



Tigerstripe 0.4.5 – Release Notes

01/23/2009

Overview	1
To install/download	2
0.4.5 Maintenance Release	2
<i>Smarter Project Auditing</i>	<i>2</i>
<i>Tigerstripe Plugins as .jar</i>	<i>2</i>
<i>Upgraded baseline 3rd party plugins to latest</i>	<i>2</i>
<i>Tigerstripe Update Site Manager fixed</i>	<i>3</i>
0.4.x Stream	3
<i>Buckminster Support</i>	<i>3</i>
<i>UML2 Import Enhancements</i>	<i>3</i>
<i>Annotations-based Icons/Decorations</i>	<i>3</i>
<i>Custom Audit Rules</i>	<i>4</i>
<i>Custom Default Naming Rules</i>	<i>4</i>
<i>Show Stereotypes in Tigerstripe Explorer</i>	<i>4</i>
<i>Package Artifacts</i>	<i>4</i>
<i>Custom Artifact Patterns</i>	<i>4</i>
<i>New icons</i>	<i>5</i>
<i>Annotations and facets</i>	<i>5</i>
Known Issues	5

Overview

This 5th maintenance release for the 0.4.x stream of Tigerstripe brings a number of additional fixes to prior 0.4.x releases.

The key change in 0.4 (vs 0.3.x) is the ability to customize Tigerstripe through a set of extension points allowing to control:

- The palette of modeling artifacts: this allows to tailor Tigerstripe for a more specific modeling domain.
- The audit rules validating the model in real-time. Along with changing the artifacts used in the model, it is possible to add additional validation rules to further guide the modeling process
- The Auto-naming rules proposed to users as they create new model elements.
- How UML2 models are mapped into the Tigerstripe metamodel.

These changes make it possible to build various flavors of Tigerstripe that are tailored to a specific domain.

To install/download

Note: the update site has been fixed (corrupted descriptor) so an upgrade through the site at <http://download.eclipse.org/technology/tigerstripe/updates/site.xml> is finally available.

To Install Tigerstripe, you will need Eclipse 3.4.x and the latest versions of the following 3rd party plugins in addition to the Tigerstripe plugins. The exact versions are as follows:

- emf-runtime-2.4.1.zip
- sdo-runtime-2.4.1.zip
- emf-transaction-runtime-1.2.2.zip
- emf-query-runtime-1.2.0.zip
- emf-validation-runtime-1.2.1.zip
- mdt-ocl-runtime-1.2.2.zip
- mdt-uml2-runtime-2.2.1.zip
- GEF-ALL-3.4.1.zip
- gmf-runtime-2.1.2.zip
- orbit-R20080807152315.zip

This can be downloaded as a single .zip file along with the Tigerstripe plugins at this URL from the Tigerstripe download page at:

http://www.eclipse.org/downloads/download.php?file=/technology/tigerstripe/bundle-tigerstripe-workbench-0.4.5_incubation.zip

0.4.5 Maintenance Release

Smarter Project Auditing

This release includes an update to how Tigerstripe models are being audited. It used to rebuild models quite frequently lead to a lot of CPU activity on large models. Starting with 0.4.5, the project auditor maintains a set of indexes to minimize what is being audited each time a model is changed. This results in more optimized, less frequent rebuilds.

Tigerstripe Plugins as .jar

It used to be that one of the Tigerstripe plugins was installed “unjar-ed” leading to several installation problems in the past. With this release, all plugins are installed as .jar OSGi bundles.

Upgraded baseline 3rd party plugins to latest

Tigerstripe is built on some of the major Eclipse projects, such as EMF, GMF, GEF. This release has been tested and updated with the latest versions of these packages. In particular, previous releases of 0.4.x had issues with Drag-n-Drop onto diagrams when running with the latest EMF, GMF or GEF as opposed to recommended versions.

The exact versions used for testing is as follows:

- emf-runtime-2.4.1.zip
- sdo-runtime-2.4.1.zip
- emf-transaction-runtime-1.2.2.zip
- emf-query-runtime-1.2.0.zip
- emf-validation-runtime-1.2.1.zip
- mdt-ocl-runtime-1.2.2.zip
- mdt-uml2-runtime-2.2.1.zip
- GEF-ALL-3.4.1.zip
- gmf-runtime-2.1.2.zip
- orbit-R20080807152315.zip

Tigerstripe Update Site Manager fixed

The Update Site descriptor for Tigerstripe has been cleaned up from a corrupted file that was preventing the installation.

0.4.x Stream

Buckminster Support

Beginning with Tigerstripe 0.4.x, we now provide integration with Buckminster allowing to use buckminster to materialize Tigerstripe Model projects and Generator projects automatically. Referenced projects are resolved by Buckminster based on the configuration of your projects (tigerstripe.xml). This feature makes it very easy to maintain large models split over multiple cross-referenced projects.

UML2 Import Enhancements

With this release, the UML2 import feature of Tigerstripe was mostly re-written to provide additional robustness and enhanced flexibility. Because the intent of Tigerstripe models is code generation, Tigerstripe enforces tighter models than other typical modeling environments. As a result, during import, when a model is missing an attribute type, an association end name, etc... Tigerstripe will now fill in the details to allow for a seamless import.

In addition, all steps of mapping of a UML2 model into a Tigerstripe model can be further controlled through Eclipse extension points. These allow developers to inject their own mapping rules, and behavior into the import logic.

Annotations-based Icons/Decorations

Tigerstripe Annotations were introduced with version 0.3 of Tigerstripe as a mean to decorate a model with arbitrary complex data, yet storing these decoration separately from the model. This mechanism allows to add separation of concerns between the model itself, and the way it is used by multiple consumers.

Beginning with 0.4, we have extended Tigerstripe to present extension points that allow to control the look-n-feel of the modeling environment based on annotations.

For example, Artifacts decorated with a specific annotation may appear with a different Icon or label, giving some additional visual hints to the modeler. Combining this feature with the “Custom Artifact Pattern Feature” (see below), it becomes possible to completely re-define Tigerstripe to present a domain specific modeling palette, with no impact on the underlying metamodel, thus enabling easier model-to-model mappings.

Custom Audit Rules

Tigerstripe can now be further customized to include additional model audit rules. Here again, this is available through an Extension Point. These rules are triggered in real-time as the model is being edited and may present “problems” in the Problems view.

Custom Default Naming Rules

Naming conventions are quite important in large models. Starting with version 0.4 Tigerstripe allows to implement new logic to provide default names upon creation of new model elements. This can not only simplify the modeler’s task but facilitate the adoption of naming conventions.

Show Stereotypes in Tigerstripe Explorer

Through a set of checkboxes in the Preferences/Tigerstripe/Explorer configuration, it is now possible to configure Tigerstripe to display stereotypes in the Explorer.

This option is off by default.

Package Artifacts

Up until version 0.3, Tigerstripe wasn’t considering “Packages” as true model artifacts. Packages were simply containers for artifacts. With 0.4, packages become model artifacts, allowing to add stereotypes, annotations on them.

Note: Package artifacts are not supported on Class Diagrams at this stage, but will be later in the future.

Custom Artifact Patterns

While modeling , you often perform the same actions when creating new model elements: create, name, add base stereotype/annotation, etc...

This can now be captured as a single click by programming the action being icons in the modeling palette to be an Artifact Pattern. Patterns may create/set anything regarding a new model element, whether used for nodes (Managed Entity, e.g.) or edge (association, e.g.).

At this time, the set up of artifact patterns within the palette is done only programmatically through a set of extensions points. The created plugins need to be installed in the target Tigerstripe install. We are planning to provide an SDK to quickly develop patterns and integrate them into a Tigerstripe model palette.

New icons

We went through a re-skinning exercise of the UI to include new icons for model elements..

The new icons are supposedly more readable (different colors and shapes) and prettier ☺... Feedback welcome.

Annotations and facets

Facets have been extended to include Annotations in two ways:

- Use of annotations to include/exclude model elements in the initial facet scope.
- Select a set of annotation namespaces as part of the facet. This allows generators to restrict their behavior to the select annotations only, and have that stored as part of a facet. In other words, facets can control the scope of the sub-model, but also the scope of the annotation namespaces.

Known Issues

There are no major known issues. Please refer to [Bugzilla](#) for a list of open bugs.

We are working on a more robust support for refactoring actions on Tigerstripe models.