

DEPLOYED MODULES LIFECYCLE AND ADMINISTRATION

DIGITAL EXPERIENCE MANAGER 7.1.1

SUMMARY

1	INTRODUCTION	3
2	MODULES INSTALLATION.....	4
2.1	Administration UI.....	4
2.2	Module installation	4
2.3	Basic information about modules.....	4
3	MODULES STATUS.....	6
3.1	Inactive.....	6
3.2	Inactive with missing dependency.....	6
3.3	Active.....	6
3.4	Error states	7
1.1.1	Definitions errors.....	7
1.1.2	Spring context error	7
1.1.3	Module not imported.....	8
3.5	Several versions available	8
3.6	Status at deployment time	9
4	MODULES DETAILED INFORMATION	10
4.1	General.....	10
4.2	Nodetypes.	10
4.3	Dependencies.....	10
4.4	Versions.....	11
4.5	Usage in sites	11
1.1.4	Enabling a module on a site	11
1.1.5	Disabling a module from a site.....	12
1.1.5.1	Simple disabling	12
1.1.5.2	Disabling a module and purge content	12

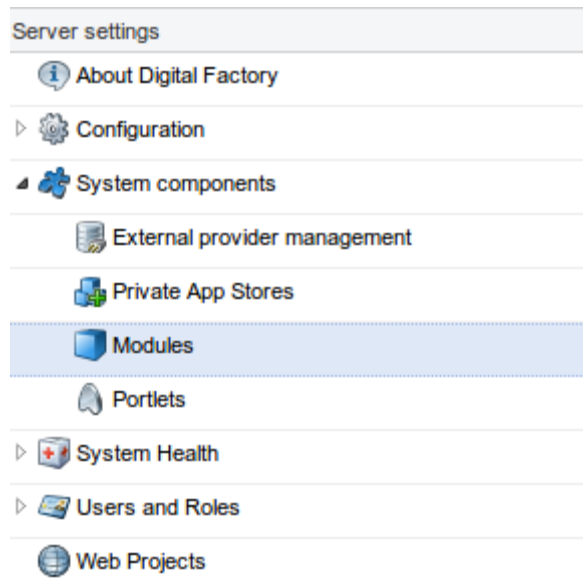
1 INTRODUCTION

Since version 7.0, modules are packaged as OSGi bundles in Digital Experience Manager, which make their management more powerful, and more granular, but also requested changes in the GUI to handle a more complex lifecycle. Before version 7 a module was basically installed on a server and then immediately active, and enabled on a site to make its components available in the content creation user interfaces. Since version 7, a module can be installed, but not started or intentionally stopped, it can be activated on a site and deactivated and even undeployed from a server. The purpose of this document is to explain those different states a module can have and the consequences on the platform, and make a link between the modules states and the modules management UI to make it clear to administrators.

2 MODULES INSTALLATION

2.1 ADMINISTRATION UI

You can manage the modules from the Administration Mode > System components > Modules



2.2 MODULE INSTALLATION

A module can be installed on a Digital Experience Manager instance

- by uploading manually a package (a jar file containing several jars, each one being a module) in the Modules administration page
- by uploading manually a jar file (an OSGi bundle for a unique module) in the Modules administration page
- by using the Modules Management GUI and installing in one click a module from a distant Public or Private App Store

2.3 BASIC INFORMATION ABOUT MODULES

In the modules administration page, modules are presented in a list of 7 columns, displaying basic information

- Module name: the name of the module
- Module ID: unique ID that defines the module, only one module by ID can be installed on a server at a time.
- Group ID: group id that defines the module, the couple group id/module id must be unique.
- Details: a button to access to a page that displays more detailed information about the module.
- Versions: shows the installed versions of the module, and possible actions on them
- Status:
 - active if the module is started
 - inactive if the module is stopped
- Used in site: 3 states possible
 - In use: the module is activated on one or several sites
 - In use (indirectly): the module is a dependency of a module activated on a site
 - In use (templates): a site uses the module as template

In *Development Mode*, a column named “source” appears and shows a button to download sources or to open the module in the studio when sources are already available locally.

3 MODULES STATUS

3.1 INACTIVE

Module name	Module ID	Group ID	Details	Versions	Status	Used in sites
Jahia Article	article		Details	Start Undeploy 2.0.1	Inactive	

The module is installed on the server (version 2.0.1 in the above example) but not currently started.

That means that Components, Views, templates, Rules, Workflows, Actions declared in the module are not usable at the moment. However, definitions are deployed, and exported classes are available for other modules.

The module can be started by clicking on the green button.

The module can also be undeployed (removed from the server). When a module is started it cannot be undeployed, administrators have to stop it first. You may also have to disable your modules from a site (section 4.5.2) before being able to undeploy it.

3.2 INACTIVE WITH MISSING DEPENDENCY

Module name	Module ID	Group ID	Details	Versions	Status	Used in sites
Bootstrap 3 Components	bootstrap3-components		Details	Depends on: bootstrap3-core Start Undeploy 1.1.8	Inactive	

Until the dependency is resolved, the module cannot be started. The missing dependency is written above the buttons.

3.3 ACTIVE

Module name	Module ID	Group ID	Details	Versions	Status	Used in sites
Jahia Article	article	org.jahia.modules	Details	Stop 2.0.1	Active	

When a module is started the stop button (red) appears, followed by the active version. This does not mean that the module is activated /in use on a particular site.


The module can be stopped. If the module is a dependency of other module(s), they will be stopped too. All the views and templates defined by the module won't be available anymore. Exported classes and node type definitions will always be available for other modules.

3.4 ERROR STATES

The system can encounter different errors when trying to deploy or start a new module.

1.1.1 Definitions errors



If the CND files cannot be parsed, the module won't be started, and the definition error will be displayed in the administration:

Module name	Module ID	Group ID	Details	Versions
My module <div style="background-color: #f8d7da; padding: 5px; border: 1px solid #f5c6cb;"> 1.0-SNAPSHOT : Invalid definitions: Unknown supertype jnt:contentxx for type jnt:test2 (URL [bundle://126.1:0/META-INF/definitions.cnd]) </div>	my-module		  Details 	  Reinstall  Undeploy 1.0-SNAPSHOT

A button « Reinstall » is available to retry the installation. In that case, it might be useful if you redeploy another module which provides the missing definition.

1.1.2 Spring context error

If the Spring context initialization of the modules crashes when starting the module, the error message is displayed in the administration:

Module name	Module ID	Group ID	Details	Versions	Status
My module <div style="background-color: #f8d7da; padding: 5px; border: 1px solid #f5c6cb;"> org.jahia.group.nonexistingclass </div>	my-module	org.jahia.modules	  Details 	  Stop 1.0-SNAPSHOT	<div style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">Active</div>

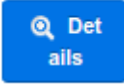



More details (full stacktrace) will be visible in the log file.

In production mode, the module won't be started if the Spring context cannot be loaded. In development, the module will be started and partly available (as shown in this example) – only spring context won't be loaded.

1.1.3 Module not imported

In cluster, if the module is deployed on one node but has not been deployed on processing server yet, an error showing « Module not imported » will be displayed.

3.5 SEVERAL VERSIONS AVAILABLE

Module name	Module ID	Group ID	Details	Versions
Jahia Article	article	org.jahia.modules	 Details	 Start  Undeploy 2.0.0  Stop 2.0.1

Several versions of a module can be installed at the same time. It is possible to choose which version should be started. **Only one version at a time can be started.** When switching from one version to another, all sites using that module will be impacted at the same time. This should be used carefully, as different versions of the same module will be used for different parts :

- Node types definitions always come from the latest version of the module
- All installed versions export classes, with different versions. If another module use classes, it will by default use the latest version.
- JSPs, Actions, RenderFilters, OSGi services, and other spring services will be taken from the active version of the module

If your modules contains definitions and/or export classes, it's strongly advised that you start only the last version of the module, to have a consistent behavior.

The coexistence of several versions allows administrators to easily proceed to upgrades and, if necessary and the possible inconsistencies described earlier are fully understood, downgrades to the previous version, if for any reason the new version reveals a problem.

To switch from one version to another, simply click on the start button of the desired version, this will automatically stop the current activated version.

3.6 STATUS AT DEPLOYMENT TIME

On a *Production server*:

- Upload a new module (or download and install from the store): the module is started immediately.
- Upload a new module (or download and install from the store) with a missing dependency: the module is not started until the dependency is resolved. When resolved the module is started automatically
- Upload a new version of a module (or update a module from the store): the new version is installed but not started, the current version is use remains the active one, until a positive action from the administrator.

On a Development server:

- Upload a new module (or download and install from the store): the module is started immediately.
- Upload a new module (or download and install from the store) with a missing dependency: the module is not started until the dependency is resolved. When resolved the module is started automatically
- Upload a new version of a module (or update a module from the store): the new version is **installed and started immediately in place of the previous version** (this is the only difference with a Production Server to ease developers work).

4 MODULES DETAILED INFORMATION

The « details screen » of a module contains various sections:

4.1 GENERAL

Modules - Jahia Article 2.0.1

These are the custom module (Jahia Article) for running on a Jahia server.

Module ID	Group ID	State	Module Type	Module provider	Module Sources
article	org.jahia.modules	Started	module	Jahia Solutions Group SA	Download sources

4.2 NODETYPES

List all the Nodetypes declared by the module and specify if the nodetype is also a component (usable as a drag and drop object in the edition interfaces)

Declared nodetypes

Nodetype	Is a usable component?
Article (title and introduction)	Yes
Article: paragraph	No

The installed node types are always coming from the latest version of the module that has ever been deployed. Whenever a new version of the module is deployed, its definitions are immediately installed into the repository. Starting or stopping the module does not have any impact on the definitions, and it is not possible to go back to an earlier version of the definitions. Note that even undeploying the module won't remove the definitions.

4.3 DEPENDENCIES

List all other modules the current module is relying-on. If dependencies are missing, the module cannot be started.

Dependencies	
Module	Details
Jahia jQuery	Details
Jahia CKEditor	Details
Jahia Static Assets	Details
Default Jahia Templates	Details

4.4 VERSIONS

List the versions of the module installed available on the server. Only one version can be started at a time.

Versions		
Version	State	Manage
2.0.0	Installed	Start Undeploy
2.0.1	Started	Stop

From this screen, like on the modules list, you can switch from versions, only one version of a module can be started at a time.

4.5 USAGE IN SITES

1.1.4 Enabling a module on a site

Enabling a module on a site makes its components, views and templates available on the site.

Usage in sites		
Site	Usage	Manage
systemsite	Not enabled	Enable
mySite	In use	Disable
ACMESPACE	Not enabled	Enable
wsfaccess	Not enabled	Enable

[Disable on all sites](#)

1.1.5 Disabling a module from a site

1.1.5.1 Simple disabling

Disabling a module from a site impeach the usage of components (content objects) declared by the module by editors. The content already created with that module remains present but cannot be copied nor edited. Templates defined in the module are not available anymore. Digital Experience Manager won't be able to generate pages relying on those templates and that will lead to 404 errors.

1.1.5.2 Disabling a module and purge content

When disabling a module from a site, administrators can choose to purge all content related to this module in the site (content created using the definitions declared by this module). Be aware that **this operation cannot be reversed**, if you are sure that the content should not appear anymore, you can do the purge, if you just want editors to stop creating new content using one of the content-types defined by the module, do not purge and just disable the module.

