

Getting Started with the Layout Painter

This chapter covers the following topics:

- General Information
 - Creating a Layout
 - Layout Templates
 - Opening an Existing Layout
 - Elements of the Layout Painter
-

General Information

Ajax Developer follows the commonly accepted paradigm of separating the layout (view) as much as possible from the application logic (model). You design the layout using the WYSIWYG Layout Painter. The result is kept in an XML page layout file. When you save the XML page layout, an HTML page is generated.

On the one hand, the XML layout contains the used controls of a page; on the other hand, it keeps information about how these controls are bound to the application code.

If you implement your application in Natural, the controls are bound to a Natural adapter. The Natural adapter is generated as soon as you save the page layout. The Natural adapter becomes part of the Natural application and forms the interface between the Natural application code and the user interface.

If you implement in Java, the controls are bound to the properties and methods of a Java adapter. This is a Java class that you can generate with the Code Assistant (which is part of the Layout Painter).

In addition, the XML layout may optionally contain so called "literals". Literals are text identifiers (text IDs) that are replaced at runtime with language-specific texts. For each language you want to support, there must be a comma-separated value (CSV) file that contains the language-specific text per literal. You maintain these translation files using the Literal Assistant (which is part of the Layout Painter).

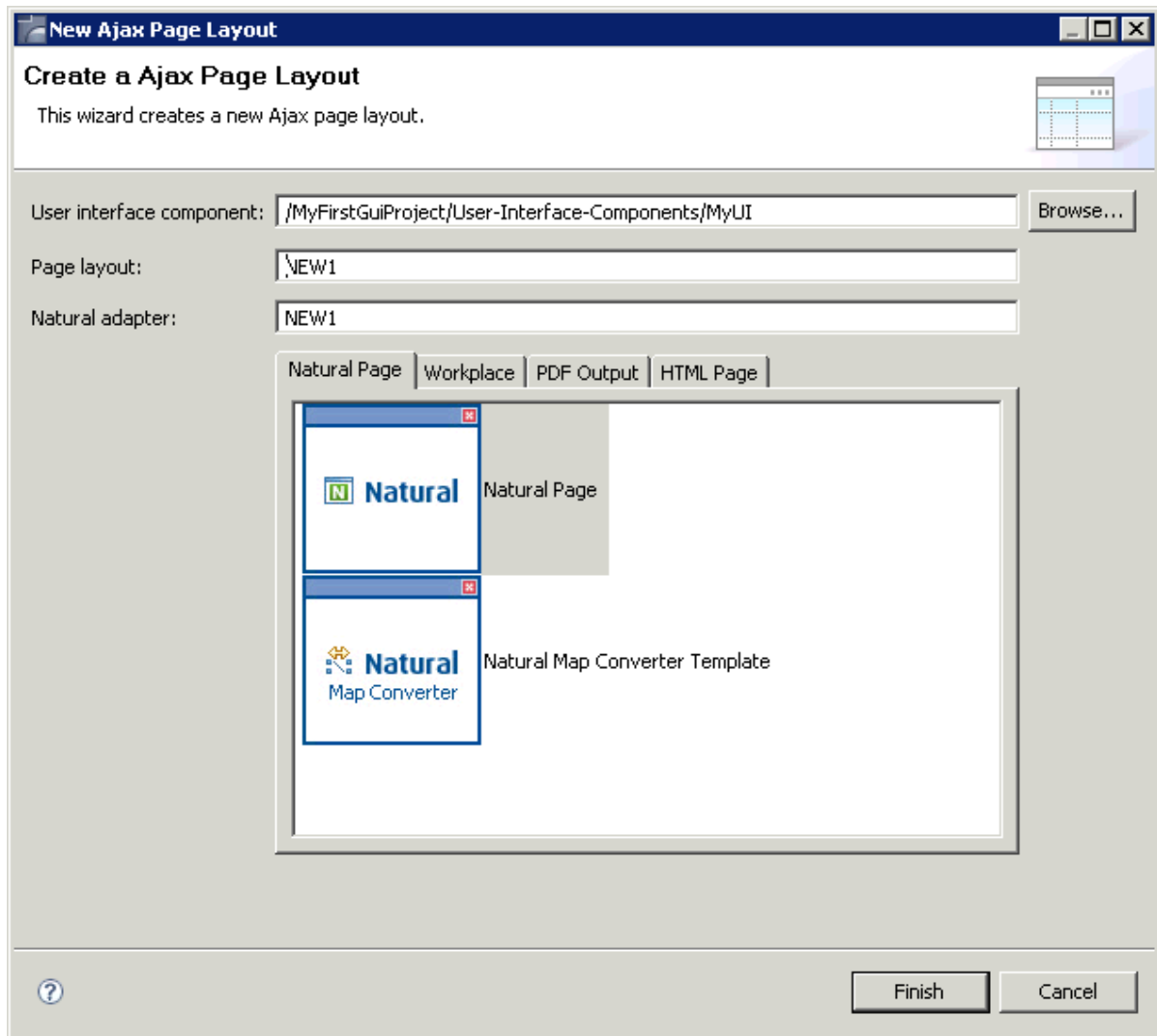
Creating a Layout

The layouts that you create as described below are stored in the *xml* folder of the user interface component.

To create a layout

1. In the **Navigator** view, select the user interface component or the *xml* folder.
2. From the **File** menu, choose **New > Page Layout**.

The following dialog box appears, providing several tabs from which you can select a layout template.



3. In the **Layout page** text box, enter the name of the file that is to contain your layout definition. Do not specify an extension; the extension ".xml" will automatically be added.

The **Natural adapter** text box is only visible when the **Natural Page** tab is currently selected. Here, you enter the name of the Natural adapter that will be generated when you save the page layout. This name must match the naming conventions for Natural objects.

4. Select the layout template for your page. See also *Layout Templates*.
5. Choose the **Finish** button.

The Layout Painter appears (see below).

Layout Templates

When you create a layout, you have to select a layout template. The following types of layout templates are available:

- **Natural Page**
These templates are used to create pages for Natural for Ajax. They have the NATPAGE control as the top node. See the NaturalONE documentation for further information.
- **Workplace**
These templates are used for arranging Application Designer pages in a frameset. They have the MFPAGE control as the top node. See *Multi Frame Pages* in *Working with Pages* for further information.
- **PDF Output**
This template is used for integrating a PDF document into an Application Designer page. It has the CISFO:FOPPAGE2 control as the top node. See *PDF and FOP Services* for further information.
- **HTML Page**
Different templates are available which have the PAGE control as the top node. See *Typical Page Layout* in *Layout Elements* for further information.

Opening an Existing Layout

Layouts are stored in the *xml* folder of the user interface component. They have the extension *.xml*.

To open a layout

- In the **Navigator** view, double-click the layout that you want to open.

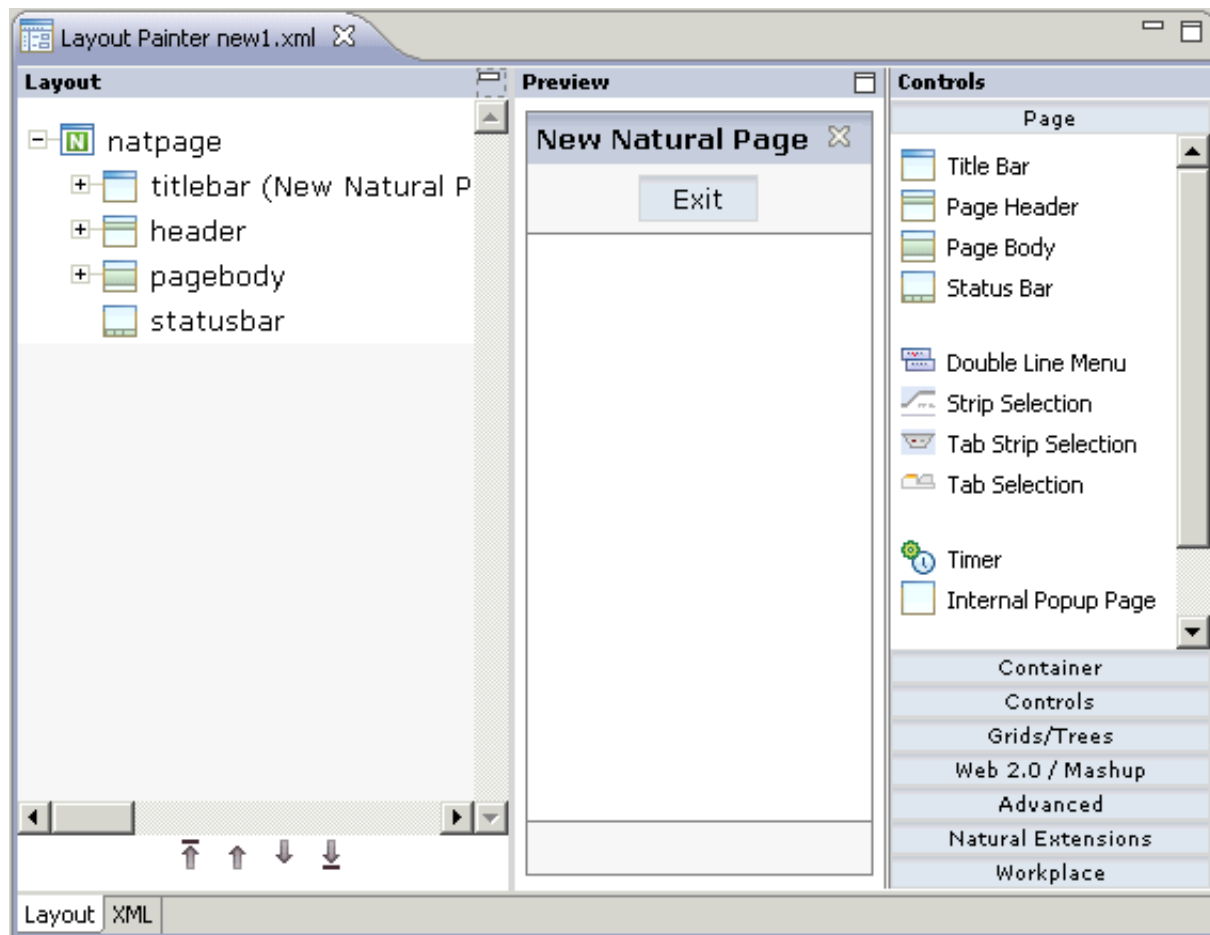
Or:

Invoke the context menu for the layout and choose **Open With > Layout Painter**

The Layout Painter appears (see below).

Elements of the Layout Painter

The Layout Painter appears when you create a new layout or when you open an existing layout.



The Layout Painter is divided into several areas:

- **Layout Area (left side)**

This area shows the layout tree which contains the controls that represent the XML layout definition. You drag these controls from the controls palette into the layout tree. Each node in the layout tree represents an XML tag.

Using the button which is located to the right of the title bar, you can hide the layout area. This enlarges the preview area. To redisplay the layout tree, choose the small arrow which is shown in the left of the preview area's title bar.

- **Preview Area (middle)**

The preview area shows the page which is created using the controls in the layout area. This page is refreshed each time, you choose the **Refresh Preview** command (see below).

Using the toggle button which is located to the right of the preview area's title bar, you can enlarge the preview area so that it takes up the space of the layout area and controls palette, and reduce the enlarged preview area so that the layout area and controls palette are shown again.

- **Controls Palette (right side)**

Each control is represented by an icon. A tool tip is provided which appears when you move the mouse pointer over the control. This tool tip also displays the XML tag which will be used in the XML layout.

The controls palette is structured into sections (for example, **Page** and **Container**), where each section represents certain types of controls.

Note:

The sections that are shown in the controls palette and the content of a section depend on the layout template that is used for the current layout.

When you select a control in the layout tree or in the preview area, the properties for this control are shown in the **Properties** view. You can modify the properties as required.

When the Layout Painter is active, the additional menu **Ajax Developer** is shown in the menu bar. Using this menu, you can invoke additional tools which are helpful for the creation of layouts:

- **Code Assistant**
Only used for developing page layouts in Java. Helps you to generate code for all properties and methods which are explicitly referenced in the page layout. See *Code Assistant* for further information.
- **Literal Assistant**
Helps you to prepare your layout for multiple languages. You can maintain texts of different languages for the literals that are referenced within the XML layout. See *Literal Assistant* for further information.
- **Validation Rules Editor**
You can define that the value that is entered in a field must meet certain conditions. See *Using the Validation Rules Editor* for further information.
- **Formula Editor**
You can define a formula for a field (for example, the sum of the fields A and B is to be shown in field C). See *Using the Formula Editor* for further information.

Some commands of the **Ajax Developer** menu (such as **Refresh Preview**) are also available as buttons in a toolbar.

