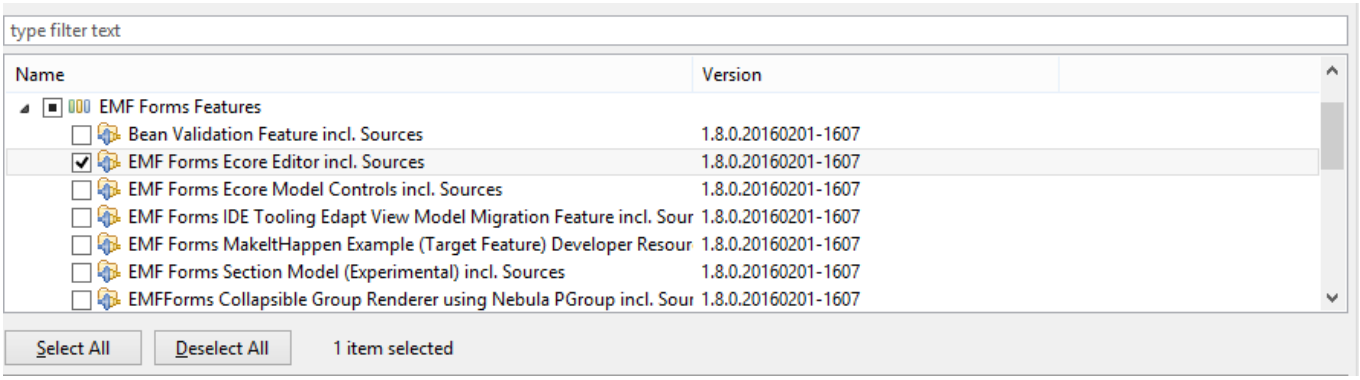
EMF Forms Editors

EMF Forms provides three ready-to-use editors, which can be integrated into any kind of Eclipse IDE or RCP application:

- The Generic Editor: A fully functional editor for the creation and modification any kind of custom model instance based on your custom model
- The Ecore Editor: Allows you to create and modify Ecore models
- The Genmodel Editor: Allows you to create and modify generator model

Please see the following sections to learn about the editors in more detail.

All editors are available starting from release 1.8.0. In 1.8.0 they are not yet part of the SDK itself, therefore, they are not yet available in the standard Eclipse Modeling Edition. This is planned for the Neon release. You can download and install them from our update site (1.8.0 or higher), the feature “EMF Forms Ecore Editor” contains all three editors.

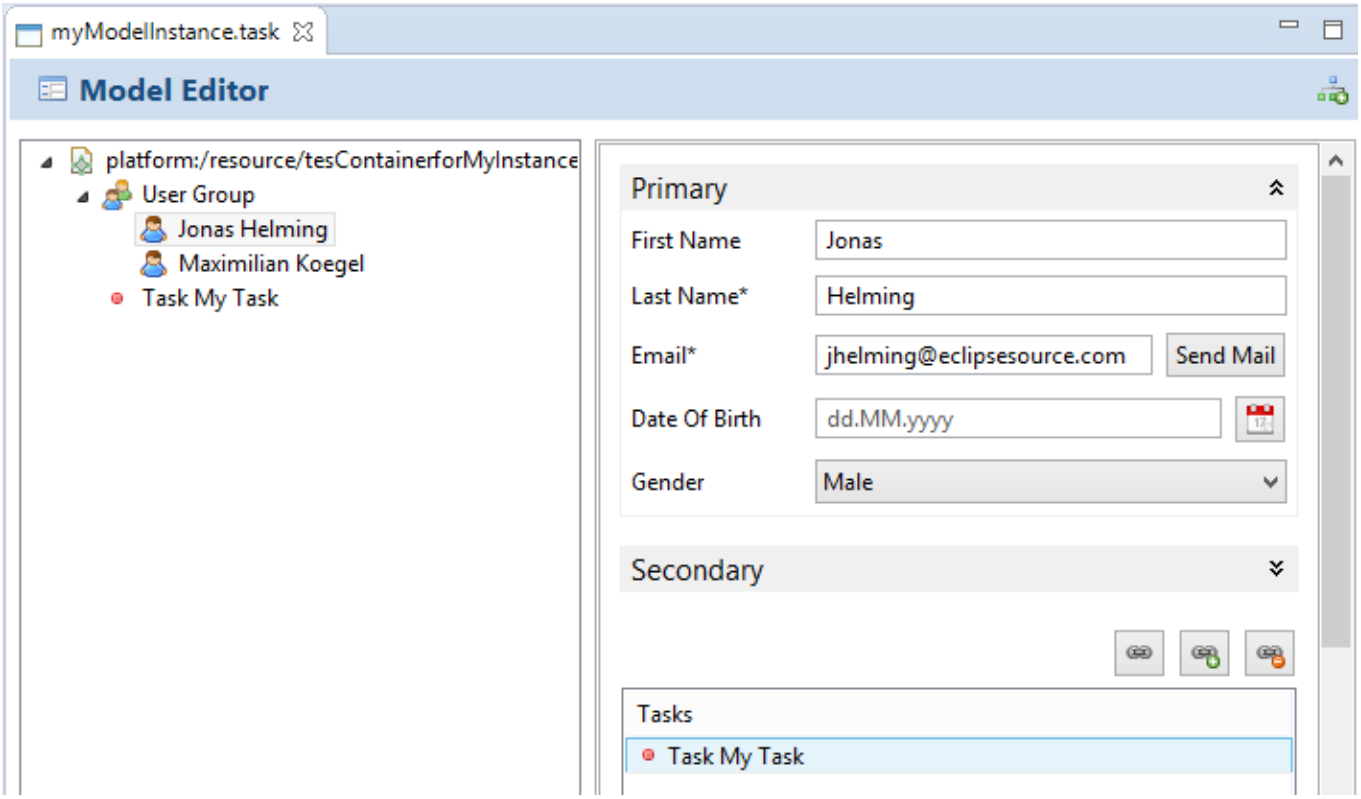


The editors are currently under active development, please provide feedback by submitting bugs or feature requests or contact us if you are interested in enhancements or support.

Generic Editor

EMF Forms is often used to build editors for EMF model instances. Therefore, the forms rendered by EMF Forms are embedded into an Eclipse editor, which loads and saves the data from files and integrates it into an Eclipse workbench. Such editors also typically provide features such as DnD, undo/redo, and context menus. The “generated editor” of EMF provides a code template on how to implement this kind of editor, but it is not really flexible and extensible.

However, as the implementation of such editors is typically pretty generic, EMF Forms provides a ready-to-use implementation of a fully generic editor. It includes a tree viewer on the left site, showing the hierarchy of a model instance, allowing you to move elements and create new ones. On the right site, it shows a detail pane including all properties of the element selected in the tree.



This generic editor can be used for any EMF model. It will provide all features such as redo/undo or DnD from scratch, without any modification. Compared to the “generated editor” it is has been built to be adaptable and extensible and is therefore a good starting point for the implementation of your custom editor.

Setup

The EMF Forms editor is fully generic. Therefore no code generation is required. You only need to register the editor to the custom file extension that you want to use the editor for. The following listing shows the registration for EMF model instances with a file ending “.task”. Please note, that the EMF Forms Genmodel editor can generate such an extension for you (see XXX), if you use the default generation, you will need to manually create it.

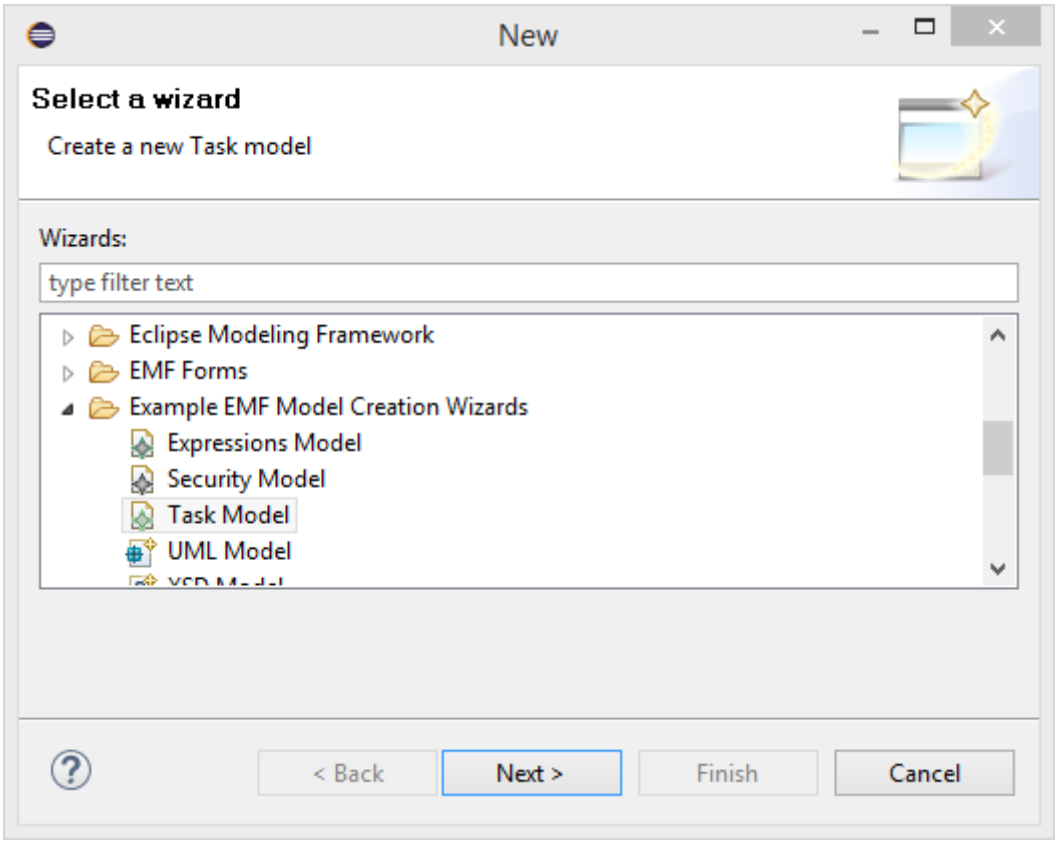
```
<extension point="org.eclipse.ui.editors">

    <editor class="org.eclipse.emfforms.spi.editor.GenericEditor" default="true"
extensions="task" icon="pathToYourIcon" id="yourID" name="Your Editor Name"/>

   </extension>
```

If you now make the bundle containing this registration part of your run/product config, you can open a file with the ".task" extension in the generic editor. Please note, that this file is expected to contain an EMF model instance.

Currently, the generic editor does not yet support the creation of such files. You can either write a custom wizard to do that or use the default wizard generated by EMF. To use the default wizard, generate the ".editor" bundle for you EMF model. It will generate a "New Wizard" for your model. If you now include the ".editor" bundle in your run/product configuration, you will see a corresponding entry in the default "New Dialog" under the category "Example EMF Model Creation Wizards". As the creation of your custom model instance probably needs to be adapted, you can change this wizard according to your needs. Please have a look at the class "XXXModelWizard" in the generated ".editor" bundle, where "XXX" is the name of your model. Also have a look at the plugin.xml of the editor bundle to change the location and name of the wizard registration. You can adapt both to meet your custom requirements.

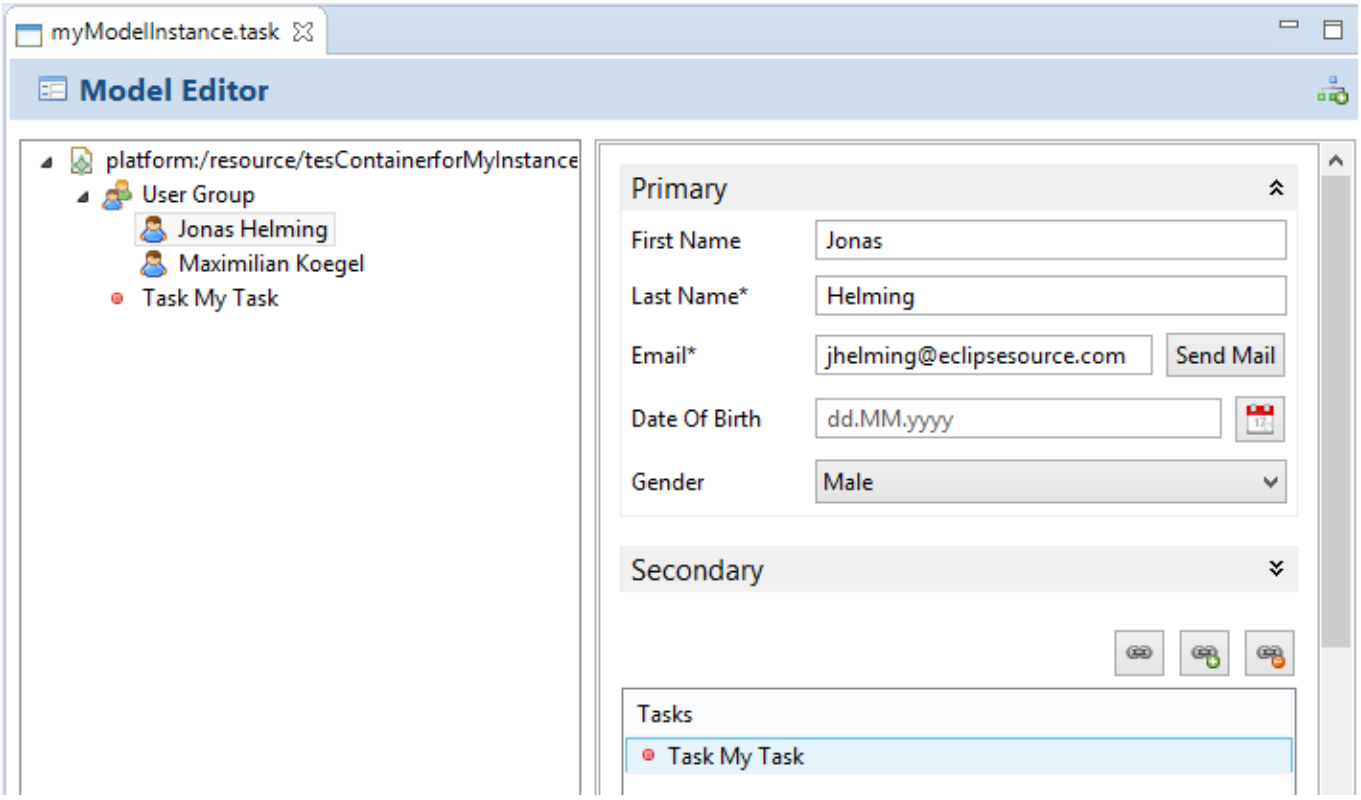


Please note, that if you use the generated ".editor" bundle as a "new wizard", it also contains the generated editor. If you add the registration for the generic editor there will be two editors registered for the same file extension. The "Open With…" dialog in Eclipse will allow you to select the right one. Alternatively, you can remove the registration of the generated editor in the plugin.xml.

Usage

Once you have registered the generic editor, you can open XML files containing model instance of your custom EMF model. It will present you a tree view, showing the file as a root node and

all contained model elements as sub elements. In the tree view, you can create new elements along the containment hierarchy of your model by clicking the parent. You can move elements around via Drag and Drop and use the default undo/redo actions. The generic editor is also capable of loading multiple files, e.g. to create references between them. Click the button in the top right corner to select a second model resource to be loaded. It will then be shown as a second root node in the tree.



If you select an element in the tree, its properties (attributes and references) will be shown on the right site. The details view is created and rendered on the fly by EMF Forms. In the default version, it will show a control for every attribute and reference in a flat list. You can use the full power of EMF Forms to adapt this details view, changing the layout and adding your own custom controls to it. See the following section for adaptations possibilities.

Adaptation and Extension

The generic editor is fully generic. That means, it does not need to be initially adapted to show and work with any kind of model instance. However, if you want to build up your own editor based on it, sooner or later you might want to adapt the look and feel as well as the features.

The editor as well as EMF Forms is designed for this purpose. The first good news is that the details view, to the right of editor uses EMF Forms for rendering. So you define view models to adapt the layout, you can add your own renderers and adapt existing ones. Please have a look at our EMF Forms documentation to learn more about this. Please note, that registering view models or renderers will only affect the details view where EMF Forms is embedded, the tree is created programmatically in the generic editor.

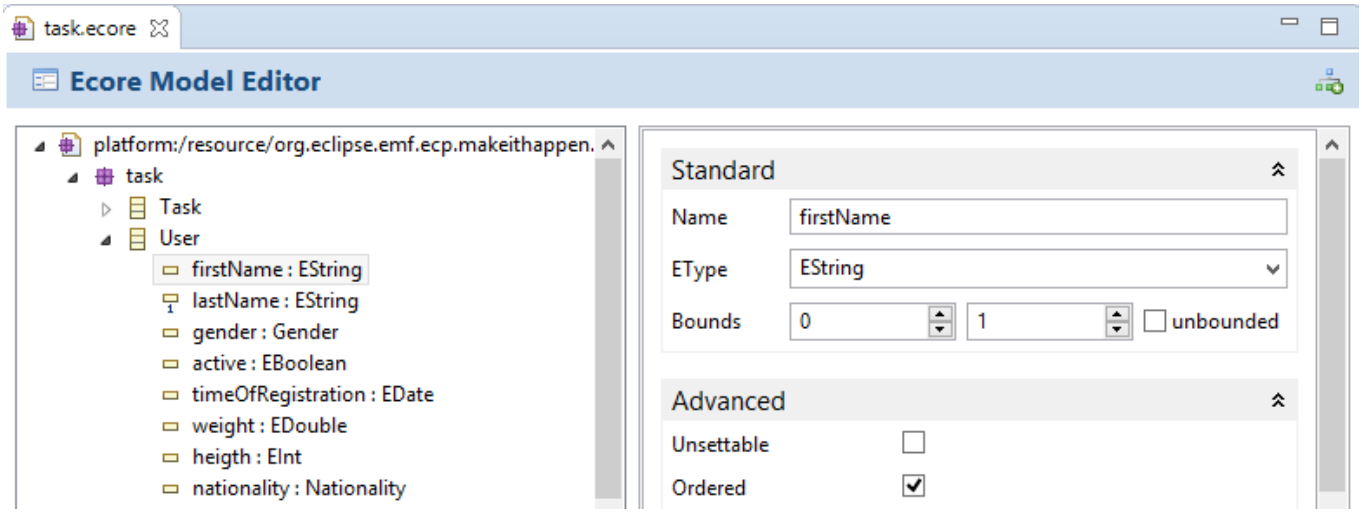
To adapt the tree the headline or the general editor behavior, you need to programmatically extend or replace the generic editor. A good start is to create a subclass of org.eclipse.emfforms.spi.editor.GenericEditor, override respective methods, and register your subclass instead of the generic editor itself.

The generic editor is currently under active development, if you miss any feature or ways to adapt it, please provide feedback by submitting bugs or feature requests or contact us if you are interested in enhancements or support.

Ecore Editor and Genmodel Editor

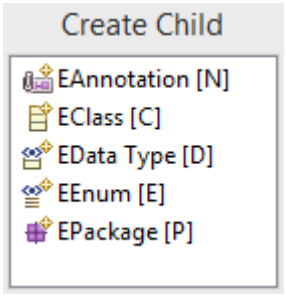
The Ecore editor and the Genmodel editor are subclasses of the EMF Forms generic editor described before. We customized the generic editor specifically for creating and modifying Ecore models and Genmodels. Those two editors are ready to be used as development tools for your modeling project. Therefore, you will need to install them to your IDE first.

Both editors are already registered for the extension ".ecore" and ".genmodel". However, they are not currently registered as default, this is still the traditional editor. Therefore, you need to right click your files, select "open with" and select the new editor ("Ecore Editor" and "Genmodel Editor").



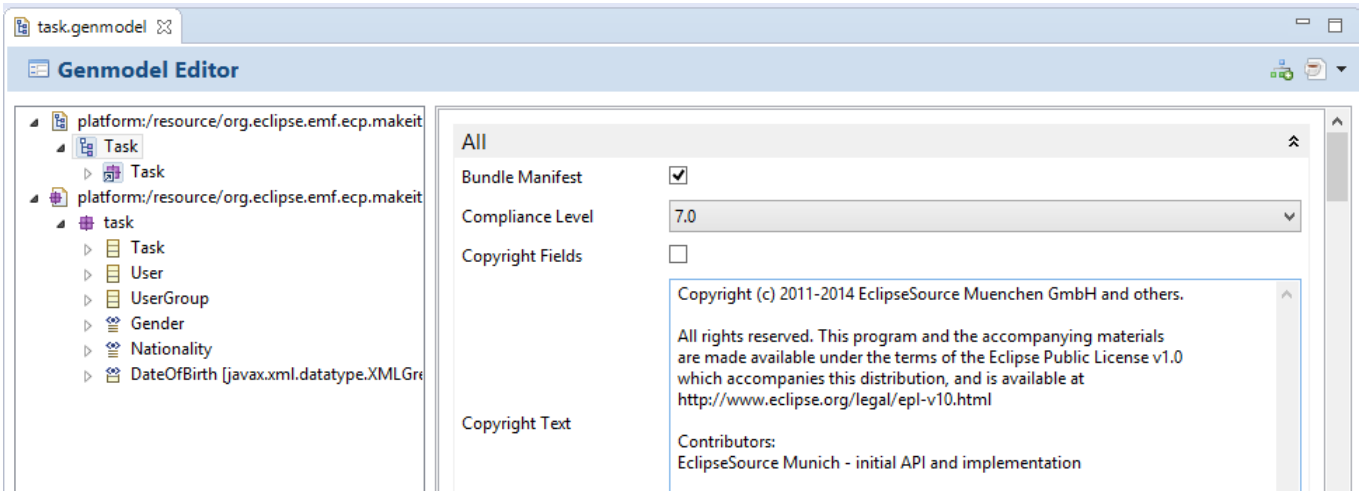
Both editors show the content of the file on the left side as a tree (as before). As you can see, in both editors we got rid of the properties view and combined the tree and the properties into one editor. We have ordered the properties by importance (instead of the former alphabetical order), grouped them and combined some fields, such as lower and upper bounds. Where it made sense, we have also added custom controls, e.g. to support auto-completion.

For the ecore editor, we added short-cuts and dialogs for the creation of new elements. This enables you to specify a model without leaving the keyboard and improves efficiency. To create a new element in the Ecore editor, select its parent and press "CTRL+N". The opened dialog shows you, which key to press, e.g. "C" for a new class.



This will open up another dialog, which lets you enter the properties of the new class extremely fast. As the generic editor, both editors also allow you to load multiple files (e.g. multiple Ecores). To do so, click the button in the top right corner of the editors.

The Genmodel editor additionally has a toolbar item to generate code from it. It provides the options to generate the model, the edit, the editor, and the test bundle, but also to generate the model and edit bundle at the same time, which is a typical use case in many projects.



All editors are available starting from release 1.8.0. In 1.8.0 they are not yet part of the SDK itself, therefore, they are not yet available in the standard Eclipse Modeling Edition. This is planned for the Neon release. You can download and install them from our update site (1.8.0 or higher), the feature “EMF Forms Ecore Editor” contains all three editors. The editors are currently under active development, if you miss any feature or ways to adapt it, please provide feedback by submitting bugs or feature requests or contact us if you are interested in enhancements or support.

Share it

You may also like...

- **EMF Forms 1.9.0 Feature: Ecore Editor Reloaded … again**
- **EMF Forms 1.8.0 Feature: Factories for TreeViewer and TableViewer**

Author:
Jonas Helming
Updated:
Feb 11th, 2016

Need help

Get Training

Get Developer Support

search

Terms of Use

Contact Us