

AtomWeaver

AtomWeaver Reference

AtomWeaver Reference

The AtomWeaver Reference gives you detailed information about AtomWeaver commands and features.

[Menus](#) - Explanation of all available menu commands.

[Context Menu](#) - Explanation of all available context menus.

[Important Folders](#) - A reference to AtomWeaver's most important folders.

Menus

Reference on AtomWeaver menu commands:

[File](#)
[Atom](#)
[Tree](#)
[Library](#)
[Project](#)
[View](#)
[Tools](#)
[Help](#)

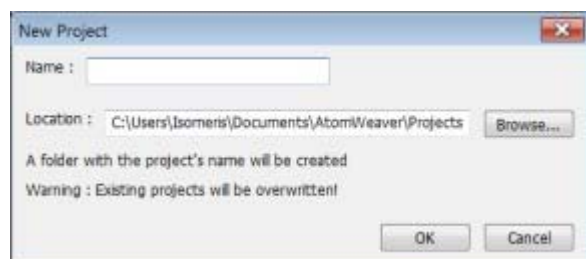
File

The *File* menu helps you manage your AtomWeaver projects.

[New Project...](#)
[Open Project...](#)
[Save Project](#)
[Exit](#)

New Project...

Start a new project from scratch. The new project will not be empty, as at least the *Root* Atom Library will be loaded. This is due to the fact that any Atom Template must ultimately inherit from *root_default*, so at least that Atom Template must be present from the start.



The New Project dialog is shown. You will be asked to give a name to the new project. This name will be used as the name of the file that stores your project on disk.

AtomWeaver default project location is set through the [Tools - Options](#) menu. You can, however, choose at this point any other location you like for your project file.

Open Project...

Open an existing project. The standard Windows Open File dialog box is shown. Choose the project file you want to open and press OK.

After selecting the project, AtomWeaver loads the database into memory, and then rebuilds it. You'll see two consecutive progress bars. Each bar reports on the two operations mentioned.

Important note: If the project was created with an older version of AtomWeaver, the project will be automatically converted to the current version, even if you don't save the project.

No code is generated after opening a project and as such the Files module will be empty. You'll need to select [Build - Generate Code](#) to generate code from the project.

Save Project

Save the current project to file. The destination file is the same you originally specified when opening or creating the project.

Exit

Exits AtomWeaver. You will be asked to save the project if there are any unsaved Atoms.

Atom

The commands on the *Atom* menu act on individual Atoms.

Cut

Prepare the currently selected Atom and its branch to be cut from its current position and moved to another location. This is a preparation operation and nothing changes in your model if you execute this command.

Only Atom Instances can be marked to be moved. Auto-Generated Slave Instances qualified as *Blocked* cannot be marked to be moved.

If the Atom Instance you've marked is an Auto-Generated Master Instance, after the move all its Slave Instances will be removed.

Keyboard shortcut : CTRL-SHIFT-X

Copy

Prepare the currently selected Atom and its branch to be copied into another location. This is a preparation operation and nothing changes in your model if you execute this command.

Only Atom Instances can be marked to be copied.

Keyboard shortcut : CTRL-SHIFT-C

Paste as Child

Move or copy a single Atom Instance or branch of Atoms as a child of the current selection.

The *Atom meta-metamodel* will be respected, so you can only paste an Atom Instance on places where an Atom Instance is Allowed: under other Atom Instances, under an Atom Template or under a Library's Include Organizers.

If you paste an Atom Instance into an Atom Template, it will be automatically duplicated to Derived Templates and all instances of that Template.

Keyboard shortcut : CTRL-SHIFT-V

Paste as Sibling

Move or copy a single Atom Instance or branch of Atoms as a sibling of the current selection. The pasted Atom will be placed after the current selection.

The *Atom meta-metamodel* will be respected, so you can only paste an Atom Instance on places where an Atom Instance is Allowed: under other Atom Instances, under an Atom Template or under a Library's Include Organizers.

If you paste an Atom Instance into an Atom Template, it will be automatically duplicated to Derived Templates and all instances of that Template.

Keyboard shortcut : CTRL-SHIFT-B

Atom Template Admin Wizard

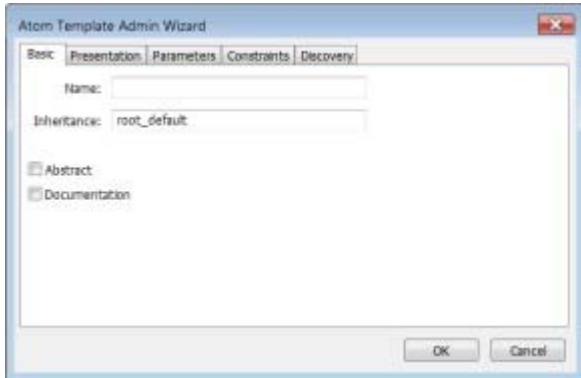
Invokes the Atom Template Admin Wizard. This wizard helps you configure your Atom Templates through several pages, creating the corresponding Admin Section code on exit. It can be used to create a new Atom Template, or edit an existing one.

If the current Atom Template's Admin Section contains errors, you must manually solve them before being able to use the wizard.

This is a Premium Feature

This feature is not available in *Free Mode*. To make use of this feature you need to be running AtomWeaver in [Full Mode](#).

Basic Page



The Basic page lets you define some basic but important aspects of your Atom Template:

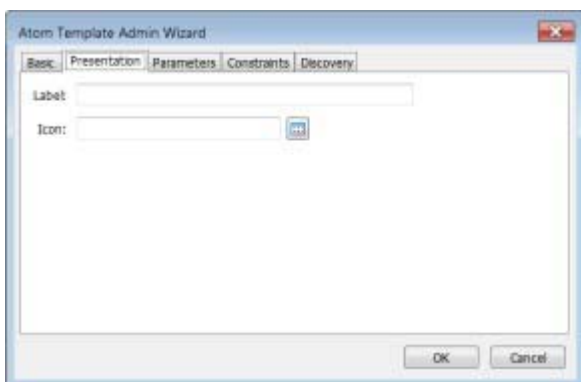
Name : The name of your Atom Template, that should be prefixed with the Atom Library's own prefix. Mandatory.

Inheritance : The name of the Atom Template that the one you are editing derives from. Mandatory. If you don't want to inherit anything, use *root_default*.

Abstract : If selected, the Atom Template will be abstract that is, it cannot be instantiated.

Documentation : If selected, the Atom Template will be a documentation-only Atom, and will help the generator optimize its performance by not running any code from it.

Presentation Page



The Presentation page lets you edit attributes that are related to the display of the Atom Template.

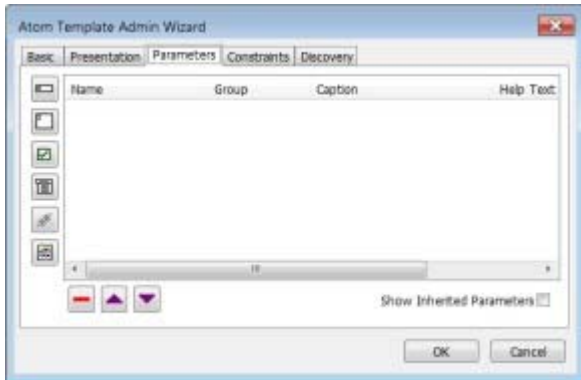
Label : The expression that will be parsed to obtain the final label of an Instance of this Template. Can be a constant string. Can be empty.

Icon : The name of the icon that will be shown on the model tree, on every Instance of this Template. Can be empty.



Icon Manager Button : Invokes the Icon Manager to help you visually select an icon to use. Double-clicking an icon will copy its name to the corresponding edit box. Not available if you are running AtomWeaver in [Free Mode](#).

Parameters Page



The Parameters page is the most useful page in the wizard as it lets you create and edit the Atom Template's parameters, as well as check what parameters are being inherited from Base Templates.



Add a Text Parameter : Opens the *Text Parameter* dialog to help you add a new Template parameter of the *Text* type (`param_text`).



Add a Text Block Parameter : Opens the *Text Block Parameter* dialog to help you add a new Template parameter of the *Block* type (multiple-line text, `param_block`).



Add a Boolean Parameter : Opens the *Boolean Parameter* dialog to help you add a new Template parameter of the *Boolean* type (`param_bool`).



Add a List Parameter : Opens the *List Parameter* dialog to help you add a new Template parameter of the *List* type (`param_list`).



Add an Atom Link Parameter : Opens the *Link Parameter* dialog to help you add a new Template parameter of the *Atom Link* type (`param_link`).



Add a Code Area Parameter : Opens the *Area Parameter* dialog to help you add a new Template parameter of the *Code Area* type (`param_area`).



Delete Selected Parameter : Removes the currently selected parameter from the list. Inherited parameters, if listed, cannot be removed: You have to do that from the Atom Template that defines them.



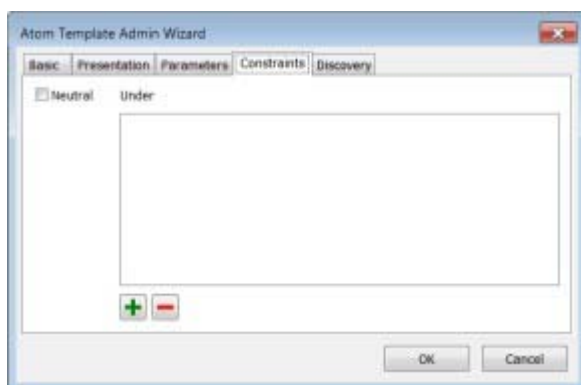
Move Parameter Up : Moves the currently selected parameter up in the list. If inherited parameters are listed, the moving parameter will not be above any inherited parameter. Inherited parameters cannot be moved: You have to do that from the Atom Template that defines them.



Move Parameter Down : Moves the currently selected parameter down in the list. Inherited parameters cannot be moved: You have to do that from the Atom Template that defines them.

Show Inherited Parameters : If selected, all parameters that the Atom Template inherits will be listed too. However, they cannot be edited: You have to do that from the Atom Template that defines them.

Constraints Page



The Constraints page will help you define this Atom Template's construction constraints. If no constraints are listed, then the Template will be considered a *Free Template*, and can be created anywhere (any place where an Atom Instance can be created). Read more about construction constraints [here](#).

Remember that unlike other Atom Template attributes, construction constraints are not inherited.

Neutral : If selected, this Atom Template will be invisible to its children regarding construction constraints: Their constraints will be checked, not against this Template, but against the parent Atom's Template. Learn more about neutral Atoms [here](#).

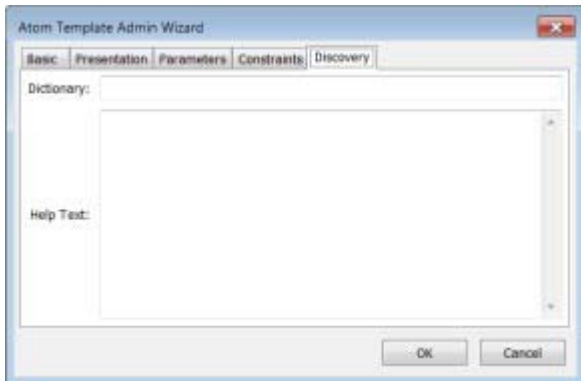


Add New Constraint : Add a new construction constraint to the list. Typically you will list those Atoms that can have Atoms of the Template's type under it. If you want this Template to be instantiated at the model tree's root, you should use the tree's name prefixed by a '#'. The project's main tree would therefore be identified as *#model*.



Delete Selected Constraint : Remove the currently selected constraint from the list.

Discovery Page



The Discovery page lets you set information that will help developers find and understand the uses for this Template, as well as identify their Instances by name.

Dictionary : Expression that will be parsed for each Instance of this Template and whose result will be entered at the Atom Dictionary, and associated to the corresponding Instance.

Help Text : Text that explains this Template and its uses.

Save Atom Changes

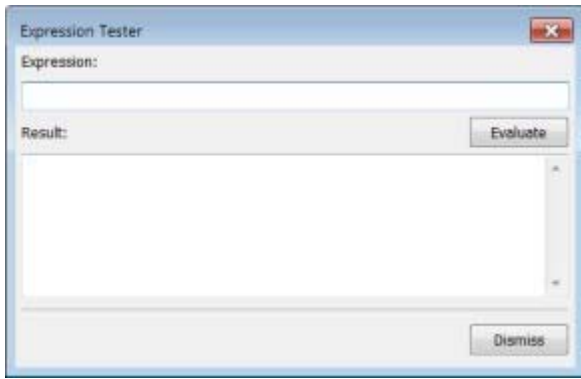
Submit the changes you made to the current Atom and updates the project accordingly. One or more Atom Trees might be updated. Depending on the changes you did and on the Atom type you changed, the submit and update operation may take a few seconds on large models.

Keyboard shortcut : F5

Test Expression...

Evaluate an expression in the context of the currently selected Atom Instance.

Sometimes you may need to check if an expression is returning the expected value or Atom. Sometimes that expression is used on an Instance variable and you have no way to debug it.



The expression tester lets you enter an expression that will be evaluated by feeding it to the [eval\(\)](#) function. The result will be presented on the bottom box.

- If the result is a string, such string is printed.
 - If the result is an Atom, its identification is printed.
 - If an error occurs, the message *Error : Could not evaluate expression* is printed.
-

Show Properties...

Show a window with information regarding the currently selected Atom.

When an Atom is in an invalid state (shown with a red label), you can use this command to get more details about what is wrong with it.

Information shown on the window:

For all Atom types:

- Atom type
- Creation date/time
- Last change date/time

Additionally, for Atom Templates:

- Base Template
- Derived Templates (if any)

Additionally, for Atom Instances:

- Integrity problems (if any), shown in red
 - Master / Slave information and Instances (if any)
 - Local variables
 - Local temporary variables
-

Tree

Commands on the *Tree* menu make selections or changes on the current module's Atom Tree.

Add Atom Organizer Child

Create a new Atom Folder Organizer as child of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Organizer under the current Atom.

A dialog asking for the Organizer's label is shown:



After pressing OK, the Atom Organizer is created and automatically selected.

Add Atom Organizer Sibling

Create a new Atom Folder Organizer as a sibling of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Organizer under the sibling's parent.

A dialog asking for the Organizer's label is shown:



After pressing OK, the Atom Organizer is created and automatically selected.

Add Atom Template Child

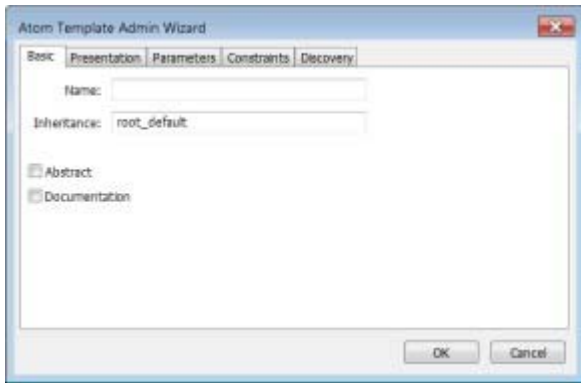
Create a new Atom Template as child of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Template under the current Atom.

If you are running AtomWeaver in [Free Mode](#), the New Atom Template dialog is shown:



Enter the name of the new Atom Template and specify its Base Template (default is *root_default*). The [prefix](#) of the new Atom Template must be the same of its Library's. For instance, on a Library whose prefix is *java*, all its Atom Templates must start with *java_*.

If you are running AtomWeaver in [Full Mode](#), the Atom Template Admin Wizard is shown:



Learn how to use the Atom Template Admin Wizard [here](#).

After pressing OK, the Atom Template is created and automatically selected. Instance Slaves and Masters from Base Templates are automatically created under the new Atom Template.

Add Atom Template Sibling

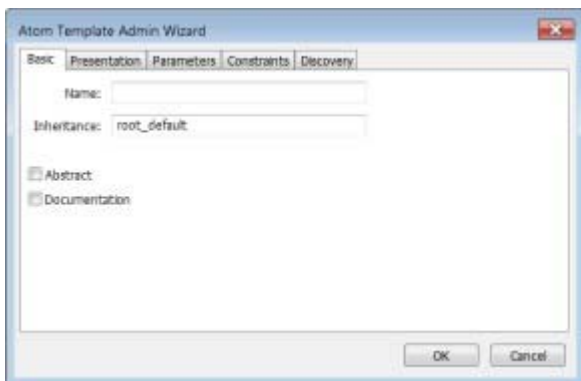
Create a new Atom Template as a sibling of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Template under the current sibling's parent.

If you are running AtomWeaver in [Free Mode](#), the New Atom Template dialog is shown:



Enter the name of the new Atom Template and specify its Base Template (default is *root_default*). The [prefix](#) of the new Atom Template must be the same of its Library's. For instance, on a Library whose prefix is *java*, all its Atom Templates must start with *java_*.

If you are running AtomWeaver in [Full Mode](#), the Atom Template Admin Wizard is shown:



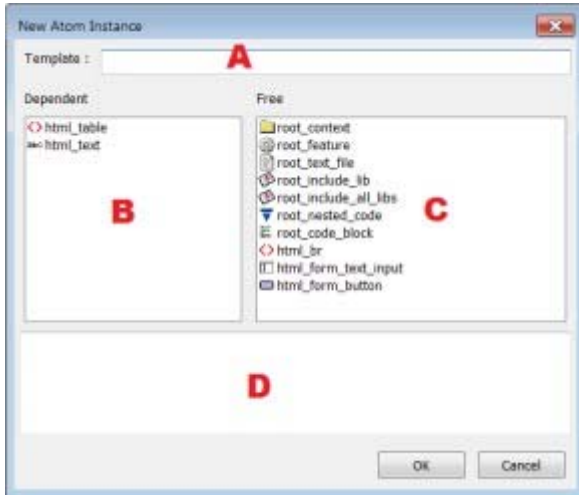
Learn how to use the Atom Template Admin Wizard [here](#).

After pressing OK, the Atom Template is created and automatically selected. Instance Slaves and Masters from Base Templates are automatically created under the new Atom Template.

Add Atom Instance Child

Create a new Atom Instance as child of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Instance under the current Atom.

The New Atom Instance dialog is shown:



This dialog lists all Atom Templates that you can instantiate at this point.

The type(s) of Atom Instances that you are allowed to create under the current Atom is dictated by their [creation constraints](#).

There are two kinds of Atom Instances you can select: *Dependent* and *Free*. Dependent Atoms **(B)** are those that are allowed by their creation constraints. Free Atoms **(C)** are those that do not have any creation constraint set, and therefore can be created under any parent.

To select an Atom Template, you can:

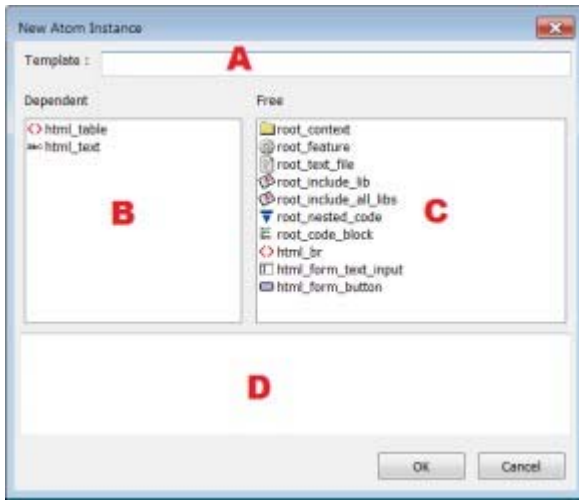
- Type its name directly on the Template box **(A)**
- Click on its label on either the B or C boxes. In this case, its [help text](#) is presented in the description area **(D)**.

Pressing *Enter*, clicking on *OK*, or double-clicking on the Atom Template label, accepts your selection, closes the dialog and creates an instance of the selected Atom template.

Add Atom Instance Sibling

Create a new Atom Instance as a sibling of the currently selected Atom. This command may not be available if ABSE rules do not allow the creation of an Atom Instance under the current sibling's parent.

The New Atom Instance dialog is shown:



This dialog lists all Atom Templates that you can instantiate at this point.

The type(s) of Atom Instances that you are allowed to create under the current Atom is dictated by their [creation constraints](#).

There are two kinds of Atom Instances you can select: *Dependent* and *Free*. Dependent Atoms (**B**) are those that are allowed by their creation constraints. Free Atoms (**C**) are those that do not have any creation constraint set, and therefore can be created under any parent.

To select an Atom Template, you can:

- Type its name directly on the Template box (**A**)
- Click on its label on either the B or C boxes. In this case, its [help text](#) is presented in the description area (**D**).

Pressing *Enter*, clicking on *OK*, or double-clicking on the Atom Template label, accepts your selection, closes the dialog and creates an instance of the selected Atom template.

Move Atom Up

Moves up the selected Atom, among its own siblings. According to the Atom's type, the operation may or may not be allowed.

The operation will not be performed if:

- The Atom is already the first child on its branch.
- The Atom is an Include Organizer: This kind of [Atom Organizers](#) cannot move.
- The Atom is a [Library Organizer](#) trying to move above the Root Library's Organizer: The Root Library is always the first Atom Library.
- The Atom is a Blocked Auto-Generated Instance Slave.
- The Atom is a non-Blocked Auto-Generated Instance Master, and is already the first among its non-blocked siblings.

Move Atom Down

Moves down the selected Atom, among its own siblings. According to the Atom's type, the

operation may or may not be allowed.

The operation will not be performed if:

- The Atom is already the last child on its branch.
 - The Atom is an Include Organizer: This kind of [Atom Organizers](#) cannot move.
 - The Atom is the Root [Library Organizer](#): The Root Library is always the first Atom Library, so it cannot move.
 - The Atom is a Blocked Auto-Generated Instance Slave.
 - The Atom is a Blocked Auto-Generated Instance Master, and is already the last among its blocked siblings.
-

Move Atom Left

Moves the selected Atom to the left, moving to the current parent's parent Atom. The moving Atom will be the last one in its new branch. According to the Atom's type, the operation may or may not be allowed.

The operation will not be performed if:

- Construction constraints don't allow it.
 - The Atom is already under the tree root.
 - The Atom is an Include Organizer: This kind of [Atom Organizers](#) cannot move.
 - The Atom is a Blocked Auto-Generated Instance Slave.
 - The Atom is an Auto-Generated Instance Master, and is already directly under an Atom Template.
-

Move Atom Right

Moves the selected Atom to the right, becoming a child of its current upper sibling. The moving Atom will be the last one in its new branch. According to the Atom's type, the operation may or may not be allowed.

The operation will not be performed if:

- Construction constraints don't allow it.
 - The Atom is the first child under its parent.
 - The Atom is an Include Organizer: This kind of [Atom Organizers](#) cannot move.
 - The Atom is a Blocked Auto-Generated Instance Slave.
 - The Atom is an Atom Template or Atom Organizer, and its upper sibling is an Atom Template.
-

Delete Selected Atom

Delete the selected Atom.

Some Atoms cannot be deleted:

- The Root Library's Organizer: The Root Library cannot be deleted from the project.
- An Include Organizer: This kind of [Atom Organizers](#) cannot be deleted. They are only deleted

when the Atom Library they belong to is removed from the project.

- The *root_default* Atom Template.

Deleting an Atom is normally not an isolated operation. When that happens, other Atoms that depend on the one you are deleting must be deleted too. This may trigger a chain reaction of delete operations. **Be careful when using this command!** Generally speaking, when you delete an Atom, its branch is deleted too. Then, for each deleted Atom:

- If the deleted Atom is a Library Organizer, the corresponding Atom Library is removed from the project.
- If the deleted Atom is an Atom Template, its Instances and derived Templates are deleted too.
- If the deleted Atom is an [Auto-Generated Instance Master](#), its slaves are deleted too.

Before proceeding, the command asks for a confirmation and presents the number of Atoms that will be deleted:



Collapse All

Collapse all branches in the current tree.

The current selection may change after this operation.

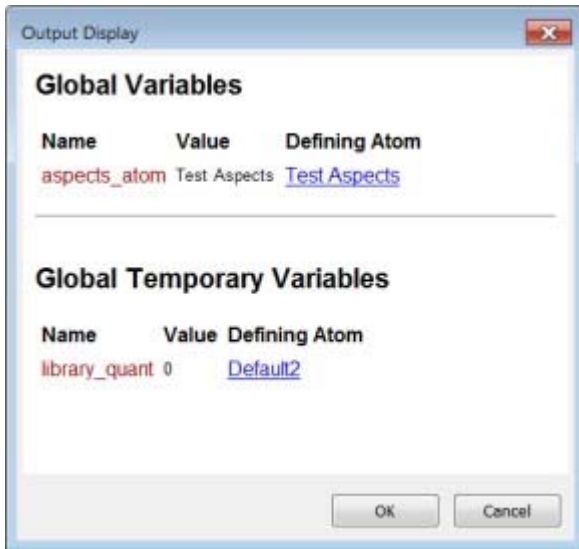
Expand All

Expand all branches in the current tree.

View Tree's Global Variables

View all global variables set on this tree by its Atom Instances.

This command not available on the Library Tree.



The list of permanent and temporary variables are shown. A link to the defining Atom is also supplied. You can click on a link to directly jump to the referenced Atom Instance.

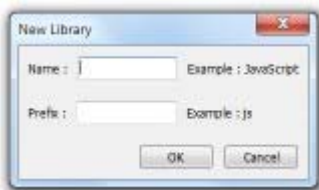
Library

The *Library* menu helps you manage your local Atom Libraries and those on the Repository.

Learn more about Atom Libraries on the [ABSE Guide](#).

New...

Create a new Atom Library. The New Library dialog is shown:



Name

This should be a short and descriptive name for your library. Usually, the library's domain is used: *C++*, *Java*, *User-Interface*, *Insurance*, *Banking*, *Web Development*, etc.

Prefix

This prefix serves both as Library ID and as a prefix for all Atom Templates you'll create under this Library. This prefix must be only composed of numbers and letters, and cannot start with a number. Learn more [here](#).

Delete

Delete an Atom Library from the project.

A Library Organizer must be selected to perform this operation. The Atom Library headed by the selected Organizer will be removed from your project.

This is a very destructive operation and should be used with care!

All the Atom Library's contents, as well as all Atoms that have been instantiated from Templates on this Library will be removed. In addition, Atom Templates on other Libraries that inherit from Templates on the Library you are deleting will also be removed. In consequence, their own instantiated Atoms will be removed too.

Save To Repository

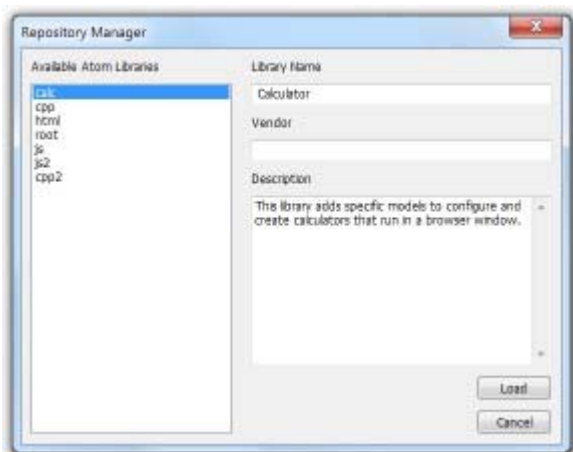
Save the currently selected Atom Library to the local Library Repository.

The project's Library will be exported as a *library_prefix.alx* file to the *Repository* folder under AtomWeaver's working folder, where *library_prefix* is the prefix of the library being exported. As an example, if your Library's prefix is *java*, a *java.alx* file will be created on the Library Repository.

If there is already a Library with the same name on the Repository, you will be prompted to decide whether you want to continue. If you do, the current [Reference Library](#) will be replaced with the one being exported.

Load From Repository...

Load a [Reference Library](#) from the Library Repository. The *Repository Manager* dialog box will be shown:



A list of the available libraries are shown on the left list. Each library is identified by its prefix and vendor ID in the form *<library_id>-<vendor_id>*. By picking one item from the list, additional information about that library is presented.

To load a library into the workspace, select the corresponding item on the list and press *Load*. The library is loaded into the workspace and its Atoms are regenerated.

Press *Cancel* to dismiss the *Repository Manager*.

Update Repository Index

Update the Library Repository Index file.

The Library Repository Index file caches information from all libraries in the repository so that load and save operations on the Repository can be made faster.

If you have externally changed a library file, or if you imported new or updated Libraries, you need to update the Index file. If you fail to do so, information displayed on the *Repository Manager* dialog may be incorrect.

Update Library Dependencies

Update the internal dependency relationships on all loaded Atom Libraries.

AtomWeaver will scan your project and will determine the library dependencies for each of the loaded Atom Libraries. Library dependencies are determined by:

- which Atom Templates are inherited;
- which Atom Instances are instantiated.

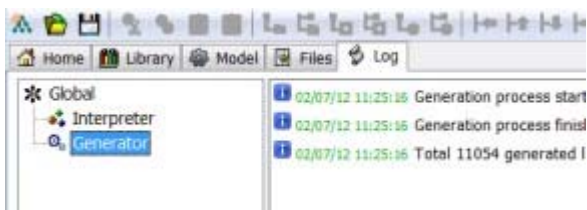
Because all Atom Templates will ultimately inherit from *root_default*, it's certain that every Atom Library will depend, at least indirectly, on the *Root Library*.

Project

The *Project* menu helps you obtain your final results.

Generate Code

Start the code generation operation. AtomWeaver will execute your current model and, according to the code that the model contains, will generate the corresponding artifacts (code, files, data, etc). The results of this operation will be placed in the Files Module. The generator's messages are logged in the *Generator* log in the Log Module:



If an error occurs during model execution, a warning message pop-up will be shown, and any error messages will be placed in the same log. You can then use the context menu on an error message to jump to the Atom Instance (and, if possible, the Atom Template) where the error occurred.

Keyboard shortcut : F7

Settings...

Sets project-specific options.



Here you can set specific options for the project you are working on. The project's main folders are automatically determined when you start your new project. You can later move your folders but you should then update the new folders here.

Project Folder : This is the folder where your main project file (".awp") file is located.

Input Folder : This folder will contain all user-supplied resources to be used on your project.

Output Folder : This will be the root folder of the code generator. Files generated at "root" will be placed here, and any sub-folders will be created here.

Generator Maps Folder : This folder will mirror the *Output Folder*. The files placed here do not contain the generated code. Each line from these files will contain the ID of the Atom that generated the corresponding line from the *Output Folder*.

View

The *View* menu helps you change visualization aspects of your project or workspace.

Show Auto-Generated Instances

Show Atom Instances that were automatically created by the Atom Templates you instantiated.

Auto-Generated Instances on a model are shown with a green label when active, and with a light green label when inactive.

This command is only available if the current module manages an ABSE tree.

Hide Auto-Generated Instances

Hide Atom Instances that were automatically created by the Atom Templates you instantiated.

After selecting this command, all Auto-Generated Instances are removed from display (but not from the model). This is particularly useful if you want to get a simplified view of your model.

This command is only available if the current module manages an ABSE tree.

Show Parameter Names

Show the names of the Atom Template Parameters that correspond to the Atom Instance Variables being shown on its form.

Atom Template Parameters are directly mapped to Atom Instance Variables when an Atom Instance is created. However, when the form is displayed, only the Parameter's caption is shown. This makes it sometimes harder to identify a specific Template Parameter.

Displaying Atom Template Parameter names helps the Atom Library architect work on the model debugging process.

This command is only available if the current module manages an ABSE tree (eg. *Library* or *Model*).

Hide Parameter Names

Hide the display of Atom Template Parameter names on Atom Instance forms.

Hiding Atom Template Parameter names is useful if the model developer or end-user needs a cleaner form by removing unnecessary information.

This command is only available if the current module manages an ABSE tree (eg. *Library* or *Model*).

Tools

This menu contains tools to help you on several tasks.

Model Integrity Report

Analyze the ABSE trees on the project and report any inconsistencies.

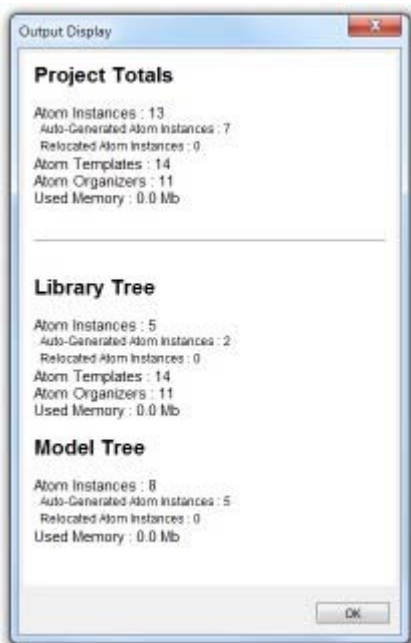
The *Model Integrity Report* command reports any problems that may be affecting the consistency of your models. AtomWeaver analyzes all the existing trees in your project and presents a summary of the results:



If invalid Atoms are reported, you should then use the *Go To Previous Invalid Atom* or *Go To Next Previous Invalid Atom* toolbar commands to navigate on all the project's invalid Atoms. You should use these commands on the tree(s) that were reported as having invalid Atoms.

Statistics...

Show simple statistics about the project's trees and Atoms:



You'll get a report, per project tree, of their included Atoms, as well as the corresponding memory usage.

Options...

Set AtomWeaver-specific options.

This command lets you set specific options regarding the operation of AtomWeaver.

The available options are:

Library Repository Folder : The location of the [Atom Library Repository](#). AtomWeaver will use this setting to locally store your [Reference Libraries](#).

Default Libraries : A comma-separated list of libraries to be automatically loaded from the Library Repository on new projects.

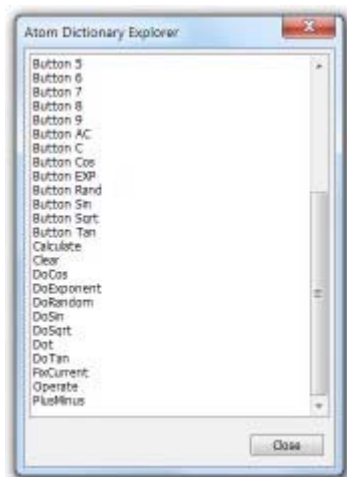
Projects Folder : The location where AtomWeaver will create new projects.

These options are stored in the *config.xml* file under AtomWeaver's [work folder](#).

Atom Dictionary Explorer

Display the entries on the project's Atom Dictionary.

The Atom Dictionary Explorer dialog box is shown:



The Atom Dictionary is a list of strings that can be used to identify a particular Atom in the project. Double-clicking on an entry will take you directly to the Atom that is associated with that entry.

This is a Premium Feature

This feature is not available in *Free Mode*. To make use of this feature you need to be running AtomWeaver in [Full Mode](#).

Atom Lists Explorer

Display the Atom Lists Explorer dialog.

The Atom Lists Explorer lets you check what Atom lists are define and what are their items. The dialog used to explore Atom lists is similar to the dialog used to select an Atom from a list:

You can't add or remove Atoms to the lists because that management is automatically made by AtomWeaver based on the usage of the `list()` command.

This is a Premium Feature

This feature is not available in *Free Mode*. To make use of this feature you need to be running AtomWeaver in [Full Mode](#).

Icon Manager

This command invokes the Icon Manager dialog window:



The icon manager lets you check what icons are available for you to use on your Atoms. A label is presented next to each displayed icon. You can use that label to set a Atom Template's icon using the [icon\(\)](#) command.

As a convenience, double-clicking on an icon on the list will automatically copy its label to the clipboard and close the manager.

Alternatively, you can use the close button to dismiss the Icon Manager.

This is a Premium Feature

This feature is not available in *Free Mode*. To make use of this feature you need to be running AtomWeaver in [Full Mode](#).

Export Branch

Export a Tree branch to an external file.

You can export a branch of one of the project's trees to an XML file. That file can then be imported to another project.

Exporting a branch headed by a [Library Organizer](#) effectively exports the complete Atom Library.

Import Branch

Import an Atom tree branch from an external file.

This command imports (and regenerates) an Atom tree branch that was stored as an XML file.

Loading incomplete branches from external projects may result in model inconsistencies like unknown Atom Templates and missing links.

Help

The *Help* menu lets you access documentation about ABSE and AtomWeaver, as well as information on this application and how to license it.

Help Topics...

Show AtomWeaver help documentation.

[Click here](#) to go to the documentation's home page.

Tutorials...

Jump directly to the tutorials section on the help documentation.

Some tutorials are started from scratch (an empty project) but others may require you to load a sample project first.

License...

Show and manage AtomWeaver's license.

This command lets you check your current AtomWeaver license as well as adding a new one. From this dialog you can also request a trial license.

There are [several licensing options](#) available.

About AtomWeaver...

Display program credits and version, as well as the ABSE version that is implemented. Additionally, other author's credits are displayed as per their respective license agreements.

Tool bar

This section lists and explains the use of AtomWeaver's tool bar icons.

New Project



Start a new project from scratch.

This button is a link to the [New Project...](#) menu command.

Open Project



Open an existing project.

This button is a link to the [Open Project...](#) menu command.

Save Project



Save the current project to file.

This button is a link to the [Save Project](#) menu command.

Cut



Prepare the currently selected Atom and its branch to be cut from its current position and moved to another location.

This button is a link to the [Cut](#) menu command.

Copy



Prepare the currently selected Atom and its branch to be copied into another location.

This button is a link to the [Copy](#) menu command.

Paste As Child



Move or copy a single Atom Instance or branch of Atoms as a child of the current selection.

This button is a link to the [Paste as Child](#) menu command.

Paste As Sibling



Move or copy a single Atom Instance or branch of Atoms as a sibling of the current selection.

This button is a link to the [Paste As Sibling](#) menu command.

Add Atom Organizer Child



Create a new Atom Folder Organizer as child of the currently selected Atom.

This button is a link to the [Add Atom Organizer Child](#) menu command.

Add Atom Organizer Sibling



Create a new Atom Folder Organizer as a sibling of the currently selected Atom.

This button is a link to the [Add Atom Organizer Sibling](#) menu command.

Add Atom Template Child



Create a new Atom Template as child of the currently selected Atom.

This button is a link to the [Add Atom Template Child](#) menu command.

Add Atom Template Sibling



Create a new Atom Template as a sibling of the currently selected Atom.

This button is a link to the [Add Atom Template Sibling](#) menu command.

Add Atom Instance Child



Create a new Atom Instance as child of the currently selected Atom.

This button is a link to the [Add Atom Instance Child](#) menu command.

Add Atom Instance Sibling



Create a new Atom Instance as a sibling of the currently selected Atom.

This button is a link to the [Add Atom Instance Sibling](#) menu command.

Move Atom Left



Moves the selected Atom to the left, moving to the current parent's parent Atom.

This button is a link to the [Move Atom Left](#) menu command.

Move Atom Up



Moves up the selected Atom, among its own siblings.

This button is a link to the [Move Atom Up](#) menu command.

Move Atom Down



Moves down the selected Atom, among its own siblings.

This button is a link to the [Move Atom Down](#) menu command.

Move Atom Right



Moves down the selected Atom, among its own siblings.

This button is a link to the [Move Atom Right](#) menu command.

Delete Atom



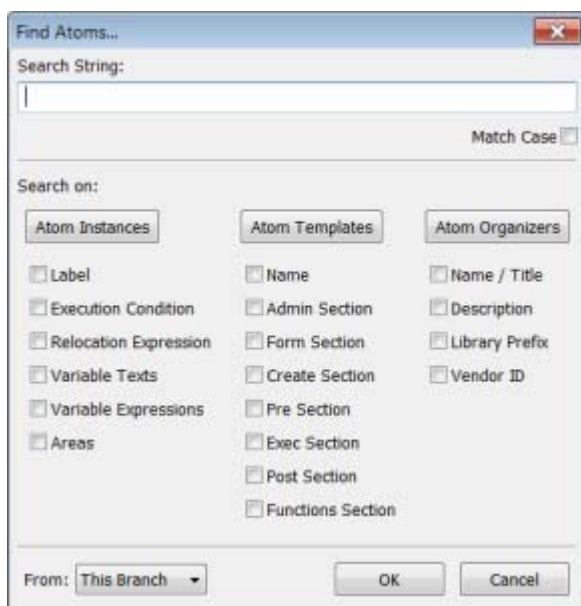
Delete the selected Atom.

This button is a link to the [Delete Selected Atom](#) menu command.

Find Atoms



Search for Atoms in the project and display the found ones as a Selection List. After invoking this command, the *Find Atoms...* dialog is shown:



The search is made based on a string you specify, and be made case sensitive or insensitive.

Search On:

Specify what parts of your Atoms will be searched. You can toggle the search on an individual

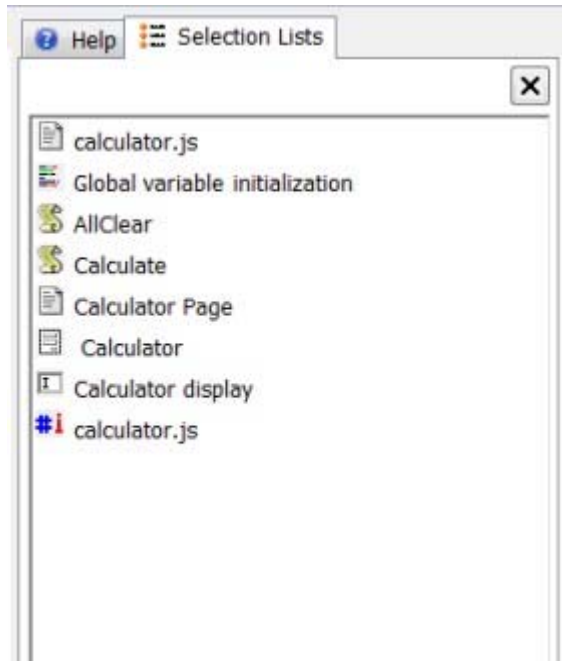
Atom part by clicking on the corresponding check box. If you press the *Atom type* buttons, all corresponding check boxes can be (un)checked.

Search scope

Select the scope of this search operation. You can search on the branch headed by the current selection, search on the tree you are currently working on, or search the complete project.

Search Results

After you press OK, the search operation takes place, and the results, if any, are displayed on the Auxiliary Panel as a Selection List. A dedicated page on this panel shows your results:



You can now double-click on an item to jump to its location on the tree.

Test Expression on Atom



Evaluate an expression in the context of the currently selected Atom Instance.

This button is a link to the [Test Expression...](#) menu command.

Go To Previous Invalid Atom



Jump to the nearest Invalid Atom that is above the current selection. You can then use several other commands to find out what is wrong with the Atom:

Atom - [Show Properties](#) menu command: The properties report window will show you the reasons for the Atom to be invalid. If the problem is on a variable expression evaluation, you

can also see here what is the invalid variable.

Show Related Errors context menu command: The error report window will list the exact error messages that were logged for the Invalid Atom.

Go To Next Invalid Atom



Jump to the nearest Invalid Atom that is below the current selection. You can then use several other commands to find out what is wrong with the Atom:

Atom - [Show Properties](#) menu command: The properties report window will show you the reasons for the Atom to be invalid. If the problem is on a variable expression evaluation, you can also see here what is the invalid variable.

Show Related Errors context menu command: The error report window will list the exact error messages that were logged for the Invalid Atom.

Show Auto-Generated Instances



Show Atom Instances that were automatically created by the Atom Templates you instantiated.

This button is a link to the [Show Auto-Generated Instances](#) menu command.

Hide Auto-Generated Instances



Hide Atom Instances that were automatically created by the Atom Templates you instantiated.

This button is a link to the [Hide Auto-Generated Instances](#) menu command.

Generate Code



Start the code generation operation.

This button is a link to the [Generate Code](#) menu command.

Atom Template Admin Wizard



Invokes the Atom Template Admin Wizard.

This button is a link to the [Atom Template Admin Wizard](#) menu command.

Show Available Commands



Shows a list of available ABSE commands and functions. This command is only available when editing an Atom Template.

The list is shown on the right-side Auxiliary Panel. The commands shown on the list are only those allowed in the Template Section currently being edited.

Click on a command name to access the corresponding online help page.

Context Menus

Reference on AtomWeaver pop-up menu commands:

[Atom Instance](#)
[Atom Template](#)
[Atom Organizer](#)

Atom Instance

The Atom Instance context menu is shown if you click the right mouse button on an ABSE tree that has an Atom Instance currently selected.

Add Atom Instance Child

Append a new regular Atom Instance as a child of the selected Atom Instance. An [Auto-Generated Atom Instance](#) is created instead if the currently selected Instance is on a branch headed by an Atom Template. In that case, your new Atom Instance will be immediately duplicated to all the Template's Instances.

Add Atom Instance Sibling

Add a regular Atom Instance next to the currently selected one. An Auto-Generated Atom Instance is created instead if the currently selected Instance is on a branch headed by an Atom Template. In that case, your new Atom Instance will be immediately duplicated to all the Template's Instances.

Go To Template

Jump directly to the Atom Template that the currently selected Atom is an Instance of. The Library module will be selected if it isn't the current module.

Go To Master

If the selected Atom Instance is an [Auto-Generated Slave Instance](#), jump directly to its Master Instance. Please note that the Atom's Master can be another Slave Instance. You may need to select this pop-up options several times until you reach the Top Master Instance, the only one that can be edited.

Go To Original Parent

If the selected Atom Instance has been relocated, jump to the Atom that was its parent before the relocation took place. Only Auto-Generated Slave Instances can be relocated, so if you want to edit the relocation expression you can use the *Go To Master* pop-up option to reach the Top Master Instance.

Show Related Errors

Show a list of all the errors that are currently associated with this Atom Instance. These messages are also displayed on the Log module.

Properties

Show detailed information about the selected Atom Instance, including creation and change dates, Auto-Generated Master and Slave Instances (if any), and locally defined variables.

Atom Template

The Atom Template context menu is shown if you click the right mouse button on the Library's ABSE tree that has an Atom Template currently selected.

Add Atom Organizer Sibling

Add a Folder Organizer next to the selected Atom Template.

Add Atom Template Sibling

Add an Atom Template next to the currently selected one.

Add Atom Instance Child

Add an [Auto-Generated Atom Instance](#) under the selected Atom Template.

Go To Base Template

Jump directly to the Atom Template from where the currently selected Atom Template derives from.

Rename

Rename this Atom Template. AtomWeaver will attempt to rename all references to this

Template, must you should make a final check after this operation.

Show Related Errors

Show a list of all the errors that are currently associated with this Atom Template. These messages are also displayed on the Log module.

Properties

Show detailed information about the selected Atom Template, including creation and change dates, Base Template and Derived Template(s).

Atom Organizer

The Atom Organizer context menu is shown if you click the right mouse button on the Library's ABSE tree that has an Atom Organizer currently selected.

Add Atom Organizer Child

Add a new Atom Organizer as child of the current one. This command is only available if the selected Organizer is a Library Organizer or a Folder Organizer.

Add Atom Organizer Sibling

Add a new Atom Organizer next to the currently selected one. If the selected Organizer is one of the Include Organizers, the new Atom will be created before them. This command is only available if the selected Organizer is a Folder Organizer or an Include Organizer.

Add Atom Template Child

Add a new Atom Template as child of the current Atom Organizer. This command is only available if the selected Organizer is a Library Organizer or a Folder Organizer.

Add Atom Template Sibling

Add a new Atom Template next to the selected Atom Organizer. If the selected Organizer is one of the Include Organizers, the new Atom will be created before them. This command is only available if the selected Organizer is a Folder Organizer or an Include Organizer.

Add Atom Instance Child

Add a new Atom Instance as child of the selected Atom Organizer. This command is only available if the selected Organizer is one of the Include Organizers.

Show Related Errors

Show a list of all the errors that are currently associated with this Atom Organizer. These messages are also displayed on the Log module.

Properties

Show detailed information about the selected Atom Organizer, including creation and change

dates.

Log

The Log context menu is shown if you click the right mouse button on the Log module's tree that has a previously selected Log.

Export to File...

Save the full contents of the selected log into an HTML file. You will be prompted to select a location and a name for the export file.

Log Entry

The Log Entry context menu is shown if you click the right mouse button on the Log module's list that has a previously selected log entry.

Copy

Copy the currently selected log entry into the clipboard.

Go To Atom Instance

Jump to the entry's associated Atom Instance. This option is only available if the entry is referring to a specific Atom Instance.

Go To Atom Template

Jump to the entry's associated Atom Template. This option is only available if the entry is referring to a specific Atom Template. For example, if the log entry is reporting an error during generation, the associated Atom Template contains the exact place where the error occurred.

Important Folders

There are some AtomWeaver folders that may be of interest to you.

Installation Folder

The Installation Folder has been specified by you or your system administrator during AtomWeaver installation. This folder mainly contains AtomWeaver's own binaries and resources.

AtomWeaver has some expansion capabilities that you can take advantage of. Some of these capabilities involve the installation of additional files under this folder.

License File

Your license file is located in this folder if AtomWeaver can write to it. Otherwise, the license file will be located at the Work Folder.

Log Entry Icons Folder

[installation_folder]\resources\log_entry_icons

This folder contains the icons that can be shown on the Log module, on a log entry. The icon's file title is the identifier you need to specify on the `log()` command. For instance, to use the *done.png* icon on a log entry, you should place the icon file in this folder, and specify *done* as the icon's ID on a `log()` command:

```
log("my_logs/test", "My log message", "done")
```

AtomWeaver uses the *info*, *warn* and *error* icons for itself. Therefore you should not remove these icons from this directory.

All icons must be 16x16 and saved in the PNG format. RGB(240,240,240) is used as transparent color.

Atom Icons Folder

[installation_folder]\resources\icons

Any icons to be used on Atoms are loaded by AtomWeaver from this folder. On startup, AtomWeaver will load:

- individual icons from this folder
- zip files from the *Packages* sub-folder

Icon packages located on the *Optional* sub-folder are not loaded. Instead, you can move these packages into the *Packages* folder so that AtomWeaver can load them.

Work Folder

The Work Folder is where AtomWeaver saves your work by default. This folder is named *AtomWeaver* and is usually located under your user profile, under the documents folder. A typical location on Windows 7 is `C:\Users\your_user\Documents\AtomWeaver`.

Configuration File

Global AtomWeaver options and other operational settings are stored on the *config.xml* file, located on the Work Folder. You can safely delete this file to reset AtomWeaver's settings, but remember that global settings (defined by the *Tools - Options* menu command) will be reset too.

The Repository Folder

The Repository Folder holds your Library Repository and its Reference Libraries.

An *index.txt* file is automatically generated by AtomWeaver to cache Library information. You can manually delete a Reference Library, but if you do, you should also remove the *index.txt* file. Then run the *Library - Update Repository Index* command to rebuild the index file.

The Projects Folder

By default, this folder will hold all your projects. Each folder will have its own folder too.

You can, however, choose another folder when starting a new project. If you move your project to another folder, you should then check your project's definitions by running the *Project - Settings* menu command.
