

KIELER Pragmatics Release 0.11.0

KIELER Pragmatics 0.11.0

- [Official Project Website](#)
- Built on Eclipse 4.3 "Kepler"

We are happy to provide release 0.11.0 of the **KIELER Pragmatics Project**! It is a framework for enhanced user interaction in graphical modeling that builds on the Eclipse platform. This document describes how to install the release and highlights the most exciting new features.

As a bonus, we are co-releasing our KLightD-based **KIELER Ptolemy Browser**. The **KIELER Web Service for Layout (KWebS)** will be updated to the new release soon as well.

- [Installation](#)
- [Release Notes](#)
 - [API Changes in 0.11.0](#)
 - [New Features Included in 0.11.0](#)
- [Quickstart](#)
 - [Diagram Layout](#)
 - [Viewing Ptolemy Models](#)
- [Known Problems and Limitations](#)
- [Bug Reports, Comments](#)

Installation

KIELER Pragmatics 0.11.0 is provided as a set of features, available to be added to your Eclipse installation via an update site. KIELER Pragmatics 0.11.0 was tested on [Eclipse 4.3 "Kepler"](#), but may also be compatible with other versions of Eclipse.

- See the [KIELER Downloads](#) site for download details.
- KIELER Pragmatics requires an installed Java Runtime Environment \geq version 1.5.
- The layout components of KIELER can make use of the layout algorithms provided by the [GraphViz](#) library. You may want to install that.

See the [KIELER Tools and Libraries](#) page to download the KIELER Ptolemy Browser and the KIELER Web Service.

Release Notes

The 0.11.0 release of KIELER Pragmatics is mainly an effort to redesign our core APIs, KIML and KLightD. **It includes a number of API-breaking changes, so beware of upgrading to this release if you currently do not have time to adapt your code accordingly.** You can find a list of solved tickets for this release at [our issue tracking system](#). Release notes of older releases can be found at our [Release Notes](#) page.

If you're wondering where components have gone that were traditionally a part of KIELER, you might not know yet that the KIELER was split into two separate projects, KIELER Pragmatics and KIELER Semantics, which release pretty much independently.

API Changes in 0.11.0

This is a summary of the API changes introduced with version 0.11.0:

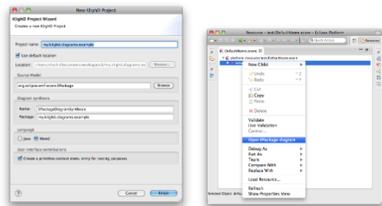
- KIELER Light-Weight Diagrams
 - `IViewer<?>` now provides the method `getViewContext()` as the view contexts contain the source/diagram elements mapping information.
 - `IViewer`'s method `getModel()` is marked deprecated and will be removed soon.
 - `ContextViewer`'s method `getCurrentViewContext()` is deprecated and will be removed soon.
 - The dedicated internal interface `ILayoutRecorder` now provides `startRecording()` and `stopRecording(...)`. These methods are not supposed to be called by application code.
 - Diagram outline pages must now implement `IDiagramOutlinePage` (internal as well), viewer implementation providing outline pages must implement the related `Provider` interface.
The base plug-in `de.cau.cs.kieler.klighd` does not provide and support for [KIVI \(KIELER Infrastructure for Viewmanagement\)](#) any more;
 - include the new plug-in `de.cau.cs.kieler.klighd.kivi` in order to use the established effects.
 - The `DiagramViewPart`, `DiagramEditorPart`, `DiagramViewManager` implementations moved to `de.cau.cs.kieler.klighd.ui`, which has been introduced with the last release.
 - The `ViewContext`, which is a record of all data required for building up a diagram, now provides a reference to the workbench part the diagram is shown in (was available via `ContextViewer` as yet).
 - `IDiagramWorkbenchPart`'s `getContextViewer()` has been marked deprecated, simply use `getViewer()` (in case really need to access the `ContextViewer` call `getViewer().getContextViewer()`)
- KIELER Infrastructure for Meta-Layout
 - A new base feature for KIML was created, containing only the API, but no UI (use this feature if you don't want the KIELER UI contributions in your application)
 - A generic type check was added to the method `setProperty(...)` in `IPropertyHolder`
 - `DiagramLayoutEngine` moved from `kiml.ui` plugin to `kiml.service` plugin

- Added layout options for animation, zoom-to-fit, and progress bar display
- Modified `DiagramLayoutEngine` methods to use these new options
- `IDiagramLayoutManager` and `LayoutMapping` moved from `kiml.ui` plugin to `kiml.service` plugin
- `LayoutDataService` implementation moved from `kiml.ui` plugin to `kiml.service` plugin
- `LayoutInfoService` implementation moved from `kiml.ui` plugin to `kiml.service` plugin and renamed to `LayoutConfigService`
- Extension point `layoutInfo` renamed to `layoutConfigs`, `element` option renamed to `staticConfig`, `element` semanticOption renamed to `semanticConfig`, `element` config renamed to `customConfig`
- Created base class for `LayoutConfigService` in `kiml` base plugin
- Layout configurators now reference domain model elements as `Object` instead of `EObject`
- `AnalysisService` moved from `kiml.service` plugin to `kiml.grana` plugin
- `TransformationService` moved from `kiml.service` plugin to `kiml.formats` plugin and renamed to `GraphFormatsService`
- Extension point `layoutManagers` moved from `kiml.ui` plugin to `kiml.service` plugin
- Removed support for `IGraphLayoutEngine` implementations from `layoutManagers` extension point
- Created new class `LayoutManagersService` in `kiml.service` from code that was previously hidden in `kiml.ui`
- Removed `KWebS` client for Eclipse (`kwebs.client`) and removed special code for that client from `KIML`
- Moved content of `KWebS` base plugin to `kwebs.server`
- Using `IFactory` interface for lazy creation of `KIML` service classes
- Moved `LayoutContext` class to `kiml.config`
- Modified `ILayoutConfig` methods: `getAffectedOptions(..)` instead of `transferValues(..)`

New Features Included in 0.11.0

Here's a few highlights of what's new in version 0.11.0:

- **KIELER Light-Weight Diagrams**
 - In order to support early prototyping further, the `KLighD` Project Wizard (introduced with the 0.10.0 release) registers simple menu contributions if corresponding item is checked. Besides, the compiler settings of the created projects are locally set to Java 1.5 compliance level in order to avoid conflicts in combination with Java 1.7 runtime environments.



- `KLighD` diagrams may be clipped in order to reduce depicted content.



- In diagram synthesis implementations colors may now be set by means of the `Colors` enumeration in `de.cau.cs.kieler.core.krendering` package and corresponding methods in the `KColoring` classes.
- The selection highlighting of diagram elements may now be specified by means of selection-specific `KStyles` with the related `selectio` flag set to `true`. Find also related helper methods in the `KRenderingExtensions`. The sequence of a `KInvisibility` style with `in visible = true` (the initial value) and a selection-specific `KInvisibility` with `invisible = false` allows to uncover additional figure components in case of selection 😊
- **KIELER Layout Web Services (KWebS)**
 - We added a 'Live' section to the supporting server where you can play around with different graphs and options.
 - A new format (`JSON`) was introduced.
 - Besides the SOAP-based web service we know also support `HTTP-Post` requests, e.g. allowing a lightweight use from JavaScript.

Quickstart

Diagram Layout

One of the core features of `KIELER` Pragmatics is the `KIELER` Infrastructure for Meta-Layout (`KIML`). It provides the glue between diagram editors and layout algorithms. If you have installed layout support for `GMF` editors, you may want to try the following:

- Open a diagram using a `GMF` editor.
- Press the `Layout` button in the toolbar: 
- Open the `Layout View` by clicking `Window -> Show View -> Other...` and choosing `Layout` from the `KIELER Layout` category.
- Change some of the values and press the `Layout` button again.

Viewing Ptolemy Models

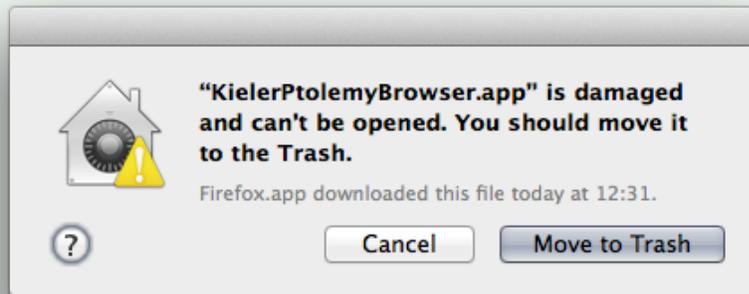
If you have a `Ptolemy` installation, try the following:

- Save a `Ptolemy` model as a `.moml` file somewhere.

- Download and start our [Ptolemy Model Browser](#).

✔ Note for Mac OS Users

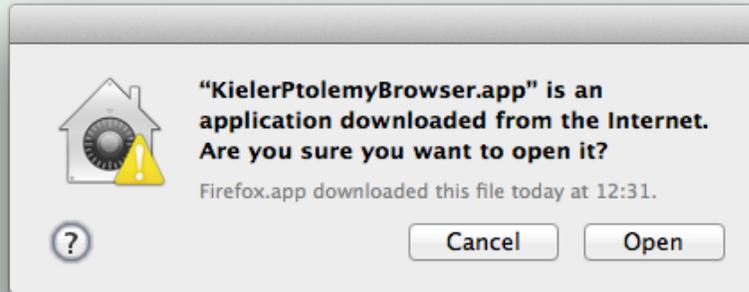
Starting with Mac OS X Mountain Lion, trying to start the Ptolemy Model Browser may give you the following error message:



The problem is not that our application file is damaged in any way. Much rather, we have not invested the money to obtain a proper developer certificate with which this error would not be generated. The solution is to open your system preferences and navigate to the *Security & Privacy* settings. Therein, make sure to set *Allow apps downloaded from* to *Anywhere*:



Start the Ptolemy Browser again. This will result in the following warning:



Simple click "Open" and you're good to go at last.

- Open your moml file.
- Double-click actors that have further models inside them. Also double-click modal model states that have refinements. Use the sidebar on the right to influence how your model is displayed.

Known Problems and Limitations

- The *Layout Hierarchy* option of KLayout Layered does not work with center labels of edges that cross hierarchy boundaries. There may also be further problems.
- The Ptolemy Model Viewer can properly display only a limited set of actors. Other actors may look a lot different than they do in Ptolemy.
- Currently there is no full-blown user documentation integrated. See our [KIELER Project](#) for more information.

Bug Reports, Comments

We're working hard to make the graphical modeling user experience as convenient as possible. However, bugs can still remain in the code and some things might not be as you would expect them. Please don't hesitate to send in bug reports or give other comments like feature requests.

Send bug reports to kieler@..., please. For news or general questions subscribe to the [rt-kieler mailing list](#).