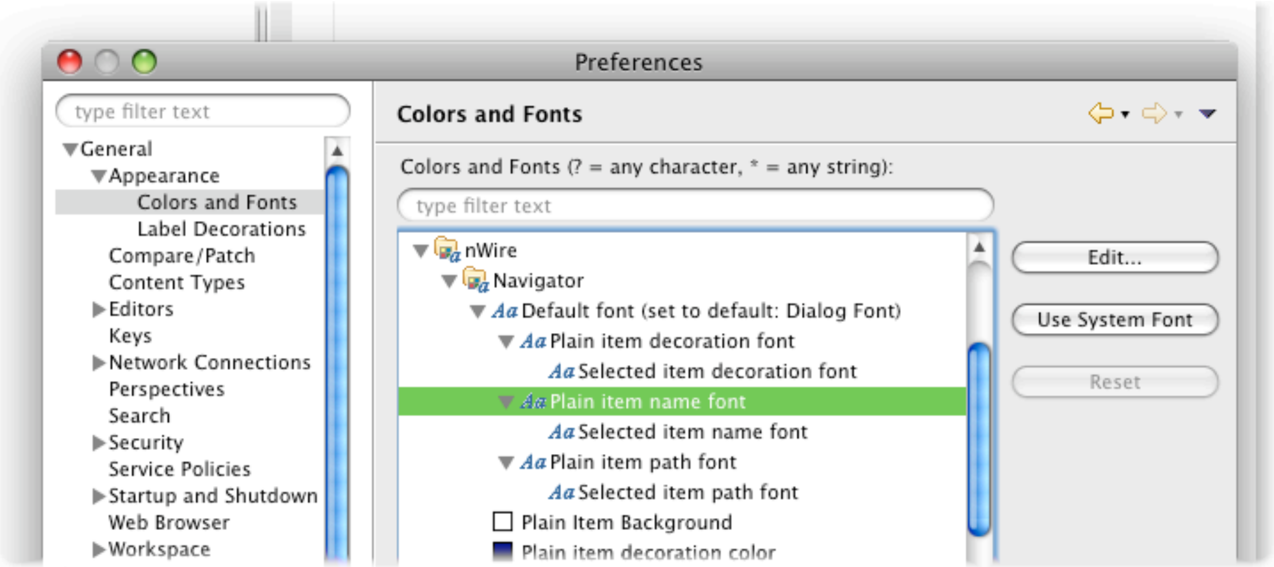


Components
  User
    removeChildrenAndResources(Session)
Associations
  Invokes org.h2.engine.Database.removeMeta(Session, int)
  Throws java.sql.SQLException
  Overrides ...DbObjectBase.removeChildrenAndResources(Session)
  Implements ...DbObject.removeChildrenAndResources(Session)

Customizing the Navigator Appearance

The navigator appearance can be customized in the following steps:

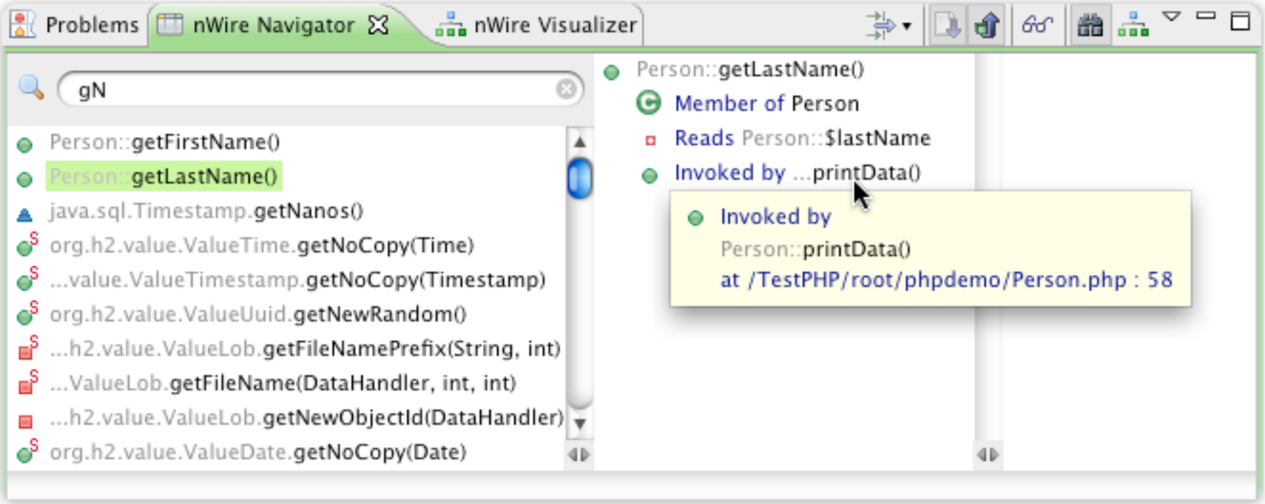
1. Open the Eclipse preferences.
2. Navigate to **General → Appearance → Colors and Fonts**.
3. In the list, expand the node named **nWire**.
4. Select an item to customize and click on **Edit...**



Quick Search

Navigator Search Mode

When in search mode, the navigator shows the search panel at the top of the first column. Start typing to find all relevant components. Once a component is found, click on a component to show its' associations. The rest of the columns behave as usual.



gN

Person::getFirstName()
Person::getLastName()
java.sql.Timestamp.getNanos()
org.h2.value.ValueTime.getNoCopy(Time)
...value.ValueTimestamp.getNoCopy(Timestamp)
org.h2.value.ValueUuid.getNewRandom()
...h2.value.ValueLob.getFileNamePrefix(String, int)
...ValueLob.getFileName(DataHandler, int, int)
...h2.value.ValueLob.getNewObjectId(DataHandler)
org.h2.value.ValueDate.getNoCopy(Date)

Person::getLastName()
Member of Person
Reads Person::$lastName
Invoked by ...printData()

Invoked by
Person::printData()
at /TestPHP/root/phpdemo/Person.php : 58

While in search, the **Link Editor to Navigator** option is disabled. To re-enable it, press the **Search** button in the navigator toolbar to return to **resource mode**.

Search Syntax

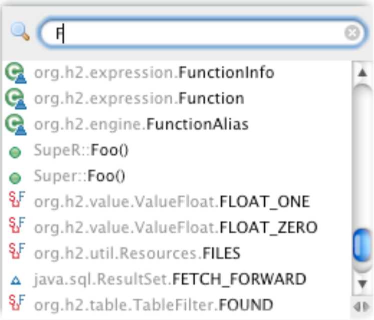
- The search is case insensitive, however, results matching the exact case will appear first.
- Use asterisk (`*`) as a wild-card character denoting any number of missing characters.
- Use a question mark (`?`) as a single wild-card character, denoting exactly one missing character.
- The search assumes a wild-card before any capital letter. This is very useful for camel case names. For example, when searching for `createTable`, type `crTa` which will be the same as writing `cr*Ta`, thus finding the desired component.
- A wild-card is assumed at the end of the search phrase.
- Wild-card characters are allowed at the beginning as well.

The camel casing feature enables searching using initials. For example, searching for the text ***`NPE`*** *will return* ***`N`***`ull`***`P`***`ointer`***`E`***`xception`.

Search Results

The maximum number of search results can be controlled from the preferences pane (under **nWire → Search**).

The results will be sorted by relevancy and by type. More important types appear earlier. For example, a Class will appear before a Method, which will appear before a Field.



F

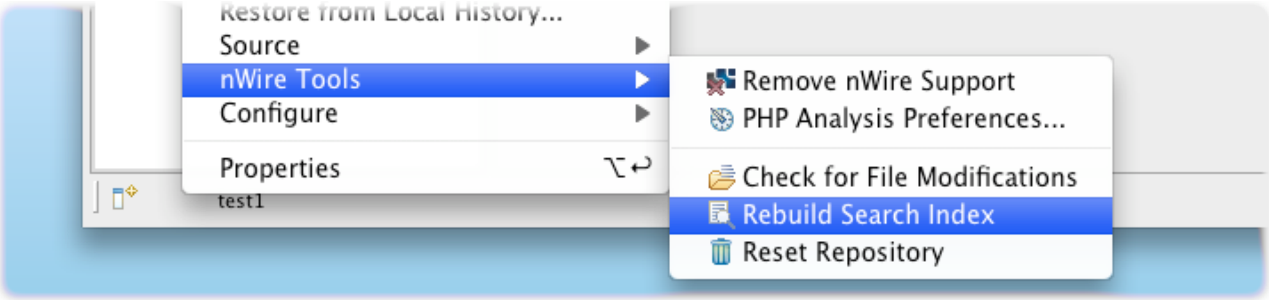
org.h2.expression.FunctionInfo
org.h2.expression.Function
org.h2.engine.FunctionAlias
SupeR::Foo()
Super::Foo()
org.h2.value.ValueFloat.FLOAT_ONE
org.h2.value.ValueFloat.FLOAT_ZERO
org.h2.util.Resources.FILES
java.sql.ResultSet.FETCH_FORWARD
org.h2.table.TableFilter.FOUND

Search Index

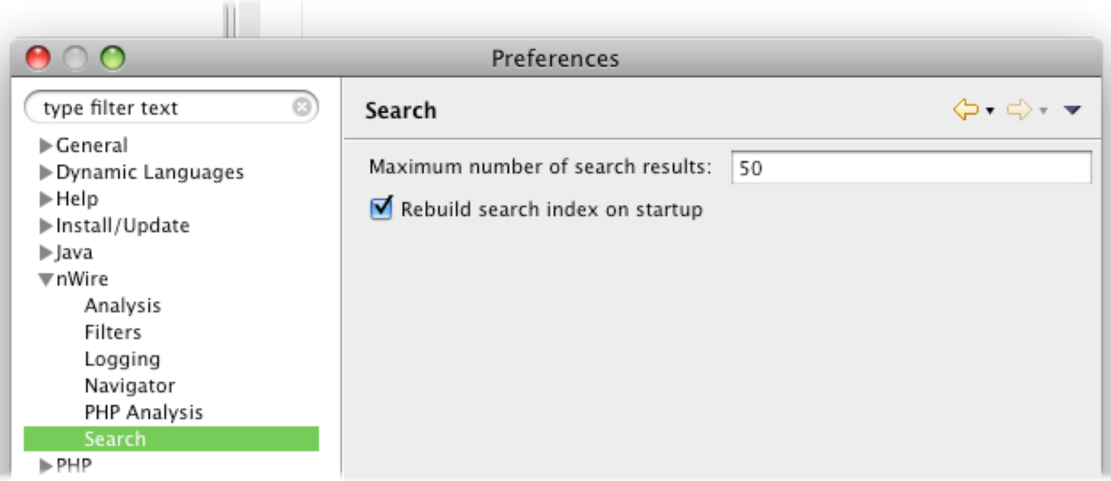
The search performance is achieved by using the vastly popular Apache Lucene search engine. The index is updated with the update of the repository during the analysis process. In some cases, the index may be outdated, and may miss some results. It can be quickly rebuilt with the following steps:

1. Right click on the project in the **Package Explorer** or **PHP Explorer**.
2. Select **nWire Tools → Rebuild Search Index**.

The process takes less then a minute in most workspaces.



The index is also rebuilt every time Eclipse starts. This is done as a precaution, making sure you get all the possible search results. This behavior may be changed in the preferences, under **nWire → Search**.

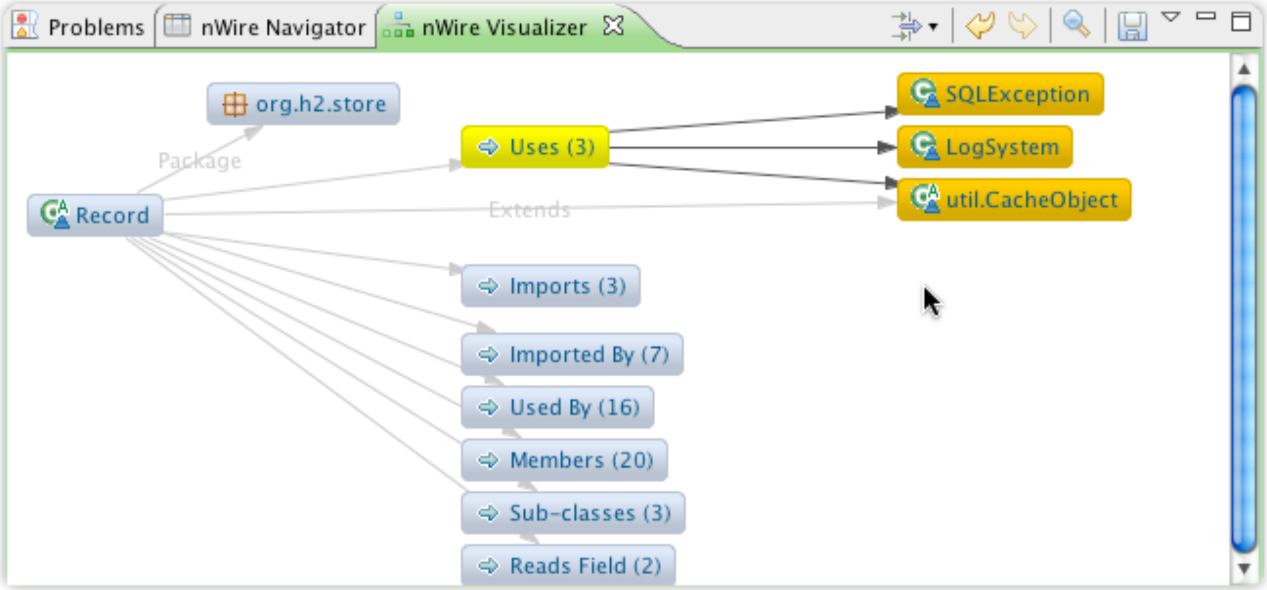


Maximum number of search results: 50

☑ Rebuild search index on startup

The Visualizer

The visualizer presents associations and components in a graphical representation. Double clicking a component will expand it to reveal all its' associations. A single click will select the component, highlighting the directly associated components.



The visualizer toolbar has the following buttons:

- **Filter** - See filters chapter.
- **Undo** / **Redo** - Undo or redo an expand node action. Focus actions cannot be undone.
- **Focus on Selection** - will clear the visualizer canvas and show only the currently selected components and associations.
- **Save Image** - saves the visualizer image to a PNG file.
- **Zoom menu** - a drop-down menu for setting the zoom level. When in **page** mode, the zoom level will be adjusted to accommodate all the components in the current view.

Filters

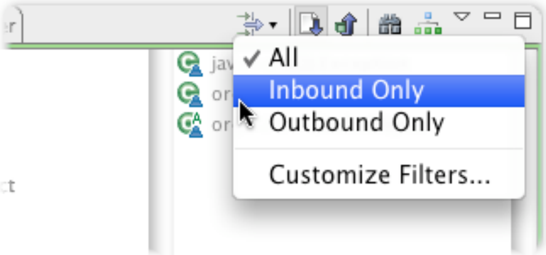
The filters feature allows filtering the associations shown in the navigator and visualizer. The common scenario:

1. Define a filter using the **Filters Preferences Page**.
2. Set the selected filter using the **Filter Selector**.
3. Browse the components and associations in the navigator and visualizer. The selected filter can be switched at any given time and the same selected filter applies to the navigator and visualizer.

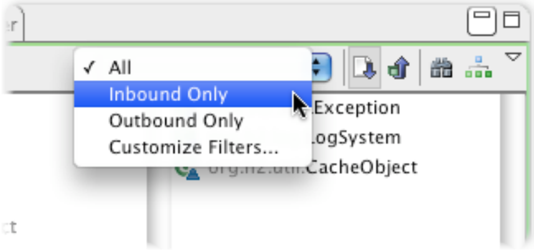
Filter Selector

The filter selector appears in the toolbar of the navigator and visualizer. It can appear as an icon with a drop-down menu or as a combo-box.

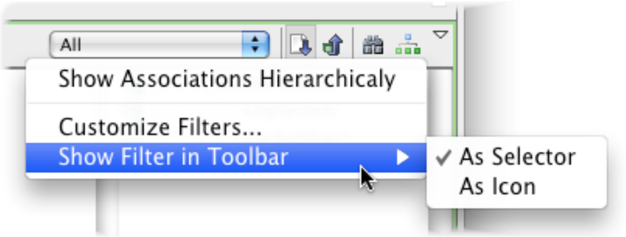
Selector Shown as Icon



Selector Show as Combo-Box

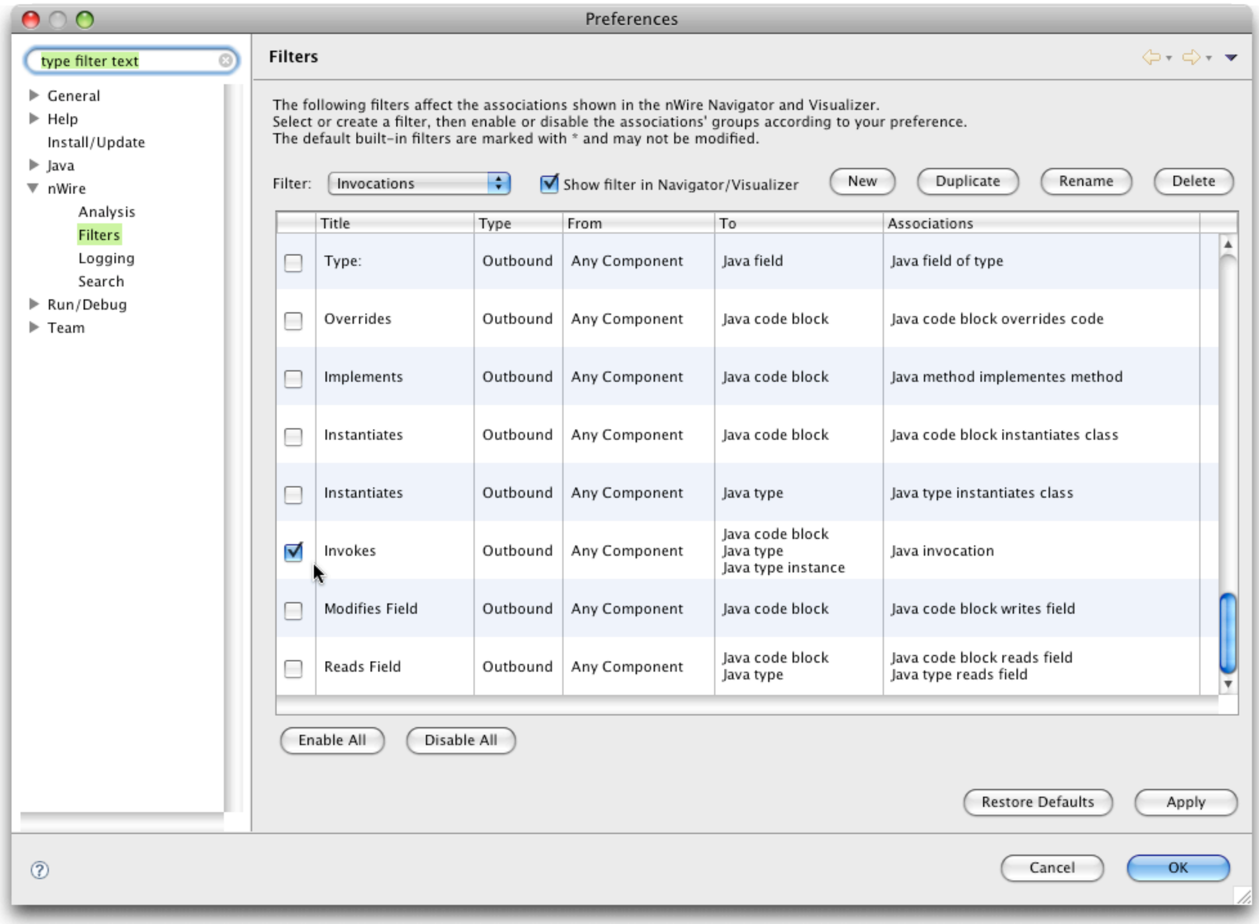


Switching between Icon and Selector can be done using the toolbar drop-down menu



Selecting **Customize Filters...** will open the filters preferences page in the preferences dialog box.

Filter Definition



The following filters affect the associations shown in the nWire Navigator and Visualizer.
Select or create a filter, then enable or disable the associations' groups according to your preference.
The default built-in filters are marked with * and may not be modified.

Filter: Invocations

☑ Show filter in Navigator/Visualizer

	Title	Type	From	To	Associations
☐	Type:	Outbound	Any Component	Java field	Java field of type
☐	Overrides	Outbound	Any Component	Java code block	Java code block overrides code
☐	Implements	Outbound	Any Component	Java code block	Java method implementes method
☐	Instantiates	Outbound	Any Component	Java code block	Java code block instantiates class
☐	Instantiates	Outbound	Any Component	Java type	Java type instantiates class
☑	Invokes	Outbound	Any Component	Java code block Java type Java type instance	Java invocation
☐	Modifies Field	Outbound	Any Component	Java code block	Java code block writes field
☐	Reads Field	Outbound	Any Component	Java code block Java type	Java code block reads field Java type reads field

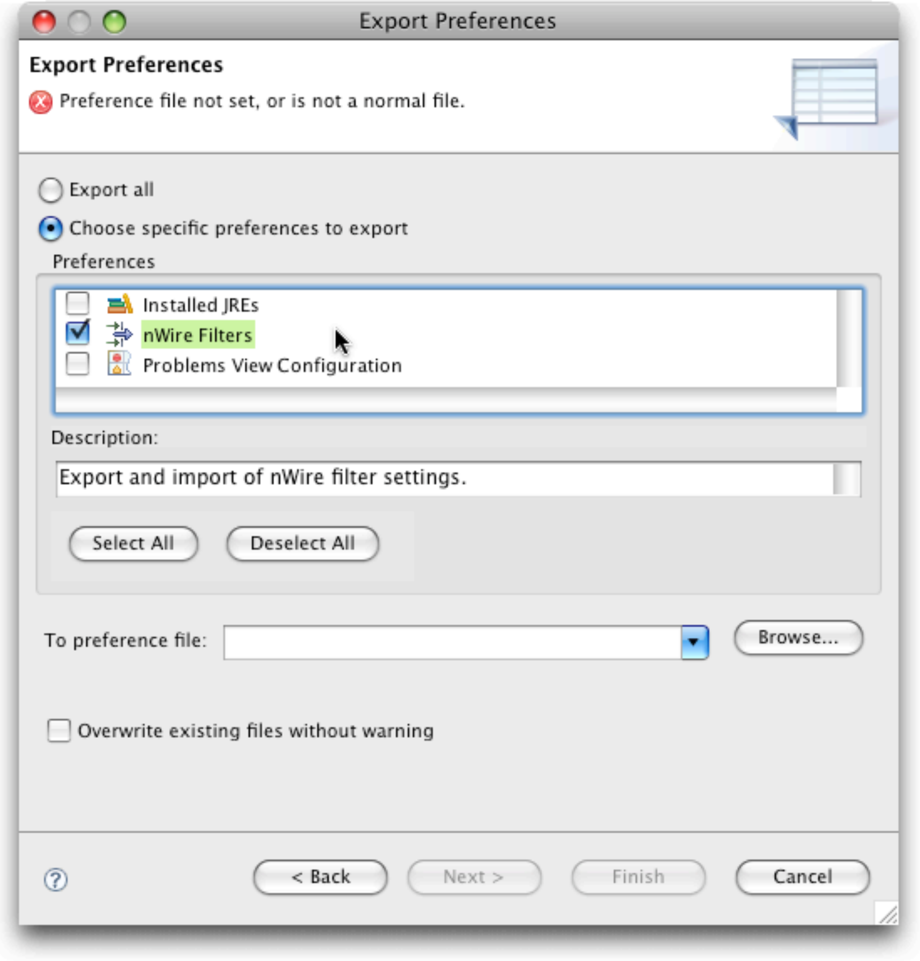
Follow these steps to define a new filter:

1. Open the **Filters preferences page**.
2. Click on **New** to create a new filter or select an existing filter and click on **Duplicate** to copy and modify it.
3. Select the associations that will be shown when the filter is selected.
4. Click **OK**.

There are several predefined filters which cannot be modified. You may use them as a starting point to create new filters.

Filter Exporting/Importing

The Filter preferences can be individually exported and imported to/from other workspaces using the Eclipse standard preferences transfer. Select **File → Export...**, then select **General → Preferences**. Now, select **nWire Filters** as the specific preferences to export.



Export Preferences

Export Preferences

Preference file not set, or is not a normal file.

○ Export all

● Choose specific preferences to export

Preferences

☐ Installed JREs

☑ nWire Filters

☐ Problems View Configuration

Description:

Export and import of nWire filter settings.

To preference file:

☐ Overwrite existing files without warning

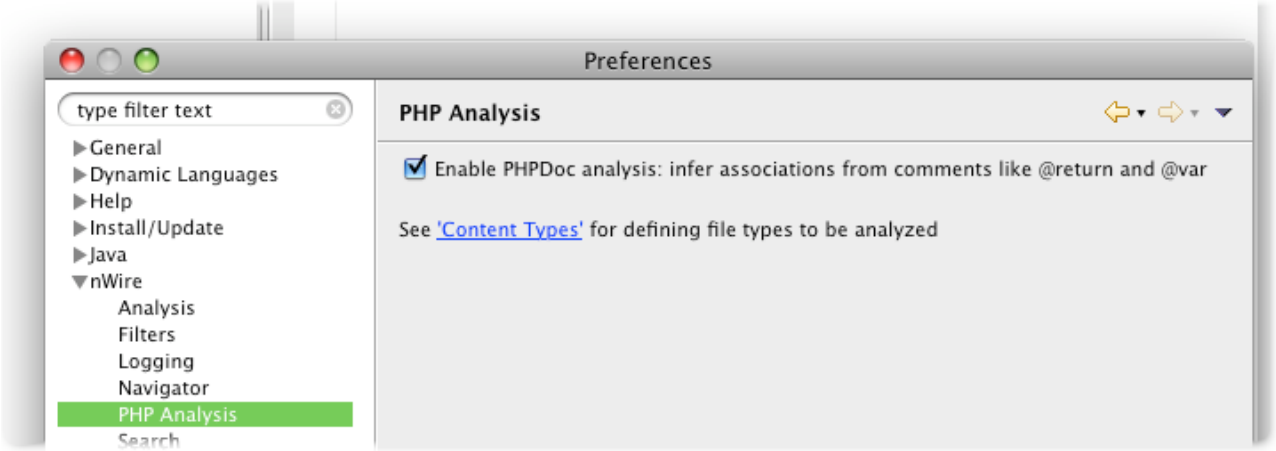
Import is done in a similar manner.

PHP Analysis

Code analysis in PHP is not as straightforward as it is in Java. PHP is a dynamic and weakly typed language, which makes it difficult to infer the right associations. The following section will explain how nWire deals with PHP and how you can get the correct information from nWire.

PHP-Doc Analysis

nWire uses PHP-Doc comments to infer associations which are unknown due to the missing type system. For example, method return type or field type. If you are not using PHP-Doc or not interested in these associations, you can turn off this option in the preferences, under **nWire → PHP Analysis**. This will slightly improve the analysis performance.



☑ Enable PHPDoc analysis: infer associations from comments like @return and @var

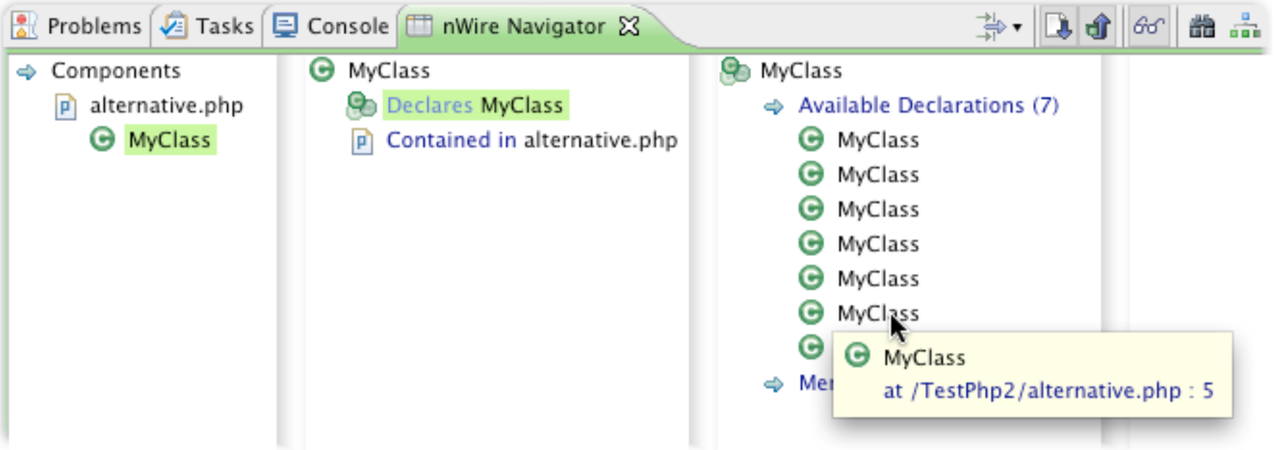
See 'Content Types' for defining file types to be analyzed

Multiple Declarations

In PHP, most components may have more than one declaration, for example, you may have two alternative declarations for a class or simply a class with the same name in two different files.

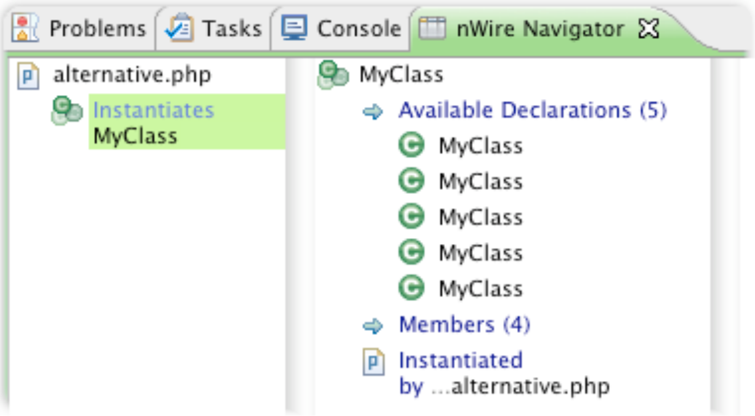
When a given component has multiple declarations, nWire will always try to determine which of those declarations is the most relevant and will show the other alternative declarations.

In the following example the class `MyClass` has several possible declarations:



When selecting the current declaration in the first column, the element **Declares MyClass** represents a **virtual class** (denoted by a special group icon) which has several **alternative declarations**. Once the virtual class is selected in the second column, all available declarations are presented in the next column.

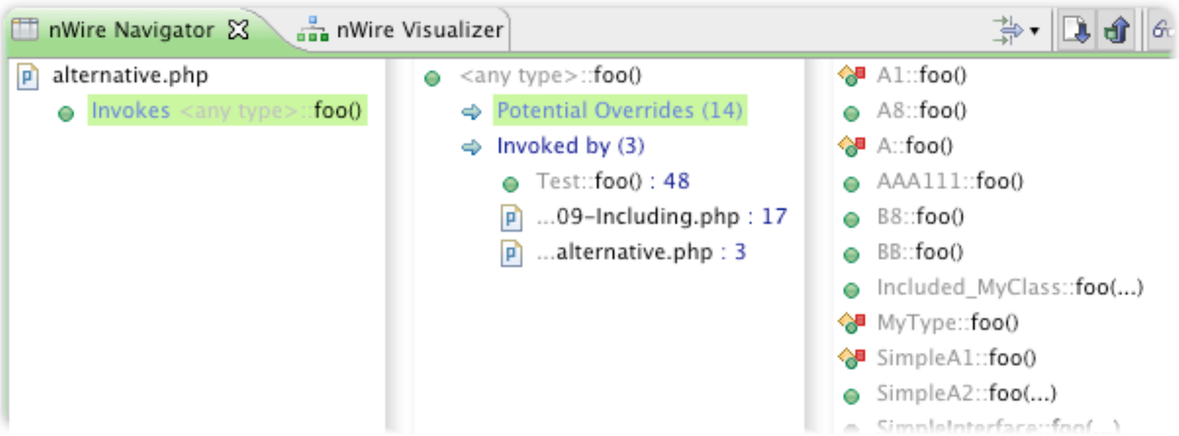
nWire will always try to associate the code to the correct declaration, according to the `include` and `require` statements in your code. In case nWire cannot determine the exact declaration, the virtual class will be used instead, as shown in the following example:



Members of <any type>

When your code refers to a type member (method, field or constant), nWire will try to resolve the reference and find the relevant type (class or interface). In some cases, this resolution fails. A common example for that would be using an object returned from a method as a target (e.g. `$a->foo()->bar()`).

In these cases, nWire will assume the member belongs to a general type called **<any type>**. Here's an example:



The current file invoke a method `foo()` on an unknown type. The member is thus considered to be a member of <any type>. Clicking on **Potential Overrides** in the second column reveals all the members of the given name and enables exploring their association.

nWire User Interface Reference

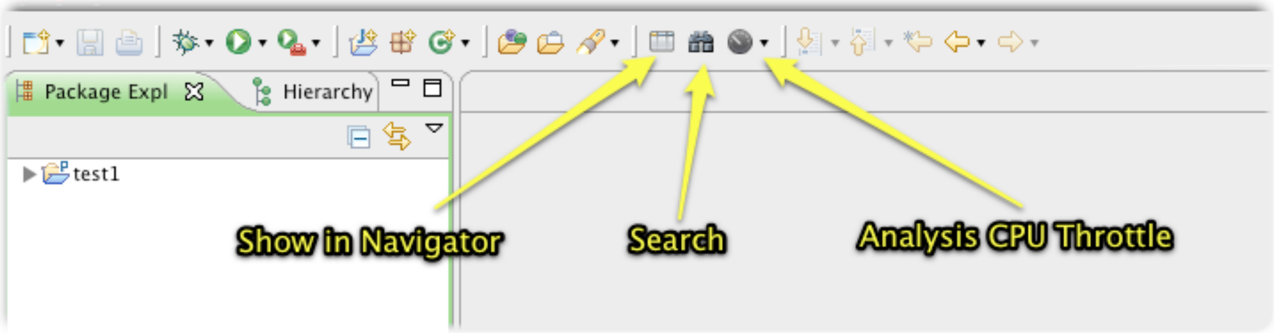
This section summarizes the UI elements of nWire.

Views

- **The Navigator** - a vertical column view which enables browsing components and associations.
- **The Visualizer** - presents the components and associations in a graphical representation.

Toolbar Buttons

- **Show in Navigator** (Ctrl + 4): shows the navigator (if not shown) and focuses the navigator of the current element in the editor.
- **Search in Navigator** (Ctrl + 5): shows the navigator (if not shown) and turns the navigator to search mode, allowing you to instantly search for components.
- **nWire Analysis CPU Throttle**: controls the CPU utilization during the analysis process.



On Mac OS X, the keyboard shortcuts are mapped to Command (⌘) key.

Using the keyboard shortcuts is the key in making nWire a true productivity booster. Master the shortcuts or customize them to your preference using the standard Eclipse keys preference page.

Menus

- The **nWire** menu under **Help → nWire**, has the following options:
  - **User Info** - shows information about nWire license and enables activating nWire with a new license key.
  - **Feedback** - Opens a dialog box for sending product feedback or asking questions. This dialog box uses an online form from **GetSatisfaction.com** and it requires an internet connection. Alternatively, you may send an email to support@nwiresoftware.com.
  - **Getting Started** - Opens the **Welcome to nWire** dialog box.

    - **Release Notes** - Shows the release notes for the latest version of nWire. This dialog box will open automatically after nWire is updated.
    - **Tip Of The Day** - Shows the tip of the day dialog box. This box is shown when Eclipse starts by default.
- The **nWire Tools** menu which appears when right-clicking a project in the Package Explorer, has the following options:
    - **Add nWire Support** or **Remove nWire Support**.
    - **PHP Analysis Preferences** *(PHP Only)* - opens the PHP Analysis preferences pane
    - **Check for File Modifications** - analyzes the files for modifications.
    - **Rebuild Search Index** - rebuilds the quick search index.
    - **Reset Repository** - erase the entire nWire repository. This will cause a complete reanalysis of all the nWire-enabled projects. Use this option if you encounter inconsistencies in the components or associations.

Preferences Panes

nWire introduces the following preferences panes:

- **Main Page** - enables showing “Tip Of The Day” on startup.
- **Analysis** - enables or disables the auto-analysis option.
- **PHP Analysis** *(PHP Only)* - controls specific PHP Analysis properties (see PHP section).
- **Navigator** - controls the navigator behavior (see navigator section).
- **Search** - controls the number of search results shown to the user.
- **Filters** - defines filters (see filters section).
- **Logging** - controls the debug logging. This is for support purposes and should not be used unless specifically instructed by our support team.

Other relevant preferences panes:

- **General → Keys** - for changing the keyboard shortcut mappings.
- **General → Appearance →Color and Fonts** - for changing the appearance of the navigator elements.
- **General → Content Types** - for setting the file types to be analyzed.