NBPCG User Guide

Table of Contents

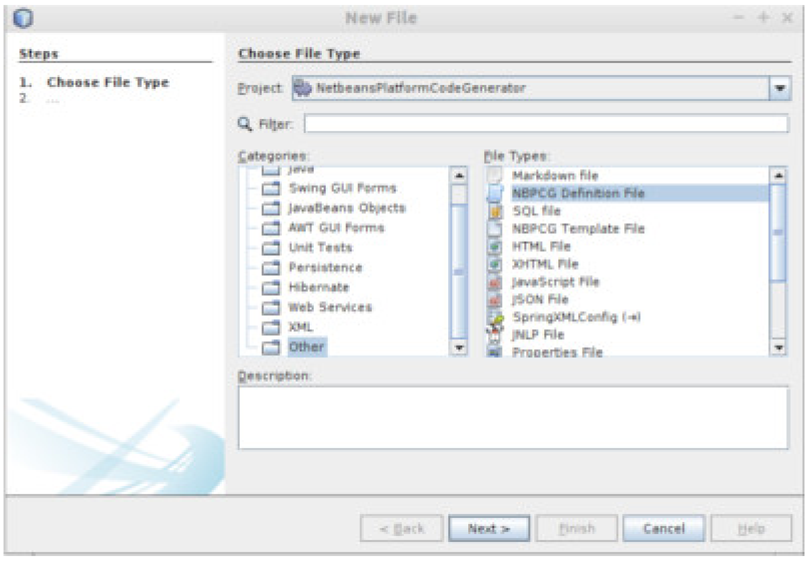
Introduction

The Netbeans Platform Application Code generator is a Netbeans plug-in which creates a set of artifacts to speed up creation of a NetBeans Platform Application. These artifacts can include:

- Database creation scripts
- Other utility scripts
- Entity classes and their associated Entity Managers
- Node class for all entities
- Explorer Views for all entities
- Editors for all entities

The NBPCG script

The generation of these artifacts is controlled by a single NBPCG script which describes the necessary build details, entities details and also the node relationships. The NBPCG script is an XML document. A template is available - using the New... entry, the template can be selected from the other category with file type NBPCG Definition file



Project: NetbeansPlatformCodeGenerator

For full details of the content and syntax of the script file can be found here.

Feedback/Support

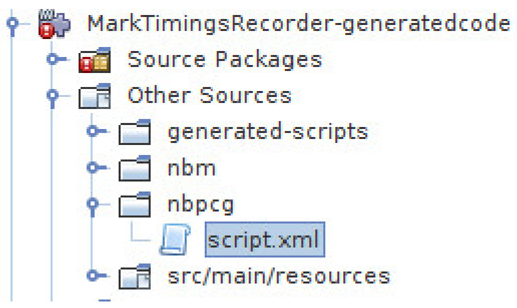
The author would be interested in any feedback on this project from users of the plug-in. Issues (bugs or possible enhancements) can be logged using the project issue tracker

Creating a project and running the code generator

A normal NetBeans module project should be created (both Maven and ANT build system projects are supported). This project is then customised so that the code generator can be add code to the project.

Customising a Maven project

- In Projects View, Add a folder named `nbpcg` under `Other Sources`.
- Place the nbpcg script file (`script.xml`) in the `nbpcg` node



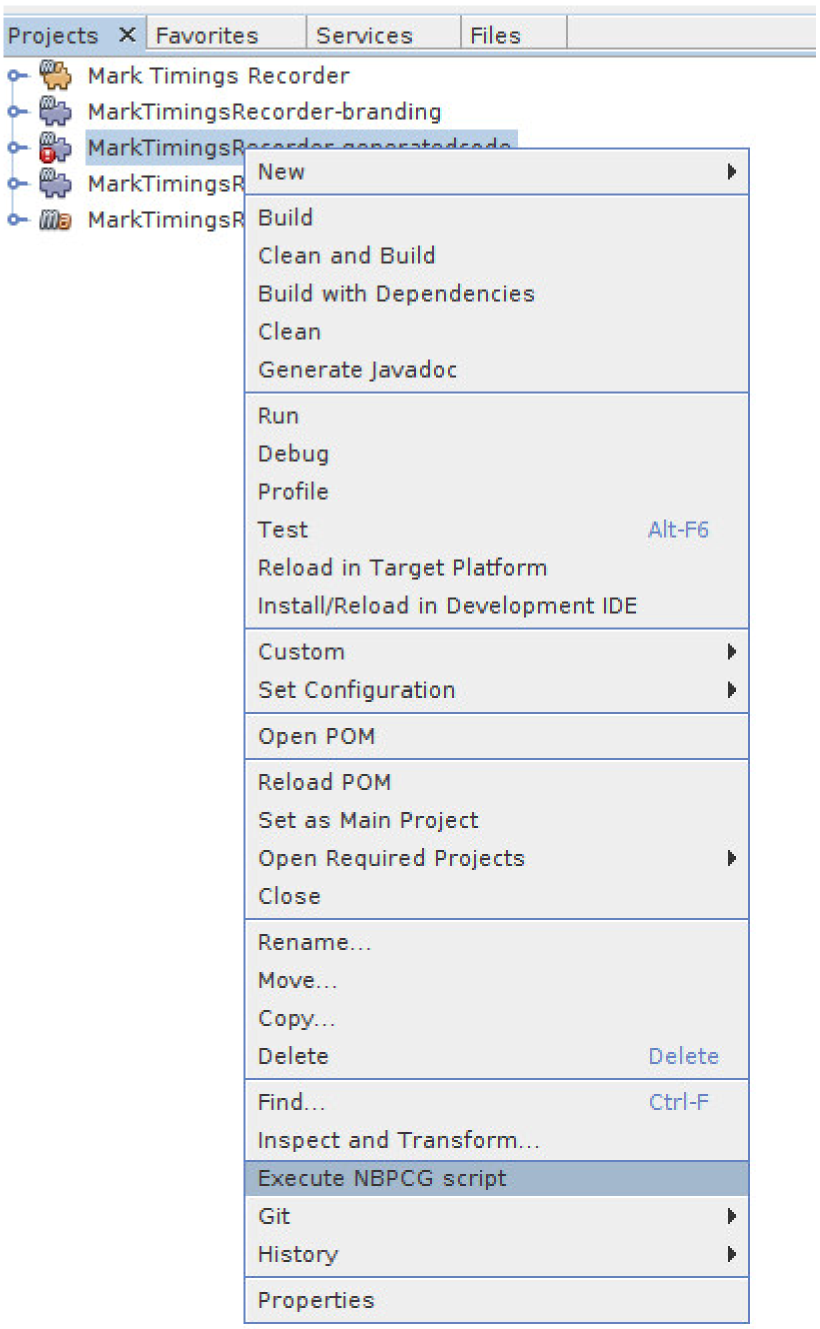
Customising an ANT project

- In Files view, Add a folder name `nbpcg-files` under the project root.
- In Project view, Place the nbpcg script file (`script.xml`) in the `NBPCG File` node.

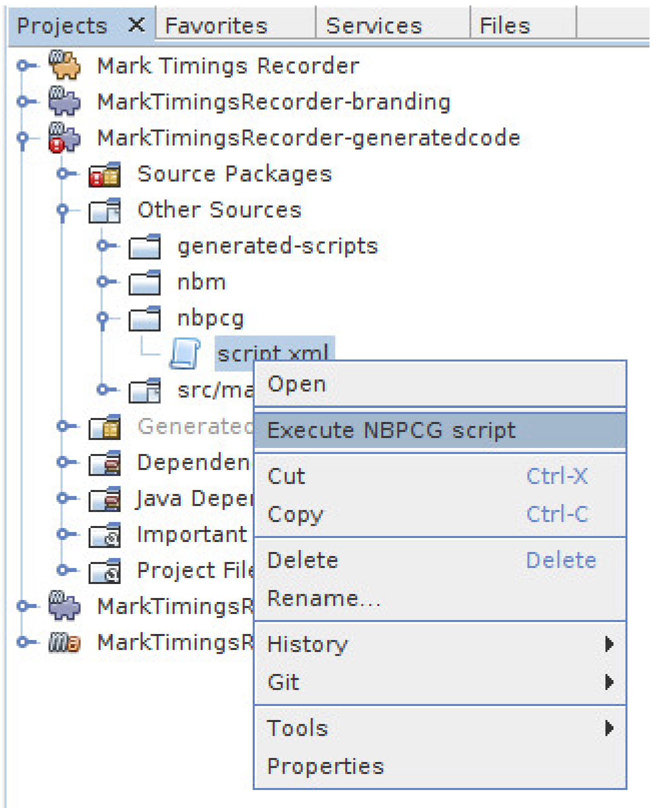
Executing the NBPCG script

The NBPCG script can be executed by actions on the popup menu on:

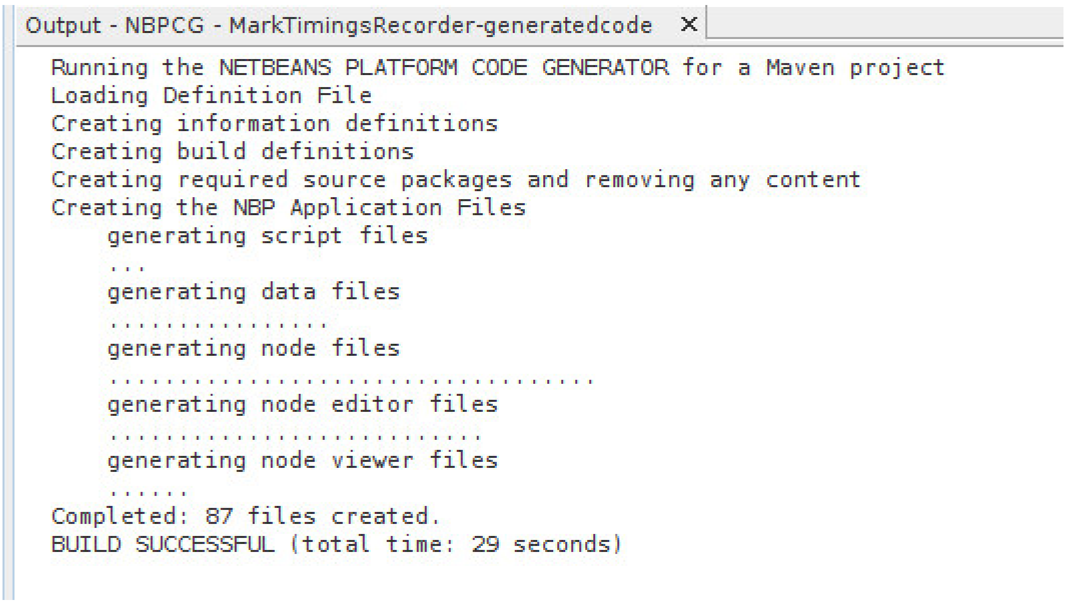
1) The Project node



2) The NBPCG script file



Executing the script will open a new tab in the output window. Progress and errors will be reported in that tab.



```
Running the NETBEANS PLATFORM CODE GENERATOR for a Maven project
Loading Definition File
Creating information definitions
Creating build definitions
Creating required source packages and removing any content
Creating the NBP Application Files
    generating script files
    ...
    generating data files
    ................
    generating node files
    ...................................
    generating node editor files
    ...........................
    generating node viewer files
    ......
Completed: 87 files created.
BUILD SUCCESSFUL (total time: 29 seconds)
```

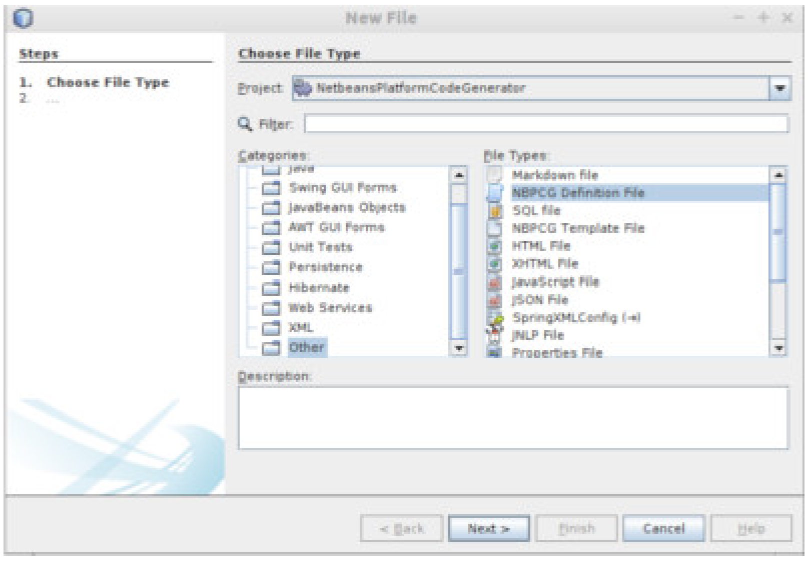
Dependency Information for NBPCG generated Modules

NBPCG does not currently setup dependencies for the generated modules (either on NBPCG support libraries or Netbean Platform modules). The following documentation details the dependencies that exist and should be added to the various generated modules.

	Datamodel	**NodeModel**	**NodeViewer**	**NodeEditor**
NetBeans Platform				
Explorer & Property Sheet API		x	x	
Lookup API	x		x	x
Nodes API		x	x	x
UI Utilities API		x	x	
Utilities API		x	x	x
Window System API		x	x	
NBPCG-Library				
MySQL	*			
NBPCG Data Support Lib	x	x	x	x
NBPCG Form Support Lib			x	
NBPCG Node Support Lib		x	x	x
NBPCG Support Lib	x	x	x	x
NBPCG TopComponent Support Lib		x	x	
GENERATED MODULES				
XX DataModel		x	x	x
XX NodeModel		x	x	
XX NodeEditor		x		
If you are using table aliases to access other tables then you will need to add addition dependencies to those modules				
ALIAS MODULES				
XX DataModel	x	x	x	x

Script Content/Syntax Notes

The NBPCG script is an XML document. A template is available - using the New... entry, the template can be selected from the other category with file type NBPCG Definition file



Project: NetbeansPlatformCodeGenerator

The structure of the script XML file is described below:

The top level element is the <nbpcg> element.

The <nbpcg> element

Attributes

- name: The name of the script (required)

Child Elements

- build: (one or more)
- node: (one or more)
- databases: (one or more)

The <build> element

Defines the build information - artifact selection, projects and packages etc.

Attributes

- viewerrole: the viewerroles for this application - comma separated list (required)
- copyright: the copyright key information - year(s) and name / email, this will be added to every licence header placed in a generated file (required)

Child Elements

- project: (one or more)

The <project> element

Defines the project information.

Attributes

- name: the project name (required)
- license: the license header to be added to generated files - one of apache20, gpl30 or lgpl21 (optional, default apache20)

Child Elements

- generate: (one or more)

The <generate> element

Requests the generation of code or resource files for various types of entities.

Attributes

- type: the type of code artifact to be generated - one of data, dataaccess, node, nodeviewer, editor, script (required)
- package: the package name to be used for all artifacts generated by this request (required)
- exclude: a list of nodes or entities to be excluded from this generate command, the list is comma separated and must not include any spaces around the comma (optional)

The <node> element

Defines the nodes required and their interrelationships.

Attributes

- name: the node name (required)
- view: the types of view required - one of both, icon, tree, none (optional, default tree)
- viewers: (optional)
- dbname: the database name (optional)
- rootlabel: (optional)
- rooticon: the root node icon name (optional)
- icon: the node icon name (optional)
- label: (optional)
- orderable: set to yes if nodes are allowed to be orderable (by drag and drop) - default is no (optional)
- nullallowed: set to yes if null value is allowed (optional)
- fkey: set to no if foreign key is not utilised - default is yes (optional)

- nomodifiers: set to yes if nodeviewers are to be created without the default copy/cut/paste/delete functionality (optional)
- customchildfactorypackage: set if user provides the child factory implementation for this node. The value is the package in which the custom code exists (optional)
- childnodesineditor: a comma separated list of child nodes which are to be displayed in this node's editor as editable tables (optional)
- dynamicicon: the method use to obtain the icon for this node, if dynamic icon selection is required (optional)
- displaynameformat: format for the node's display name (optional)
- displaytitleformat: format for the node's display title (optional)
- sortformat: format for the node's sort string - also implies sortable (optional)
- childnodesineditor: the list of child nodes in this editor (optional)
- min: minimum number of this node type (as a childnode) (optional)
- max: maximum number of this node type (as a childnode) (optional)

Child Elements

- entryfields: entry fields (zero or more)
- action: (zero or more)
- node: child nodes of this node (zero or more)

The <entryfield> element

Defines an entry fields and its configuration and mapping to an entity field.

Attributes

- name: the name of the field (required)
- label: (optional)
- type: the entry field type (must be password) (required)
- mapsto: the entity field to which this field is mapped (required)
- mapping: the method used to converted the entry field value to the entity field value (required)
- rule: the method to be applied to this entry field to test it correctness (optional)
- errormessage: The error message to be generated if the rule fails (optional)

The <action> element

Attributes

- name (required)
- label (required)

The <databases> element

Collection of all database definitions.

Child Elements

- database: (zero or more)

The <database> element

Definition of a database object.

Attributes

- name: the logical name of this database (required)
- dbname: the database name of this database - default is attribute name (optional)
- pkey: database key type - one of idauto (optional)
- extrafields: additional standard fields to add - one of usertimestamp (optional)
- usepackage: ?? (optional)

Child Elements

- table: (zero or more)

The <table> element

Definition of a database table object.

Attributes

- name: entity name (required)
- dbname: the database name of this table - default is attribute name (optional)
- pkey: database key type - one of idauto (optional)
- extrafields: additional standard fields to add - one of usertimestamp (optional)
- rule: the table level rule to be applied (optional)

Child Elements

- field: (zero or more)
- insertentity: (zero or more)

The <field> element

Definition of a database table field (column) object.

Attributes

- pkey: yes if this field is the primary key (optional)
- name: field name (required)
- dbname: the table column name - defaults to attribute name (optional)
- index: define the indexing for this field - one of unique or yes - default is no (optional)
- unique: define the uniqueness of this field - one of yes - default is no (optional)

- type: data type of the field one of boolean, long, int, date, datetime, enum, password or reference - default is string (optional)
- decimalsize: the size of the decimal number (optional)
- choicemethod: the method to be used to obtain set of possible choices for this field (optional)
- nullallowed: are null values allowed? - one of yes - default is no (optional)
- label: the field label (optional)
- references: the target entity for this reference (optional)
- fkey: set to no if foreign key is not utilised - default is yes (optional)
- values: set of enum values (comma separated) (optional)
- min: min length of string entered, min value of number entered (optional)
- max: max length of string entered, max value of number entered (optional)
- future: set to yes if date or datetime entered must be in future (optional)
- past: set to yes if date or datetime entered must be in past (optional)
- tablecolumnname (optional)

The <insertentity> element

Definition of an data row to be inserted into the database table object.

Child Elements

- insertfield: (zero or more)

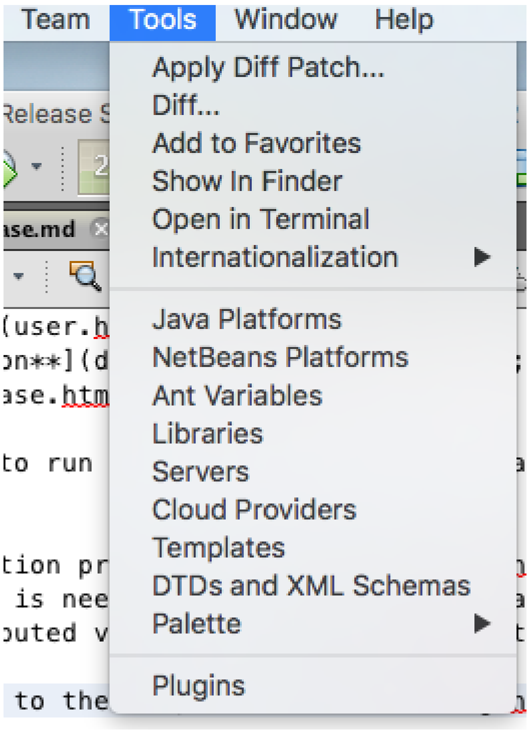
Installation Notes

Current release: v5.0.1

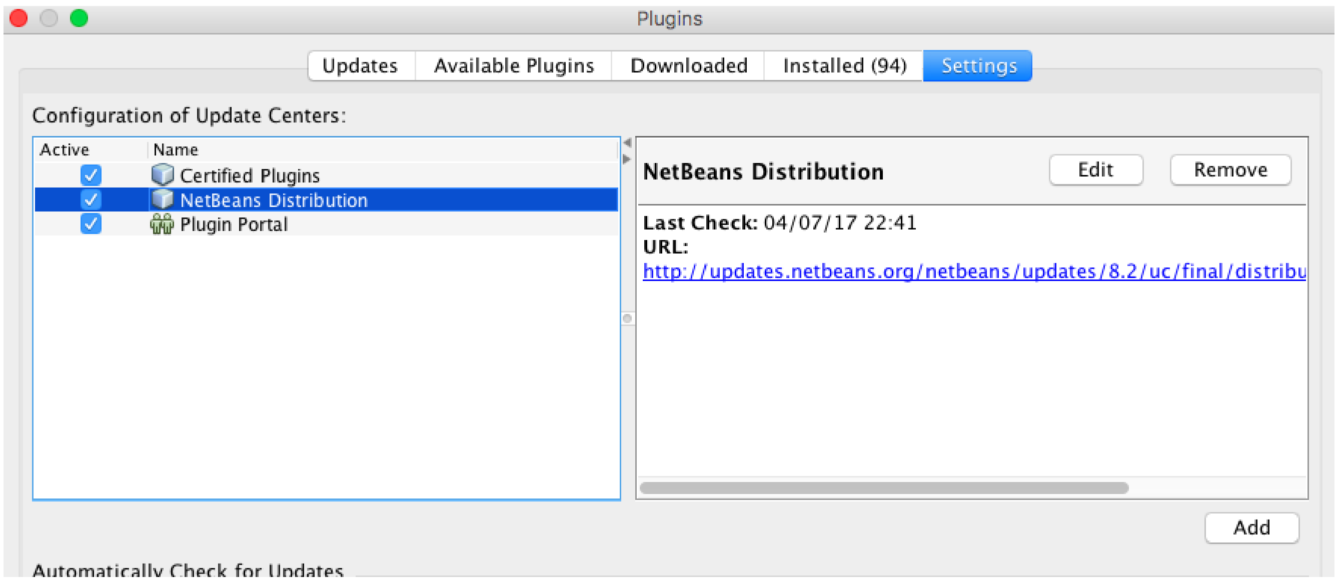
Minimum Environment: Netbeans 8.2 and Java 8

The recommended installation process is obtain the Plugin using an additional Update Centre. This must be defined in your IDE, but is needed to be done only once as all IDE plugins developed by the The-Retired-Programmer programme will be distributed via this single Uodate Centre.

To add the Update Centre to the IDE, click on the Plugins entry in the Tool Menu to display the Plugins Dialog.



Select the Setting tab, to display the current list of Update Centres.



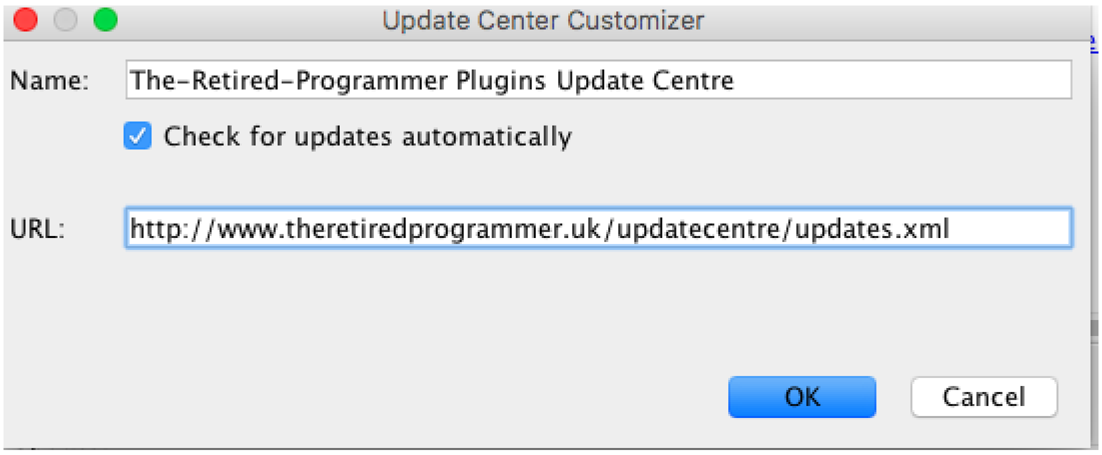
Configuration of Update Centers:

Active	Name
✓	Certified Plugins
✓	NetBeans Distribution
✓	Plugin Portal

NetBeans Distribution

Last Check: 04/07/17 22:41
URL:
http://updates.netbeans.org/netbeans/updates/8.2/uc/final/distribu

Assuming that the additional Update Centre has not already been defined, press the Add button and enter the two required fields (Name and URL). Once correctly entered, press OK to complete this operation.



Update Center Customizer

Name: The-Retired-Programmer Plugins Update Centre

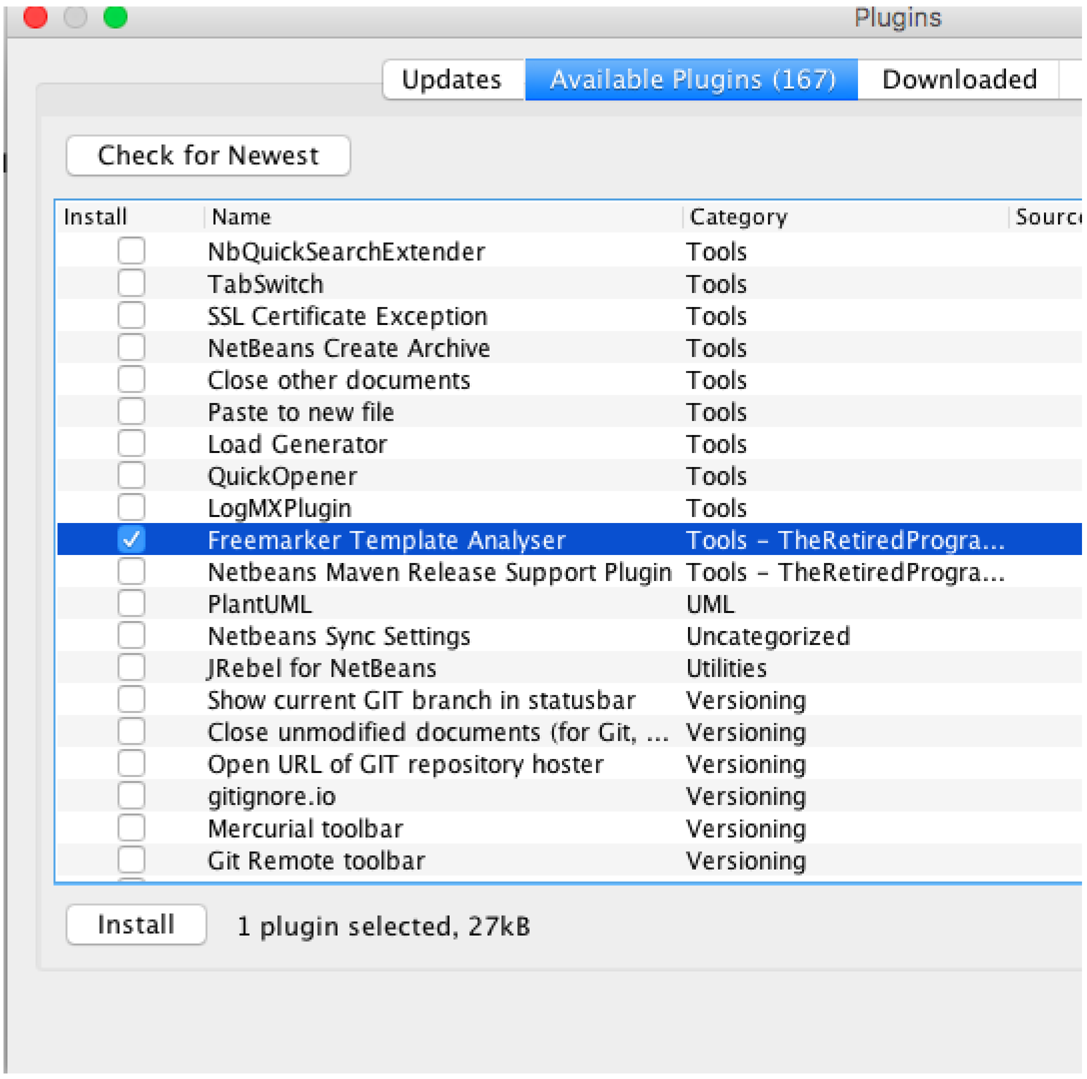
✓ Check for updates automatically

URL: http://www.theretiredprogrammer.uk/updatecentre/updates.xml

Now select the "Available Plugins" tab. You should now find a list of all available plugins, including a set of The-Retired-Programmer plugins.

If you can’t find any entries you could press the "Check for Newest" button to refresh this list, after which the new entries should be visible.

Select one or more plugins and then press the Install button. This will install the plugins into your IDE.

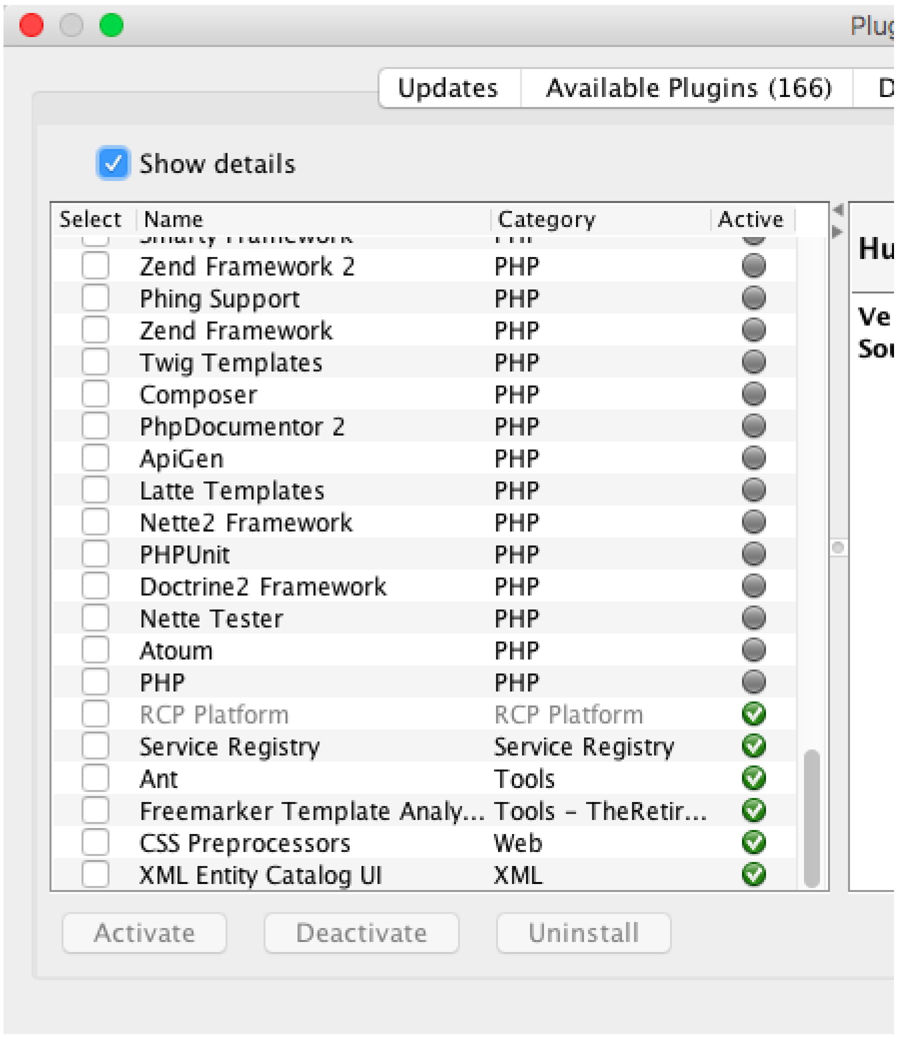


Install	Name	Category	Sourc
☐	NbQuickSearchExtender	Tools	
☐	TabSwitch	Tools	
☐	SSL Certificate Exception	Tools	
☐	NetBeans Create Archive	Tools	
☐	Close other documents	Tools	
☐	Paste to new file	Tools	
☐	Load Generator	Tools	
☐	QuickOpener	Tools	
☐	LogMXPlugin	Tools	
✓	Freemarker Template Analyser	Tools - TheRetiredProgra...	
☐	Netbeans Maven Release Support Plugin	Tools - TheRetiredProgra...	
☐	PlantUML	UML	
☐	Netbeans Sync Settings	Uncategorized	
☐	JRebel for NetBeans	Utilities	
☐	Show current GIT branch in statusbar	Versioning	
☐	Close unmodified documents (for Git, ...	Versioning	
☐	Open URL of GIT repository hoster	Versioning	
☐	gitignore.io	Versioning	
☐	Mercurial toolbar	Versioning	
☐	Git Remote toolbar	Versioning	

1 plugin selected, 27kB

During this process a warning is raised indicating that the plugin(s) are not signed and so may be a risk. By pressing the Continue putton the installation wil continue.

To confirm this has occurred you can switch to the Installed tab and view the installed plugins. Note you will have to tick the box labelled "Show detail" on this tab in order to list all plugins.



Select	Name	Category	Active
☐	Zend Framework 2	PHP	●
☐	Phing Support	PHP	●
☐	Zend Framework	PHP	●
☐	Twig Templates	PHP	●
☐	Composer	PHP	●
☐	PhpDocumentor 2	PHP	●
☐	ApiGen	PHP	●
☐	Latte Templates	PHP	●
☐	Nette2 Framework	PHP	●
☐	PHPUnit	PHP	●
☐	Doctrine2 Framework	PHP	●
☐	Nette Tester	PHP	●
☐	Atoum	PHP	●
☐	PHP	PHP	●
☐	RCP Platform	RCP Platform	✓
☐	Service Registry	Service Registry	✓
☐	Ant	Tools	✓
☐	Freemarker Template Analy...	Tools - TheRetir...	✓
☐	CSS Preprocessors	Web	✓
☐	XML Entity Catalog UI	XML	✓

Alternatively the plugin can be downloaded directly from Maven Central. The current release is v5.0.1, which can be downloaded from Maven Central.

The nbm file can be placed anywhere in the filesystem on your machine, and loaded manually into the IDE using the Plugin Dialog.

Developer Notes

The source code is available on GitHub.

The build/dependency manangement system is Maven.

The development environment uses the Netbeans IDE, but this is not mandatory, any development environment which can work with a Maven project can be used.

Source Code

Sources for this release can be found here.

Alternatively the project can be forked from here to obtain the whole source tree. Tag v5.0.1 refers to the sources relevant to this release.

Online Javadoc for this release can be found here, whilst a downloadable jar file containing the same Javadoc is available from here

Developers modifying the source code are encouraged to submit their changes back to the project using a Pull-Request (for consideration for inclusion in a future release).

Release Notes

Release 5.0.1

Updated build process relating to auto update.

Release 5.0.0

Updated to work with Netbeans 8.2; Major update to the release process to use Maven Central for delivery; Documenation now delivered via GitHub-Pages

Release 4.0.26

Updated to work with NetBeans 8.1; Revisions to internal implementation of forms.

Release 4.0.20

Added creation of separate entity source classes and any required table classes (for use with node editors).

Release 4.0.15

Build Process Update - uses latest(V2) POMs. No user or functional changes.

Release 4.0.14

Major update/refactoring of NBPCG generated code (and its associated libraries: nbpcg-library 3.1.7 and lindos 3.1.7). Includes explicit generation of additional fields (timestamps, ids and indexes), restructuring of the data access to utilise new APIs and other changes to templates and XSLT generated definintions