Changes in Magnolia 6 UI

Related topics:
- UI module
- Release notes for Magnolia CMS 6.2

This page covers the changes in Magnolia 6 UI as accumulated over Magnolia releases 6.0, 6.1.x and 6.2.x.

- Summary
- Why we have changed things
- What is preserved
  - Class names
  - Magnolia module names
- What has changed
  - Look and feel
  - Maven modules
  - Features
- Magnolia 6 and Magnolia 5 UI frameworks
- Backward compatibility
  - Runtime compatibility
  - Compile-time compatibility
- How to upgrade
  - With Maven
  - Manually
- Status
- Complete list of submodules

Summary

Magnolia UI 6.x comes with a completely resurfaced UI featuring the new Find Bar at its core. We have also rewritten selected parts of the UI framework, particularly the content-app framework. We generally adhere to the same concepts as before, but we have developed better implementations by using Vaadin 8.

The new implementations simplify app mechanics and configuration and use less custom Vaadin code, thus enabling cleaner and faster customization.

Magnolia UI 6.x contains both the UI framework used in Magnolia 5.7 (referred to as Magnolia 5 UI) and the Magnolia 6 UI framework. You can use:

- Magnolia 5 UI components as well as your custom components and modules that rely on the previous Magnolia UI. We provide backward compatibility at both runtime and compile-time levels.
- New Magnolia 6 UI implementations.

All UI components from both implementations are rendered with the new resurfaced theme design.

Why we have changed things

The main reasons for rewriting some of the UI components include:

- Replacing the deprecated Vaadin 7 with Vaadin 8 (classes and frameworks)
- Developing new implementations that are more component-driven so that the components are easier to reuse
- Decreasing the amount of custom front-end (GWT) code and instead using the new stock Vaadin
- Simplifying app mechanics and configuration

The resulting implementations are easier to maintain for us and cleaner and faster to customize for you.

What is preserved

In Magnolia UI 6.2, the functions and frameworks known from version 5.7 and earlier remain. By preserving class names and Magnolia module names, we keep runtime backward compatibility of custom modules with custom classes extending Magnolia classes.

Class names
Class and package names of the classes existing in Magnolia 5.7 have not changed and are still available in Magnolia UI 6.2. You can use these classes or extend them in your custom code.

**Magnolia module names**

A Magnolia module defines dependencies to other Magnolia modules and many other tasks. A Magnolia module can be defined with one file (i.e. a module descriptor). The Magnolia module descriptors of the UI project reside in Maven modules.

In Magnolia UI 6.2, no Magnolia module has been removed. The names of the Magnolia modules are still the same.

All previously known Magnolia modules still exist with the same name and provide the same functions. However, the module descriptor files defining those Magnolia modules may have been relocated due to changes in the composition of the Maven modules.


**What has changed**

**Look and feel**

With Magnolia 6.x, the whole UI has been resurfaced to provide a modern, state-of-the-art user experience. At its core, the UI features the new **Find Bar**. All components of *Magnolia 5 UI* and *Magnolia 6 UI* are rendered using the new resurfaced theme design.

**Maven modules**

From Magnolia 5.7 and earlier, seven Maven modules have been removed. The classes of these modules are not lost. Instead, they have been relocated to the new `magnolia-ui-framework-compatibility` module or to the still existing `magnolia-ui-framework` module. New implementations have been added to the new `magnolia-ui-framework-jcr` module.

**Removed Maven modules**

- magnolia-ui-actionbar
- magnolia-ui-contentapp
- magnolia-ui-dialog
- magnolia-ui-form
- magnolia-ui-imageprovider
- magnolia-ui-vaadin-integration
- magnolia-ui-workbench

We have created Maven relocations for these modules. This ensures backward compatibility for compile-time dependencies.

**Added Maven modules**

- magnolia-ui-framework-jcr

```
<dependency>
  <groupId>info.magnolia.ui</groupId>
  <artifactId>magnolia-ui-framework-jcr</artifactId>
</dependency>
```

- magnolia-ui-framework-compatibility

```
<dependency>
  <groupId>info.magnolia.ui</groupId>
  <artifactId>magnolia-ui-framework-compatibility</artifactId>
</dependency>
```
Features

Magnolia UI 6.2 brings, among others, the following new features:

- Resurfaced design of the admin interface
- Integrated Find Bar
- Simplified app definition
- Improved back-end data binding
  - datasource replacing contentConnector for apps and subapps
  - Flexible and versatile provisioning of data source components (DataProvider, PropertySet, etc.)
  - Clear and reusable data source observation mechanism
  - DataProvider replacing Container
    - Works with domain objects (for example, JCR nodes) instead of items
  - PropertySet replacing Item
    - Describes how to interact with domain objects to access/modify their properties via lambda functions
- Flexible form definition
  - Simplified field API
  - Ability to define complex fields using form definitions
  - Separation of binding and logic
  - Custom layout support for forms
- Generic field definition
  - Better compatibility with the new, more type-safe Vaadin data-binding APIs

Magnolia 6 and Magnolia 5 UI frameworks

The Magnolia UI project version 6.x contains both the UI framework used in Magnolia 5.7 (referred to as Magnolia 5 UI) and the Magnolia 6 UI framework. You can use either framework.

Magnolia 5 UI

In Magnolia 6, Magnolia 5 UI has been relocated to the magnolia-ui-framework-compatibility module. Many Magnolia apps still rely on this old framework. The fully qualified class names have not changed, which ensures runtime compatibility for older custom components based on the classes from the Magnolia 5 UI framework.

Magnolia 6 UI

In the Magnolia 6 UI framework, the core components of the UI are located in the magnolia-ui-framework module inherited from Magnolia 5.7. The new magnolia-ui-framework-jcr module is also part of Magnolia 6 UI.

UI modules

For more information on the Magnolia 6 UI modules, refer to UI module.

Backward compatibility

The API changes are marginal. We provide backward compatibility at both runtime and compile-time levels.

Runtime compatibility

All classes from Magnolia UI 5.7 are also available in Magnolia UI 6.2. They have kept their fully qualified class names, which provides binary compatibility. Compiled custom modules (JAR files) built with Magnolia 5.7 also work in Magnolia 6.2 with the 6.2 UI.

Magnolia Maven module names are still the same. You do not have to change Magnolia module dependencies.

The binary compatibility together with the same-named Magnolia modules ensures runtime backward compatibility.

Compile-time compatibility

Module and submodule structures have been reorganized (see module changes). We have created Maven relocations for the removed modules. This ensures that you do not have to update the Maven module dependencies within the custom modules that depend on Magnolia UI modules.

How to upgrade

Make sure you upgrade your entire Magnolia bundle, not just the UI modules.
For general information on upgrading your Magnolia projects, read Upgrading to Magnolia 6.2.x.

**With Maven**

Make sure to set the version for the Magnolia UI project to 6.2.

**In your bundle(s)**

If you have a custom bundle based on a preconfigured Magnolia bundle, set the magnoliaBundleVersion property to 6.2. The Magnolia bundle will manage the UI version accordingly. For an example, see Upgrading to Magnolia 6.2.x: Updating Maven-managed webapps.

**In custom modules**

The composition of the submodules of the UI project has changed. Some modules have been removed, and their classes have been relocated to the magnolia-ui-framework-compatibility module. See What has changed: Maven modules.

Since we have created Maven relocations for the removed modules, you do not have to change the dependencies to them within your custom modules. However, to reflect the actual dependencies and to improve the readability of your POM files, you may want to replace the dependencies to the removed modules with the new compatibility module:

1. Remove all dependencies from these Maven modules:
   - magnolia-ui-actionbar
   - magnolia-ui-contentapp
   - magnolia-ui-dialog
   - magnolia-ui-form
   - magnolia-ui-imageprovider
   - magnolia-ui-vaadin-integration
   - magnolia-ui-workbench
2. Instead, add a dependency to magnolia-ui-framework-compatibility.

```xml
<dependency>
  <groupId>info.magnolia.ui</groupId>
  <artifactId>magnolia-ui-framework-compatibility</artifactId>
</dependency>
```

**Manually**

If you choose to do this manually, make sure you upgrade the entire bundle and not just the UI modules. Please follow the instructions on Upgrading to Magnolia 6.2.x: Updating manually.

**Status**

The following apps make use of the improved implementations (compatibility modules exist for their old implementations):

- Assets
- Definitions
- Notifications
- Pages
- Personas
- Preview
- Resource Files
- Segments
- Tags
- Tasks

Note that most of the other apps still rely on the old implementations. We will adapt them soon.

**Complete list of submodules**

Here is a list of all modules in version 6.2 of the Magnolia UI project. The list includes both the Maven modules and the Magnolia modules.
<table>
<thead>
<tr>
<th>Maven groupId</th>
<th>Function</th>
<th>Magnolia module name(s)¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>info. magnolia.ui magnolia-ui-project</td>
<td>Parent reactor</td>
<td>—</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-ui-framework</td>
<td><strong>The primary module/SDK for Magnolia app development</strong>  This provides (among others) the content app framework, IoC support for UI components and scopes, Vaadin 8 data-binding facilities and UI view/state management.  It also offers convenience builders for selected Magnolia UI components (for example, alerts and dialogs).</td>
<td>ui-framework-core</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-ui-api</td>
<td>This defines some core components for the UI such as Action, App, SubApp, Message and others.</td>
<td>—</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-ui-framework-jcr</td>
<td>JCR implementation of the <em>Magnolia 6 UI framework</em></td>
<td>ui-framework-jcr</td>
</tr>
<tr>
<td>info. magnolia.admincentral magnolia-admincentral</td>
<td>Magnolia 6 AdminCentral, an OS-like web application running Magnolia apps. This is where authors enter, preview and publish content and where administrators configure Magnolia.  <strong>⚠️ This module is considered a private API</strong> and may evolve rapidly without any guarantee of backward/binary compatibility. Developers are encouraged to use what the UI framework offers instead.</td>
<td>admincentral</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-resurface-theme</td>
<td>Magnolia 6 Vaadin theme based on Valo</td>
<td>—</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-ui-vaadin-common-widgets</td>
<td>Magnolia custom Vaadin/GWT components and extensions  This module contains mainly former Magnolia 5 widget implementations. For Magnolia 6, most of the components have been reimplemented with plain Vaadin. Consequently, this module may be split and partially deprecated in any future version.</td>
<td>—</td>
</tr>
<tr>
<td>info. magnolia.ui magnolia-ui-mediaeditor</td>
<td>Magnolia 6 UI Media Editor</td>
<td>ui-mediaeditor</td>
</tr>
</tbody>
</table>

**Magnolia stock apps**

<table>
<thead>
<tr>
<th>Maven groupId</th>
<th>Function</th>
<th>Magnolia module name(s)¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>info. magnolia.about magnolia-about-app</td>
<td>About Magnolia app</td>
<td>about-app</td>
</tr>
<tr>
<td>Magnolia 5 compatibility</td>
<td>Magnolia 5 AdminCentral</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>info. magnolia.ui</td>
<td>Magnolia 5 AdminCentral</td>
<td></td>
</tr>
<tr>
<td>magnolia-ui-admincentral</td>
<td>Accessible via /magnolia/admincentral-m5 for compatibility and migration purposes</td>
<td></td>
</tr>
<tr>
<td>info. magnolia.ui</td>
<td>Magnolia 5 AdminCentral</td>
<td></td>
</tr>
<tr>
<td>magnolia-ui-framework-compatibility</td>
<td>As of Magnolia 6.2, this module still provides (among others) the app launcher configuration, login form and user preferences dialog. These may be replaced or relocated to the new magnolia-admincentral module in any future version.</td>
<td></td>
</tr>
<tr>
<td>info. magnolia.ui</td>
<td>Magnolia 5 Vaadin theme</td>
<td></td>
</tr>
<tr>
<td>magnolia-ui-framework-compatibility</td>
<td>The following Magnolia 5 Maven modules have been relocated to this module:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-actionbar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-contentapp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-dialog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-form</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-imageprovider</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-vaadin-integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>magnolia-ui-workbench</td>
<td></td>
</tr>
</tbody>
</table>

1) Magnolia modules that belong to the given Maven submodule