_Known issues

Systems

Tomcat

Trailing slash issue

When accessing the webapp without the trailing slash after the context http://localhost:8080/magnoliaAuthor after login the user is presented with a 404 page rather than admincentral. This behavior is configurable via the `mapperContextRootRedirectEnabled` and `mapperDirectoryRedirectEnabled` attributes of the Context which may be used to restore the previous behavior.

Reloading a page causes a 404 error

When you open a page for editing and reload it in the browser, an extra semicolon is added to the URL fragment and the server returns a 404 error. Every subsequent reload adds another semicolon. This issue occurs with older versions of the Tomcat application server. To resolve this issue, upgrade to Tomcat 7.0.47+. See ~~MGNLUI-2426~~ - Reloading page in editor causes 404 CLOSED

Tomcat 8 using BCEL may throw class format exception

On Tomcat 8.5 or lower, you may encounter a class format exception.

```
SEVERE [localhost-startStop-1] org.apache.catalina.startup.ContextConfig.processAnnotationsJar Unable to
process Jar entry [module-info.class] from Jar [file:/home/magnolia/dev/magnolia-5.6.6/apache-tomcat-8.5.5
/webapps/magnoliaPublic/WEB-INF/lib/javax.json-api-1.1.jar] for annotations
 org.apache.tomcat.util.bcel.classfile.ClassFormatException: Invalid byte tag in constant pool: 19
```

Reason:

Apache Commons BCEL is used by Tomcat to scan for annotations. Some Java EE 8 APIs were produced from JDK9 as Multi-Release JARs. This means they're compatible with Java 8, but may also contain some Java 9 classes, including the `module-info.class`. Older Commons BCEL versions did not know how to read such classes.

Tomcat 8.5.12 and 9.0.0 updated Commons BCEL to support this. See the extract below from the Tomcat changelog.

> *60688: Update the internal fork of Apache Commons BCEL to r1782855 to add early access Java 9 support to the annotation scanning code. (markt)*

Solution:

Use Tomcat 8.5.12 or higher.

IBM Websphere and Linux

If you experience a JVM crash when running Magnolia 6.2 on Linux and IBM Websphere server, disable the Periscope Result Ranker module by excluding the `magnolia-periscope-result-ranker` artifact from the dependencies of your project. For example:

```
  <dependencies>
    <dependency>
      <groupId>info.magnolia.dx</groupId>
      <artifactId>magnolia-dx-core-demo-webapp</artifactId>
      <type>war</type>
      <exclusions>
        <exclusion>
          <groupId>info.magnolia.periscope</groupId>
          <artifactId>magnolia-periscope-result-ranker</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
    <dependency>
      <groupId>info.magnolia.dx</groupId>
      <artifactId>magnolia-dx-core-demo-webapp</artifactId>
      <type>pom</type>
      <exclusions>
        <exclusion>
          <groupId>info.magnolia.periscope</groupId>
          <artifactId>magnolia-periscope-result-ranker</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
    <dependency>
      <groupId>info.magnolia</groupId>
      <artifactId>magnolia-module-websphere</artifactId>
    </dependency>
  </dependencies>
```

JBoss 7.2 with JDK 11

If you are using JBoss 7.2 with JDK 11 to deploy Magnolia, you may get an error during installation. To resolve the issue, add the following lines to `WEB-INF/jboss-deployment-structure.xml`:

```
<deployment>
 <dependencies>
  <module name="jdk.unsupported"/>
 </dependencies>
</deployment>
```

MySQL JDBC 8.0.x

If you are using the MySQL JDBC driver 8.0.x to install a new Magnolia instance, you may encounter the following error:

```
2020-02-14 14:35:07,538 ERROR org.apache.jackrabbit.core.RepositoryImpl: failed to start Repository: org.apache.
jackrabbit.core.state.ItemStateException: failed to read bundle (stacktrace on DEBUG log level): deadbeef-face-
babe-cafe-babecafebabe: java.sql.SQLSyntaxErrorException: Table '<DATABASE_NAME>.<TABLE_NAME>' doesn't exist
```

This happens only when you install a new instance from scratch or create database tables during installation. As a workaround, use JDBC 5.1.x or create the tables manually in your database.

Oracle WebLogic

Magnolia 6.2 is not compatible with Oracle WebLogic.

OS X / macOS

Magnolia starts up very slowly

Your Magnolia instance starts up very slowly, noticeably more than before. You may encounter this after upgrading your macOS to Sierra 10.12 or later.

Workaround:

1. Open a terminal and figure out the hostname of your Mac. To get the hostname, use the command `hostname`.

   ```
   hostname
   joesLittleMacBookPro.local
   ```

   The command returns the hostname of your Mac. In the given example, the hostname is `joesLittleMacBookPro.local`.

2. Open the file `/etc/hosts` with an editor of your choice. You will edit the file in the next step.
   The file belongs to the system user `root`. Your user must belong to the group admin; otherwise, you cannot save the changes to the file. You will be asked for your password either when opening the file or when trying to save it.

3. Add these two lines to `/etc/hosts`:

   /etc/hosts

   ```
   127.0.0.1   <hostname>
   ::1         <hostname>
   ```

   In place of `<hostname>`, use your real hostname.
   Most probably, your hosts file already contains entries starting with `127.0.0.1` and with `::1`. Just add the real hostname at the end of one line that starts with `::1` and at the end of another line that starts with `127.0.0.1`. Make sure to add a space character before the hostname.
   With the given hostname, the two lines would look like this:

   /etc/hosts

   ```
   127.0.0.1   localhost joesLittleMacBookPro.local
   ::1         localhost joesLittleMacBookPro.local
   ```

4. Save the file. The system may ask you for your password.

Further information:

- https://thoeni.io/post/macos-sierra-java/
- https://stackoverflow.com/questions/33289695/inetaddress-getlocalhost-slow-to-run-30-seconds/33289897#33289897

Windows

JVM issues

32-bit JVM does not work in Windows

(ℹ Tested on Windows 10.) When starting up Tomcat with the Magnolia CLI `mgnl start` command, you get a flickering window, the server hangs and nothing is written into the logs. To see the actual error, don't use Magnolia CLI to start Magnolia, start Tomcat directly with `<your-magnolia-install-folder>\apache tomcat\bin\catalina.bat run`. This should start Magnolia in the same window and allow you to see the error message:

```
Error occurred during initialization of VM
Could not reserve enough space for 2097152KB object heap
```

The most likely cause is that you are trying to allocate too much heap space in the 32-bit JVM.

Solution: Replace the JVM with a 64-bit version.

See also: Why can't I get a larger heap with the 32-bit JVM?

Neither JAVA_HOME nor JRE_HOME is defined

You get the following error message after a fresh Java installation:

```
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variables is needed to run this program
```

Solution: Set the variables through the "Edit the system environment variables" dialog.

JRE_HOME is undefined

If you get an error like this one:

```
The JRE_HOME environment variable is not defined correctly
The environment variable is needed to run this program
```

then you are most probably attempting to run Tomcat with the `JAVA_HOME` variable set but without the `JRE_HOME` variable defined.

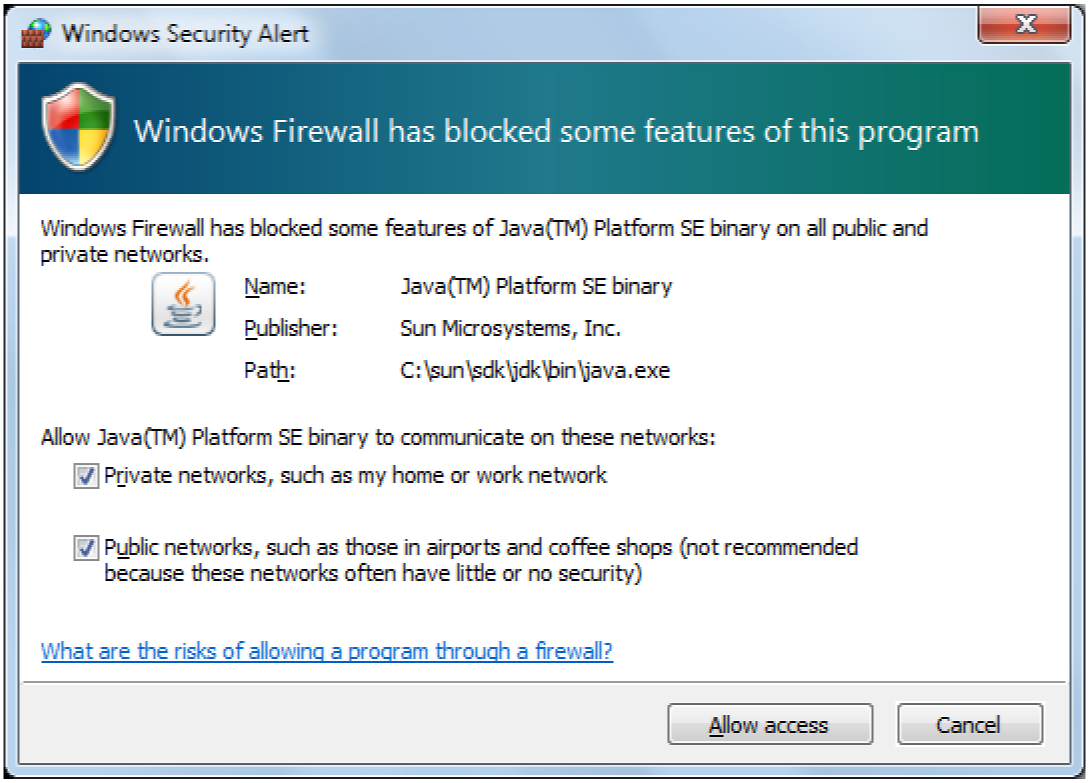
Solution: Set the `JRE_HOME` variable.

Firewall is blocking Java

Allow an exception in Windows Firewall for Java.

1. Go to **Control Panel > Windows Firewall**.
2. Go to the **Exceptions** tab.
3. Click **Add Program** and browse to the java.exe file in the Java installation directory, for example `C:\Sun\SDK\jdk\bin\java.exe`.
4. Click **OK**.

If you get a security alert during startup, check the **Private networks** checkbox and click **Allow access**.



Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of Java(TM) Platform SE binary on all public and private networks.

Name: Java(TM) Platform SE binary

Publisher: Sun Microsystems, Inc.

Path: C:\sun\sdk\jdk\bin\java.exe

Allow Java(TM) Platform SE binary to communicate on these networks:

☑ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

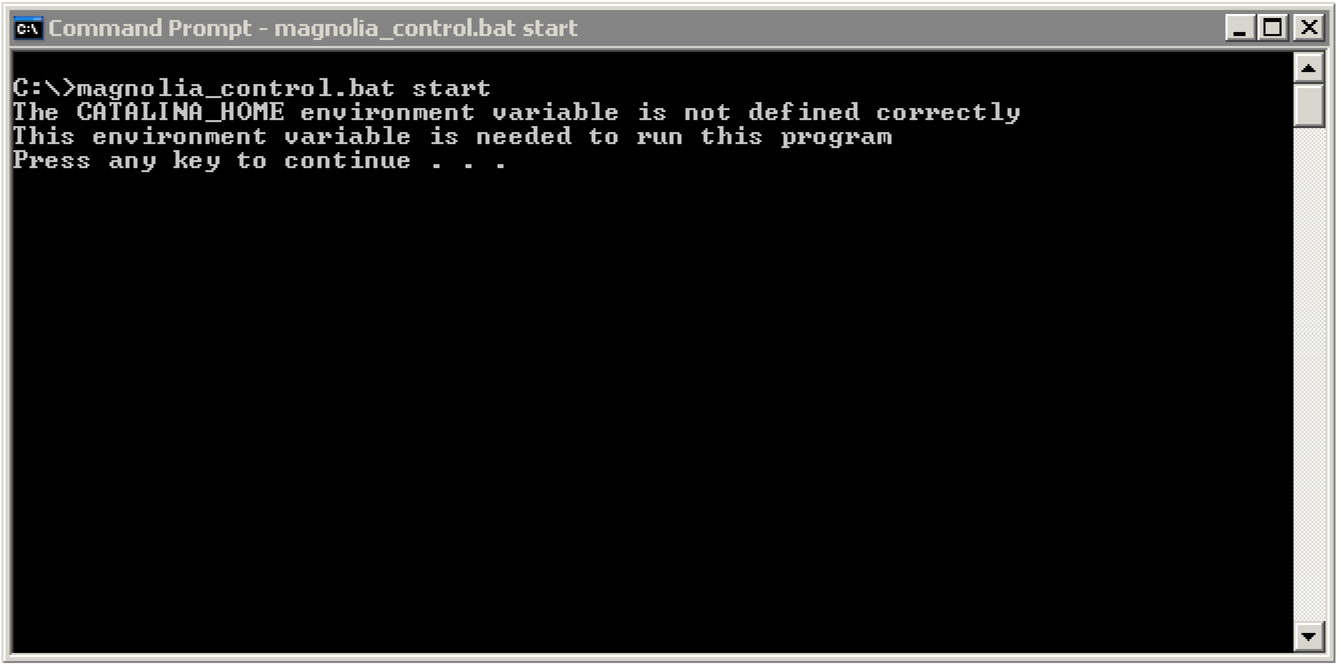
What are the risks of allowing a program through a firewall?
Allow access
Cancel

CATALINA_HOME environment variable is not defined

CATALINA_HOME environment variable identifies the Tomcat home directory, for example `C:\Program Files\magnolia\apache-tomcat`. Usually Magnolia finds this directory automatically. When you type `magnolia_control.bat start` in the `bin` directory to start the system, a second script named `startup.bat` tries to find Tomcat home. It assumes that Tomcat home is one level above the `bin` directory where you issued the command, and sets the value of CATALINA_HOME to that directory.

```
C:\Program Files
    magnolia
        apache-tomcat   <-- Tomcat home directory
            bin         <-- magnolia_control.bat is here
```

However, if you added the `bin` directory to your PATH environment variable you can execute `magnolia_control.bat` from anywhere. This means `startup.bat` does not find Tomcat home directory by simply moving up one level from where you are and displays the following error:



```
C:\>magnolia_control.bat start
The CATALINA_HOME environment variable is not defined correctly
This environment variable is needed to run this program
Press any key to continue . . .
```

To correct this define CATALINA_HOME in environment variables. Follow the instructions in Set JAVA_HOME environment variable.

JAR file does not start

When installing DX Core on Windows, you can start the installer by double-clicking the JAR file. If this does not work there is a chance that some application on your system has registered the .jar extension.

You can try to fix it yourself by restoring the association of the .jar extension with the `javaw.exe` executable. .Right-click the JAR file and select **Open With** . Typically the `javaw.exe` file is in `C:\Program Files (x86)\java\jre6\bin`). Alternatively, start the installer from a command prompt with the following command: `java -jar magnolia-dx-core-installer-x.y.z.jar`. Make sure the file extension is .jar. Internet Explorer has a tendency to append or change it to .zip.

Other issues

H2 database

⚠ We don't recommend using H2 database for production environments.

H2 does not accept more than one connection

Our default configuration uses the server mode for H2. If you migrate from Magnolia below 5.5.9 in the 5.5 branch or below 5.6.2 in the 5.6 branch and try to initiate a backup call using CLI or REST, it fails because H2 does not allow more than one connection at a time. Configure H2 to run in server mode by adding `AUTO_SERVER=TRUE` in the URL parameters:

```
<param name="url" value="jdbc:h2:${wsp.home}/db;AUTO_SERVER=TRUE" />
```

```
<param name="url" value="jdbc:h2:${rep.home}/version/db;AUTO_SERVER=TRUE" />
```

See ~~MGNLCE-114~~ - H2 doesn't accept more than one connection anymore CLOSED

Log full of index rules warnings when upgrading from 5.4.x directly to 6.2.x

Jackrabbit 2.16 introduced existence checks for node-types when refreshing indexing configurations. Previously it would accept arbitrary node-type names as strings. If your project is not using Magnolia's SearchIndex implementation, you may find the following messages in your logs:

```
2018-07-25 17:48:33,691 WARN  rabbit.core.query.lucene.IndexingConfigurationImpl: Unable to refresh index rules
javax.jcr.nodetype.NoSuchNodeTypeException: {http://www.magnolia.info/jcr/mgnl}page
        at org.apache.jackrabbit.core.nodetype.NodeTypeRegistry.getNodeTypeDef(NodeTypeRegistry.java:552) ~
[jackrabbit-core-2.16.1.jar:2.16.1]
```

```
2018-03-16 11:30:57,892 WARN  rg.apache.jackrabbit.core.query.lucene.SearchIndex: Exception initializing
indexing configuration from: /info/magnolia/jackrabbit/indexing_configuration_website.xml
javax.jcr.nodetype.NoSuchNodeTypeException: {http://www.magnolia.info/jcr/mgnl}page
        at org.apache.jackrabbit.core.nodetype.NodeTypeRegistry.getNodeTypeDef(NodeTypeRegistry.java:552) ~
[jackrabbit-core-2.16.1.jar:2.16.1]
```

To mitigate this problem, replace in your Jackrabbit search configuration files

```
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
```

with

```
<SearchIndex class="info.magnolia.jackrabbit.lucene.SearchIndex">
```

The warnings should not appear in the logs if you follow our recommendation to update first to the latest minor release version before upgrading to the most recent major release, see How to update.

Too many open files

The embedded Derby database, an option that can be used instead of the H2 database, opens several file handles and may run over the maximum limit set by the system. This issue can occur on some Linux and OS X systems such as Macbook Air.

The solution is to increase the system-wide limit on the number of open files. The exact procedure varies from one OS to the next, see Too many open files.

1. Find the current maximum number of open files per user in a single session:
   `ulimit -n`
   By default the number is 1024 which is too small.
2. Edit the `limits.conf` file:
   `sudo gedit /etc/security/limits.conf`
3. Add the following lines to the file:
   `* soft nofile 10000`
   `* hard nofile 50000`
   This sets for all users a soft limit of 10000 open files and a hard limit of 50000. These are just example numbers. Set them according to your system needs. Note that the wildcard option applies only to regular users, not to superuser. If you run Magnolia as superuser replace the asterisk with `root`.
4. Save the file.
5. Edit the configuration file for session-related modules:
   `sudo gedit /etc/pam.d/common-session`
6. Add the following line to the file:
   `session required pam_limits.so`

   > On *ubuntu 17.04 you'll probably also have to add the following line
   >
   > `DefaultLimitNOFILE=65535`
   >
   > to `/etc/systemd/system.conf`
   >
   > and to `/etc/systemd/user.conf` .
   >
   > See Cannot Increase open file limit past 4096 (Ubuntu).

7. Save the file.
8. Restart Ubuntu.
9. Verify the new maximum number of open files:
   `ulimit -n`
   The command should now return 10000.

OS X 10.8 Mountain Lion:

1. Find the current maximum number of open files per user in a single session:
   `ulimit -n`
   By default the number is 256 which is too small.
2. Add `ulimit -n 65536` to your `~/.profile` file. This increases the limit for the shell.
3. Create a file `/etc/sysctl.conf` if it doesn't exist.
4. Add the following lines in `/etc/sysctl.conf`. This increases the limit for the kernel.
   ```
   kern.maxfiles=65536
   kern.maxfilesperproc=65536
   ```
5. Open a terminal and type:

   ```
   sudo sysctl -w kern.maxfiles=65536
   sudo sysctl -w kern.maxfilesperproc=65536
   sudo ulimit -n 65536
   ```

   -- OR --
   Restart OSX to read the settings from the files you edited.
6. Type `ulimit -n`. The response should be `65536`. (If not, restart OSX)
7. Install Magnolia

OS X 10.9 (Mavericks), 10.10 (Yosemite) and 10.11 (El Capitan), see Shell Session Limit

OS X 10.12 (Sierra) and 10.13 (High Sierra), see the answer given for For OS X Sierra (10.12.X) on this page.

Session de-serialization

When installing or updating to a new version, you may see this error message:

```
2009-11-24 13:02:14,970 ERROR org.apache.catalina.session.ManagerBase: IOException while loading persisted
sessions
```

This can be due to changes in signatures of classes that are stored in user sessions, such as permissions, user, etc. The error happens when Tomcat attempts to de-serialize serialized sessions as the container starts. The de-serialization causes the loss of persisted sessions. Users will have to log in again. Otherwise it is a harmless error and can be ignored.

Port 8080 is already in use

Port 8080 is the default port for Tomcat. You can see it at the end of the default address `http://localhost:8080` . If another application on the computer is already using the same port, you may need to change it.

1. Open `<CATALINA_HOME>/conf/server.xml` in a text editor. This file is under your Magnolia installation directory.
2. Find the following section and set the value of `port` to something other than 8080, for example 8090:

   ```
   <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
   <Connector port="8090" maxHttpHeaderSize="8192"
              maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
              enableLookups="false" redirectPort="8443" acceptCount="100"
              connectionTimeout="20000" disableUploadTimeout="true" />
   ```

Change the `defaultBaseUrl` property, which is used to create absolute links in emails or other external systems. To do this, you need to now access Magnolia at the new port 8090.

1. Log into the author instance at `http://localhost:8090/magnoliaAuthor/.magnolia` .
2. Go to **Configuration**.
3. Set the value of `/server/defaultBaseUrl` property to `http://localhost:8090/magnolia/`.
4. Log into the public instance at `http://localhost:8090/magnoliaPublic/.magnolia` .
5. Go to **Configuration**.
6. Set the value of `/server/defaultBaseUrl` property to `http://localhost:8090/magnolia/`.

Now the Welcome page at `http://localhost:8090` has the correct URLs too.

The port also needs to be changed in publishing configuration, otherwise publishing (activating) changes from the author to public instance fails.

1. On the author instance,

    a. If you are using the Publishing module, go to **Configuration > modules > publishing-core > config > receivers** .
    b. If you are using the Activation module, go to **Configuration > modules > activation > subscribers** .
2. Under the `magnoliaPublic8080` receiver (subscriber), set the `URL` property to `http://localhost:8090/magnoliaPublic` .
3. Rename the receiver (subscriber) to `magnoliaPublic8090`.
4. Activate the modified receiver (subscriber) including its subnodes.

> If you want to run two different Tomcats simultaneously you need to change other ports too. This is useful if you want to run different versions of Magnolia at the same time. In `<CATALINA_HOME>/conf/server.xml`, change the port numbers for the shutdown and AJP sections and any custom sections you have enabled.

Java out of memory

If the Java Virtual Machine (JVM) does not have enough memory you may see a `java.lang.OutOfMemoryError` in the startup log and Magnolia fails to start.

```
Exception in thread "Timer-1" java.lang.OutOfMemoryError: Java heap space
at org.apache.jackrabbit.core.query.lucene.IndexingQueue.getFinishedDocuments
```

Increase Java heap size to allocate more memory to JVM:

1. Stop the server.
   `./magnolia_control.sh stop`
2. Open file `/apache-tomcat/bin/setenv.sh` (`/apache-tomcat/bin/setenv.bat` on Windows) in a text editor.
3. Edit the `Xmx` parameter to set a new maximum heap size. Default size for Magnolia is 2048M, try a higher amount such as 4096M.
4. Save the file and start the server.
   `./magnolia_control.sh start && tail -f ../logs/catalina.out`

Publishing and activation errors

For issues related to publishing/activation keys and the handshaking process, see either the Publishing errors page or the Activation errors page, depending on the module you use.

New security filter

MAGNOLIA-6865 introduced a new security filter that checks each POST request. As this may affect the functionality of your existing forms, please contact us and we'll provide you with more information about this issue.

For security reasons, however, we cannot disclose further details on this page.

Content stored in CE apps not displayed

In the Community Edition, if you edit content created in an app and the instance has no default i18n support, the content will not be displayed in the detail subapp of the app. (MGNLUI-5770)

Rendering Single-Page Applications (SPA)

Currently, the following functionalities are not supported in SPA rendering:

- component autogeneration
- component inheritance
- component personalization
- fragment definition
- area-in-area nesting

HTTP range requests from Facebook don't work

Facebook uses HTTP range requests to read Open Graph tags. This may cause that the Facebook crawler is not working for Magnolia websites. A workaround for this issue is to use a voter which bypasses the gzip filter if the incoming request is of the range type and is coming from Facebook:

config.server.filters.gzip.bypasses.facebookCrawler.yaml

```
'facebookCrawler':
  'class': info.magnolia.voting.voters.BoolVoterSet
  'op': AND
  'voters':
    'userAgent':
      'class': info.magnolia.voting.voters.RequestHeaderPatternRegexVoter
      'headerName': User-Agent
      'pattern': .*facebookexternalhit.*
    'rangeRequest':
      'class': info.magnolia.voting.voters.RequestHeaderPatternRegexVoter
      'headerName': Range
      'pattern': .*
```

Virtual URI mappings not working if too many are configured

To mitigate an issue caused by having more than 500 configured virtual URI mappings in light modules, a WARN-level message is now logged when a `DirectoryWatcher` overflow occurs (MAGNOLIA-7762). We recommend to keep the number of files in a single folder below 100 and to use folder hierarchies whenever possible. For the upcoming fix, see MAGNOLIA-7798.

Natural order not respected in results of content searches

If you search for content using `searchfn.searchPage()`, `searchfn.searchContent()` or by executing search queries directly from a template, the search result order will not respect the natural order. This issue is caused by a bug in Apache Jackrabbit filed as JCR-3932: "Result of JCR-SQL2 query doesn't respect document order in JCR".

Until the JCR bug is fixed, do not use `searchfn`. To get the correct order, execute JCR XPath queries directly.

In Blossom module, DialogCreationContext attributes are not initialized

In the latest release of the Blossom module, there are still certain objects that the old `BlossomFormDialogPresenter` used to make injectable, mostly Vaadin 7 items, JCR Node and node path.

For the time being, please keep using the Pages compatibility app for these use cases. Magnolia 6 UI framework does not have configurable form-dialog presenters, nor Vaadin 7 items.

The issue has been fixed only for the compatibility Pages app (`magnolia-pages-app-compatibility`), not for the new Pages app (`magnolia-pages-app`). For more details, see the following ticket:

~~BLOSSOM-269~~ - DialogCreationContext attributes are not initialised CLOSED

DL4J/ND4J issues

There are limitations related to the deep learning and search features of the Find Bar, which are provided by the Periscope and Periscope Result Ranker modules.

Off-heap memory starvation

You should configure DL4J memory limits to avoid off-heap memory starvation. This is particularly the case with the `-Dorg.bytedeco.javacpp.maxbytes` JVM argument. For more details, see MGNLPER-121.

Periscope prevents deploying the public webapp as ROOT along with the author instance

When experiencing this issue, the following message can be found in the logs: `java.lang.NoClassDefFoundError: Could not initialize class org.nd4j.linalg.factory`. The issue may appear even on more than just one Tomcat instance, running on the same VM and the same JDK.

The ndj4 library can only be initiated once. Therefore, result ranking on public instances should be disabled, which you can do either by changing the configuration of the Periscope Result Ranker module, or rather by completely removing the `magnolia-periscope-result-ranker` module from your WAR package. After this modification, the search function will be available but search results will not be ranked. For more details see MGNLPER-112.

AVX/AVX2 CPU Support

Advanced Vector Extensions are a set of instructions extending the instruction set of the x86 family of microprocessors. AVX was first released in March 2008 for Intel and AMD. AVX2 was first released 2013. If your hardware uses a processor which is not supporting AVX2 then you will see a message similar to:

```
libnd4j binary was built with AVX/AVX2 support, but current CPU doesn't have this instruction set. Exiting
now...
```

By default Magnolia is shipped with the AVX2 libraries installed:

- `nd4j-native-1.0.0-beta6-macosx-x86_64-avx2.jar`
- `nd4j-native-1.0.0-beta6-windows-x86_64-avx2.jar`
- `nd4j-native-1.0.0-beta6-linux-x86_64-avx2.jar`

```
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>linux-x86_64-avx2</classifier>
</dependency>
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>macosx-x86_64-avx2</classifier>
</dependency>
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>windows-x86_64-avx2</classifier>
</dependency>
```

If your CPU only supports AVX then use the AVX libs instead:

- `nd4j-native-1.0.0-beta6-macosx-x86_64.jar`
- `nd4j-native-1.0.0-beta6-windows-x86_64.jar`
- `nd4j-native-1.0.0-beta6-linux-x86_64.jar`

```
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>linux-x86_64</classifier>
</dependency>
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>macosx-x86_64</classifier>
</dependency>
<dependency>
  <groupId>org.nd4j</groupId>
  <artifactId>nd4j-native</artifactId>
  <classifier>windows-x86_64</classifier>
</dependency>
```