Alfresco SDK 3.0

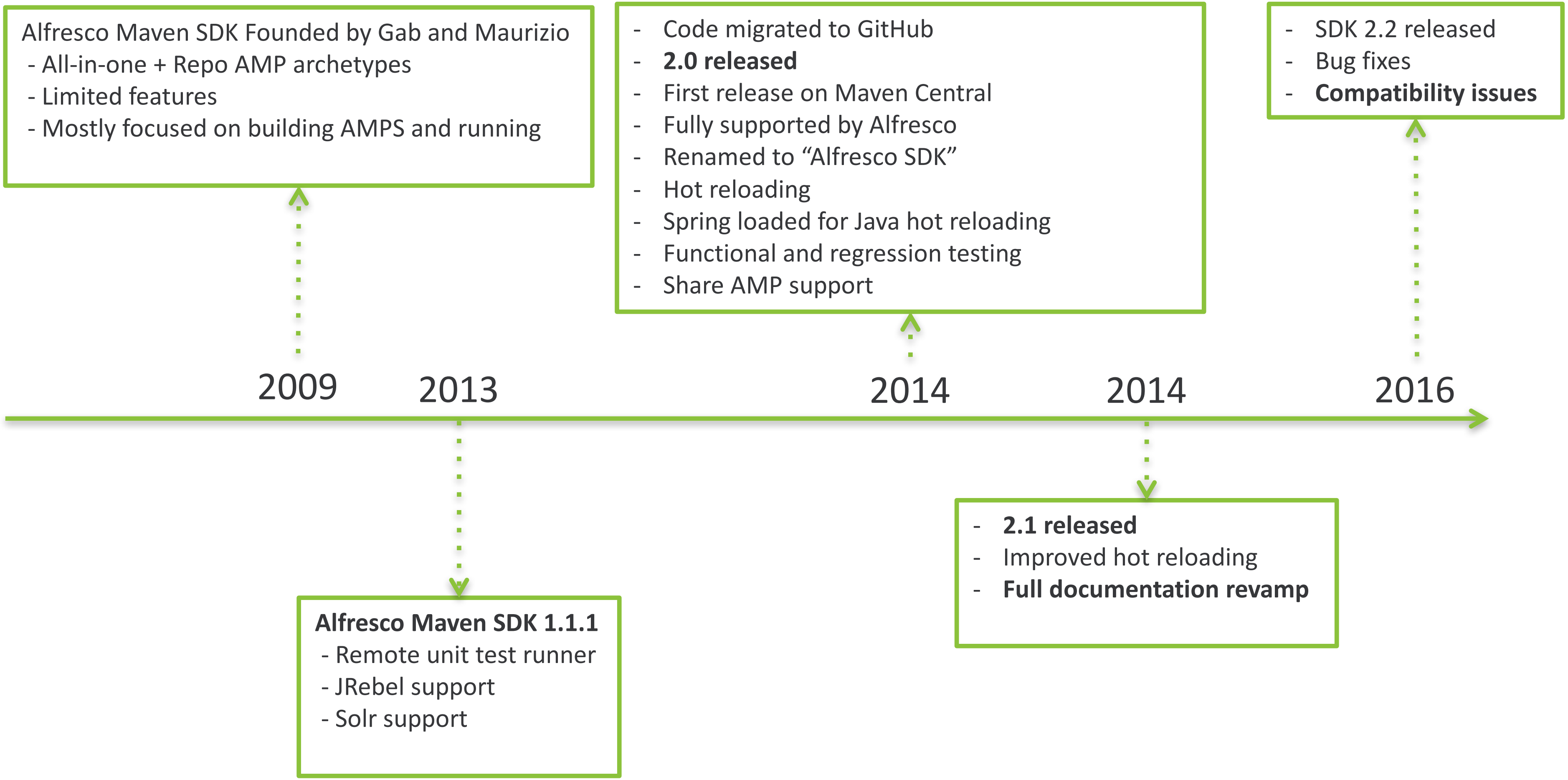
Ole Hejlskov
Product Manager, Developer Platform

Agenda

- Brief History
- What’s new in Alfresco SDK 3.0
- …. ”about that Maven plugin.. ??”
- Demo
- Questions

Brief History

Alfresco®



2009

Alfresco Maven SDK Founded by Gab and Maurizio

- All-in-one + Repo AMP archetypes
- Limited features
- Mostly focused on building AMPS and running

2013

Alfresco Maven SDK 1.1.1

- Remote unit test runner
- JRebel support
- Solr support

2014

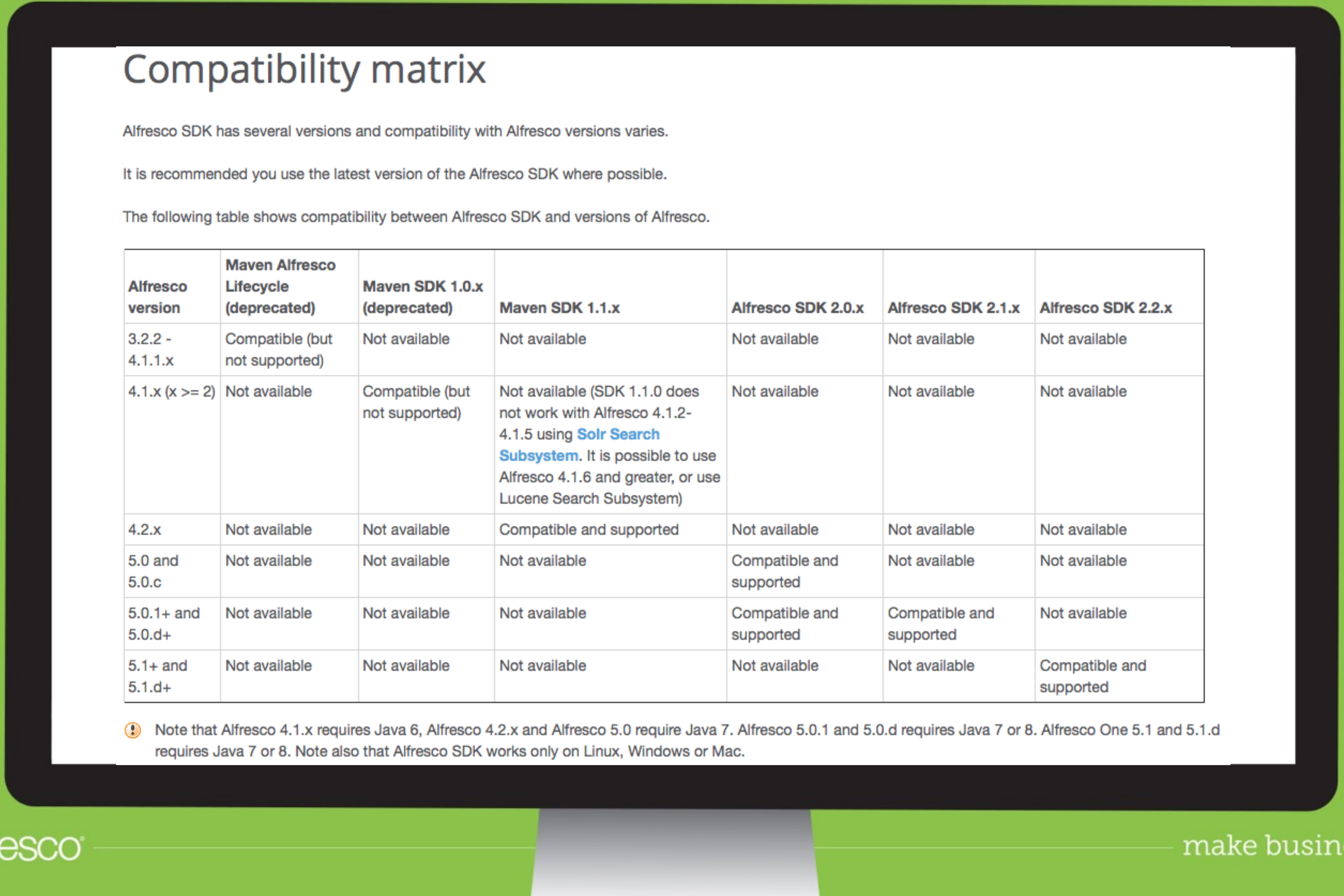
- Code migrated to GitHub
- **2.0 released**
- First release on Maven Central
- Fully supported by Alfresco
- Renamed to “Alfresco SDK”
- Hot reloading
- Spring loaded for Java hot reloading
- Functional and regression testing
- Share AMP support

2014

- **2.1 released**
- Improved hot reloading
- **Full documentation revamp**

2016

- SDK 2.2 released
- Bug fixes
- **Compatibility issues**



Compatibility matrix

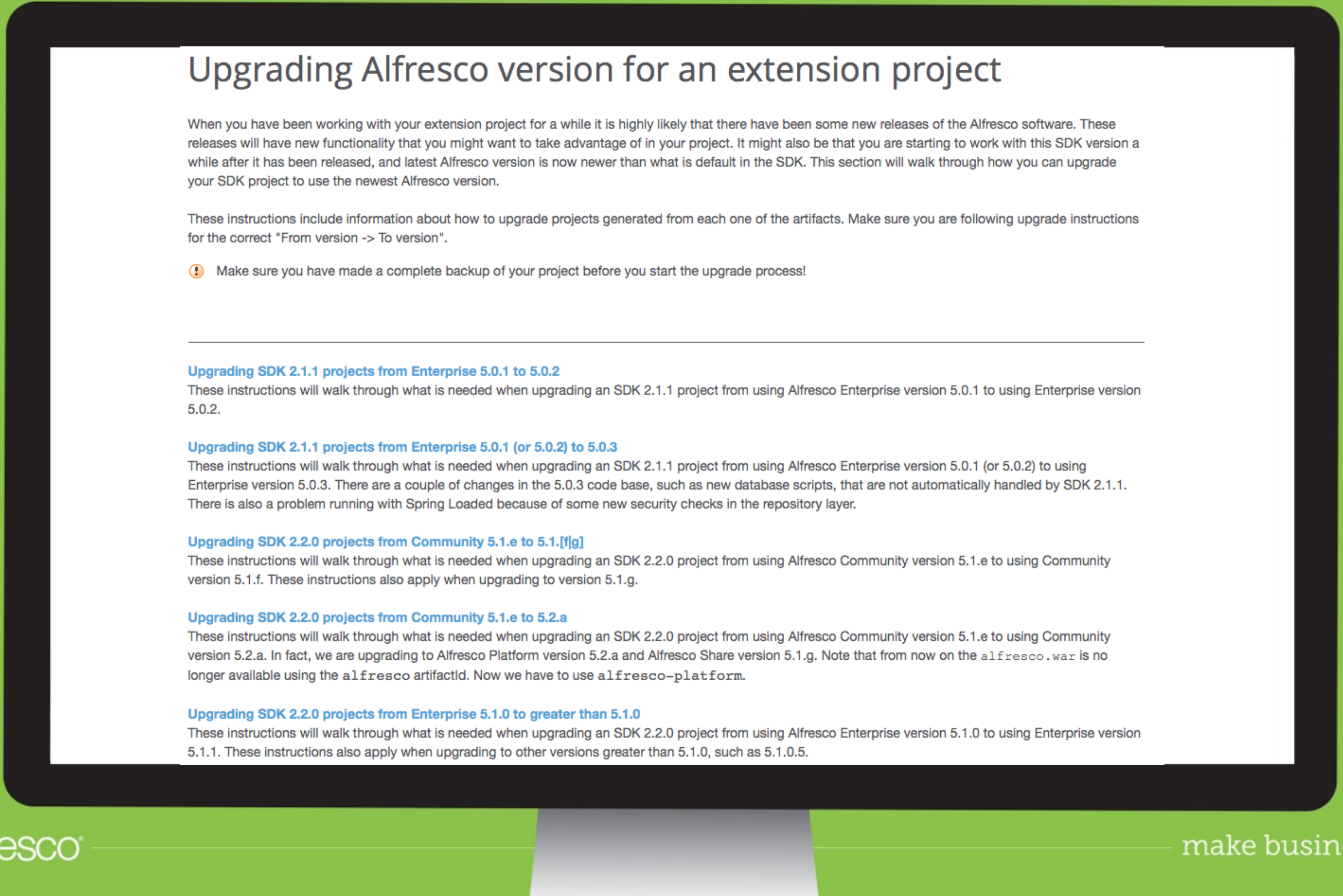
Alfresco SDK has several versions and compatibility with Alfresco versions varies.

It is recommended you use the latest version of the Alfresco SDK where possible.

The following table shows compatibility between Alfresco SDK and versions of Alfresco.

Alfresco version	Maven Alfresco Lifecycle (deprecated)	Maven SDK 1.0.x (deprecated)	Maven SDK 1.1.x	Alfresco SDK 2.0.x	Alfresco SDK 2.1.x	Alfresco SDK 2.2.x
3.2.2 - 4.1.1.x	Compatible (but not supported)	Not available	Not available	Not available	Not available	Not available
4.1.x (x >= 2)	Not available	Compatible (but not supported)	Not available (SDK 1.1.0 does not work with Alfresco 4.1.2-4.1.5 using Solr Search Subsystem. It is possible to use Alfresco 4.1.6 and greater, or use Lucene Search Subsystem)	Not available	Not available	Not available
4.2.x	Not available	Not available	Compatible and supported	Not available	Not available	Not available
5.0 and 5.0.c	Not available	Not available	Not available	Compatible and supported	Not available	Not available
5.0.1+ and 5.0.d+	Not available	Not available	Not available	Compatible and supported	Compatible and supported	Not available
5.1+ and 5.1.d+	Not available	Not available	Not available	Not available	Not available	Compatible and supported

Note that Alfresco 4.1.x requires Java 6, Alfresco 4.2.x and Alfresco 5.0 require Java 7. Alfresco 5.0.1 and 5.0.d requires Java 7 or 8. Alfresco One 5.1 and 5.1.d requires Java 7 or 8. Note also that Alfresco SDK works only on Linux, Windows or Mac.



Upgrading Alfresco version for an extension project

When you have been working with your extension project for a while it is highly likely that there have been some new releases of the Alfresco software. These releases will have new functionality that you might want to take advantage of in your project. It might also be that you are starting to work with this SDK version a while after it has been released, and latest Alfresco version is now newer than what is default in the SDK. This section will walk through how you can upgrade your SDK project to use the newest Alfresco version.

These instructions include information about how to upgrade projects generated from each one of the artifacts. Make sure you are following upgrade instructions for the correct "From version -> To version".

⚠ Make sure you have made a complete backup of your project before you start the upgrade process!

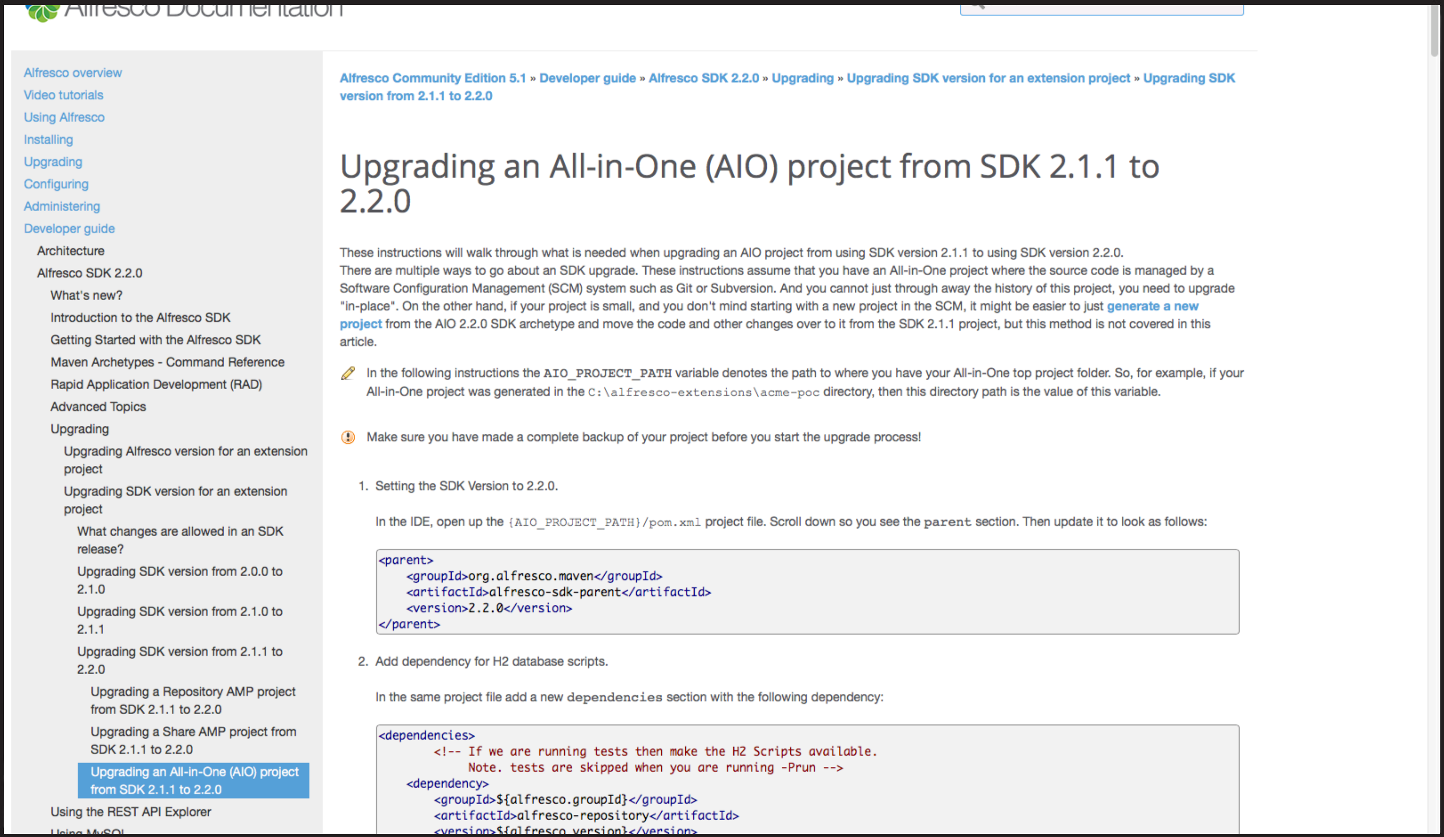
Upgrading SDK 2.1.1 projects from Enterprise 5.0.1 to 5.0.2
These instructions will walk through what is needed when upgrading an SDK 2.1.1 project from using Alfresco Enterprise version 5.0.1 to using Enterprise version 5.0.2.

Upgrading SDK 2.1.1 projects from Enterprise 5.0.1 (or 5.0.2) to 5.0.3
These instructions will walk through what is needed when upgrading an SDK 2.1.1 project from using Alfresco Enterprise version 5.0.1 (or 5.0.2) to using Enterprise version 5.0.3. There are a couple of changes in the 5.0.3 code base, such as new database scripts, that are not automatically handled by SDK 2.1.1. There is also a problem running with Spring Loaded because of some new security checks in the repository layer.

Upgrading SDK 2.2.0 projects from Community 5.1.e to 5.1.[f|g]
These instructions will walk through what is needed when upgrading an SDK 2.2.0 project from using Alfresco Community version 5.1.e to using Community version 5.1.f. These instructions also apply when upgrading to version 5.1.g.

Upgrading SDK 2.2.0 projects from Community 5.1.e to 5.2.a
These instructions will walk through what is needed when upgrading an SDK 2.2.0 project from using Alfresco Community version 5.1.e to using Community version 5.2.a. In fact, we are upgrading to Alfresco Platform version 5.2.a and Alfresco Share version 5.1.g. Note that from now on the `alfresco.war` is no longer available using the `alfresco` artifactId. Now we have to use `alfresco-platform`.

Upgrading SDK 2.2.0 projects from Enterprise 5.1.0 to greater than 5.1.0
These instructions will walk through what is needed when upgrading an SDK 2.2.0 project from using Alfresco Enterprise version 5.1.0 to using Enterprise version 5.1.1. These instructions also apply when upgrading to other versions greater than 5.1.0, such as 5.1.0.5.


Alfresco Documentation
Alfresco overview
Video tutorials
Using Alfresco
Installing
Upgrading
Configuring
Administering
Developer guide
Architecture
Alfresco SDK 2.2.0
What's new?
Introduction to the Alfresco SDK
Getting Started with the Alfresco SDK
Maven Archetypes - Command Reference
Rapid Application Development (RAD)
Advanced Topics
Upgrading
Upgrading Alfresco version for an extension project
Upgrading SDK version for an extension project
What changes are allowed in an SDK release?
Upgrading SDK version from 2.0.0 to 2.1.0
Upgrading SDK version from 2.1.0 to 2.1.1
Upgrading SDK version from 2.1.1 to 2.2.0
Upgrading a Repository AMP project from SDK 2.1.1 to 2.2.0
Upgrading a Share AMP project from SDK 2.1.1 to 2.2.0
Upgrading an All-in-One (AIO) project from SDK 2.1.1 to 2.2.0
Using the REST API Explorer
Using MySQL

Alfresco Community Edition 5.1 » Developer guide » Alfresco SDK 2.2.0 » Upgrading » Upgrading SDK version for an extension project » Upgrading SDK version from 2.1.1 to 2.2.0

Upgrading an All-in-One (AIO) project from SDK 2.1.1 to 2.2.0

These instructions will walk through what is needed when upgrading an AIO project from using SDK version 2.1.1 to using SDK version 2.2.0. There are multiple ways to go about an SDK upgrade. These instructions assume that you have an All-in-One project where the source code is managed by a Software Configuration Management (SCM) system such as Git or Subversion. And you cannot just through away the history of this project, you need to upgrade "in-place". On the other hand, if your project is small, and you don't mind starting with a new project in the SCM, it might be easier to just generate a new project from the AIO 2.2.0 SDK archetype and move the code and other changes over to it from the SDK 2.1.1 project, but this method is not covered in this article.

In the following instructions the `AIO_PROJECT_PATH` variable denotes the path to where you have your All-in-One top project folder. So, for example, if your All-in-One project was generated in the `C:\alfresco-extensions\acme-poc` directory, then this directory path is the value of this variable.

Make sure you have made a complete backup of your project before you start the upgrade process!

1. Setting the SDK Version to 2.2.0.

   In the IDE, open up the `{AIO_PROJECT_PATH}/pom.xml` project file. Scroll down so you see the `parent` section. Then update it to look as follows:

```
<parent>
    <groupId>org.alfresco.maven</groupId>
    <artifactId>alfresco-sdk-parent</artifactId>
    <version>2.2.0</version>
</parent>
```

2. Add dependency for H2 database scripts.

   In the same project file add a new `dependencies` section with the following dependency:

```
<dependencies>
        <!-- If we are running tests then make the H2 Scripts available.
             Note. tests are skipped when you are running -Prun -->
    <dependency>
        <groupId>${alfresco.groupId}</groupId>
        <artifactId>alfresco-repository</artifactId>
        <version>${alfresco.version}</version>
```

make business flow

What’s new in Alfresco SDK 3.0

Alfresco

Alfresco SDK 3.0 Vision

Compatibility

- 4.2.x → 5.2.x
- Easy upgrades

Align to standards

- Revamp archetypes
- JAR Support

Empower Developers

- Hot reloading
- Integration testing

Configurable

- AMPs still supported as an assembly
- Full control over the Alfresco Maven Plugin

Compatibility

- SDK 3.0 is compatible with:
  - Alfresco One 4.2.x
  - Alfresco One 5.0.x
  - Alfresco One 5.1.x
  - Alfresco Content Services 5.2.x
    (and all the community releases)

- Switching between versions is easy:
  - <alfresco.platform.version />
  - <alfresco.share.version />

- Upgrades are easy is one line:
  - <alfresco.sdk.version />

Align to standards

- All SDK 3.0 projects will produce a JAR
- Archetypes has been revamped to align with a standard Maven JAR directory structure
- No forced parent pom
- All-In-One reduced to three modules
  - Platform JAR
  - Share JAR
  - Integration Tests
    (No more war or runner modules)
- All "run" logic moved to Alfresco Maven Plugin, no more profiles

Empowering Developers

- Hot Reloading via HotSwap Agent (DCEVM) or JRebel

- Improved Integration Testing
  - Execute integration tests against the full stack without having to start up the context
  - If the stack is not running it will spin it up
  - Integrates well with IDE’s, almost instant results when executing tests

Configurable

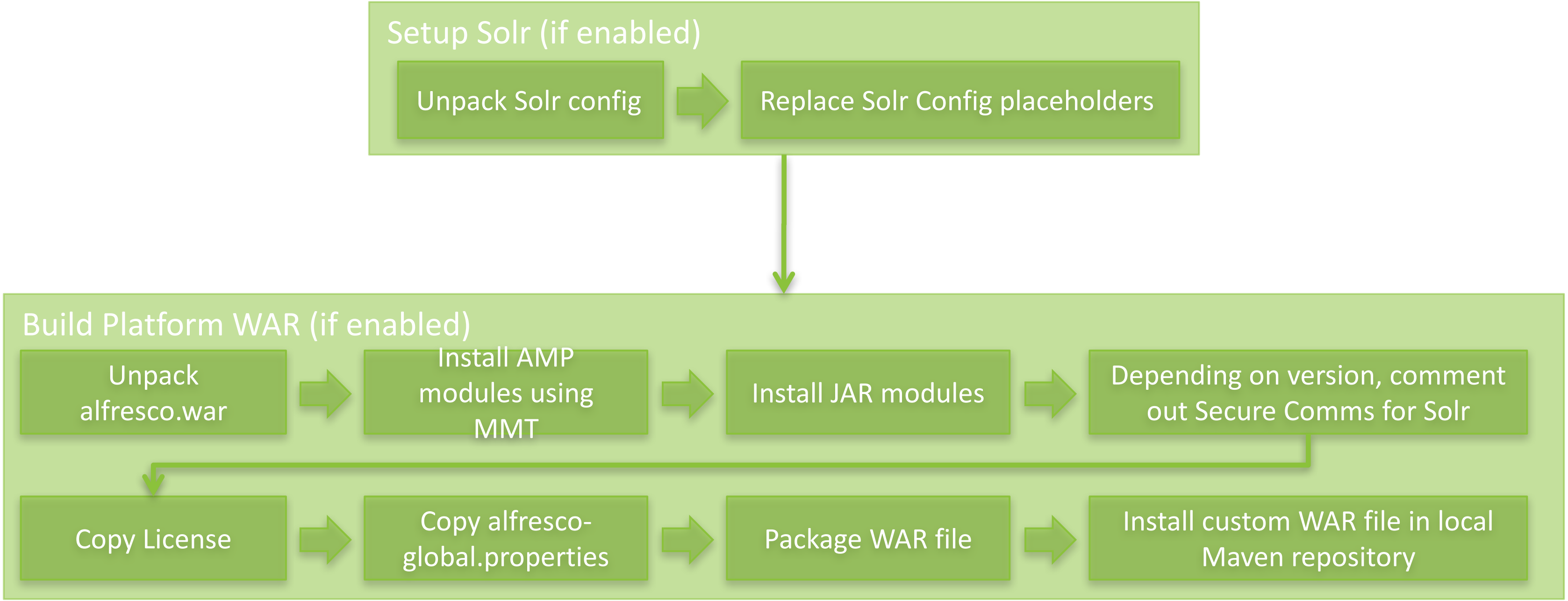
- Alfresco Maven Plugin has 40 different configuration parameters, e.g:
  - <enablePlatform />, <enableShare />, <enableSolr/>, <enableMySQL />, <platformModules />, <shareModules />
- AMPs are still supported as an assembly, if you need to bring in 3rd party dependencies or override OOTB files

Alfresco Maven Plugin

Alfresco Maven Plugin

- The Alfresco Maven Plugin handles running and testing your modules
- Install 3rd party dependencies (both AMPs and JARs)
- Start the Platform, Solr, Share, API Explorer *(Also allows you to deploy a custom WAR)*
- Can be configured to a lot of common use cases – over 40 different configuration parameters
- Executed with "mvn install alfresco:run"

Alfresco Maven Plugin – “Run (alfresco:run)”



Setup Solr (if enabled)

Unpack Solr config

Replace Solr Config placeholders

Build Platform WAR (if enabled)

Unpack alfresco.war

Install AMP modules using MMT

Install JAR modules

Depending on version, comment out Secure Comms for Solr

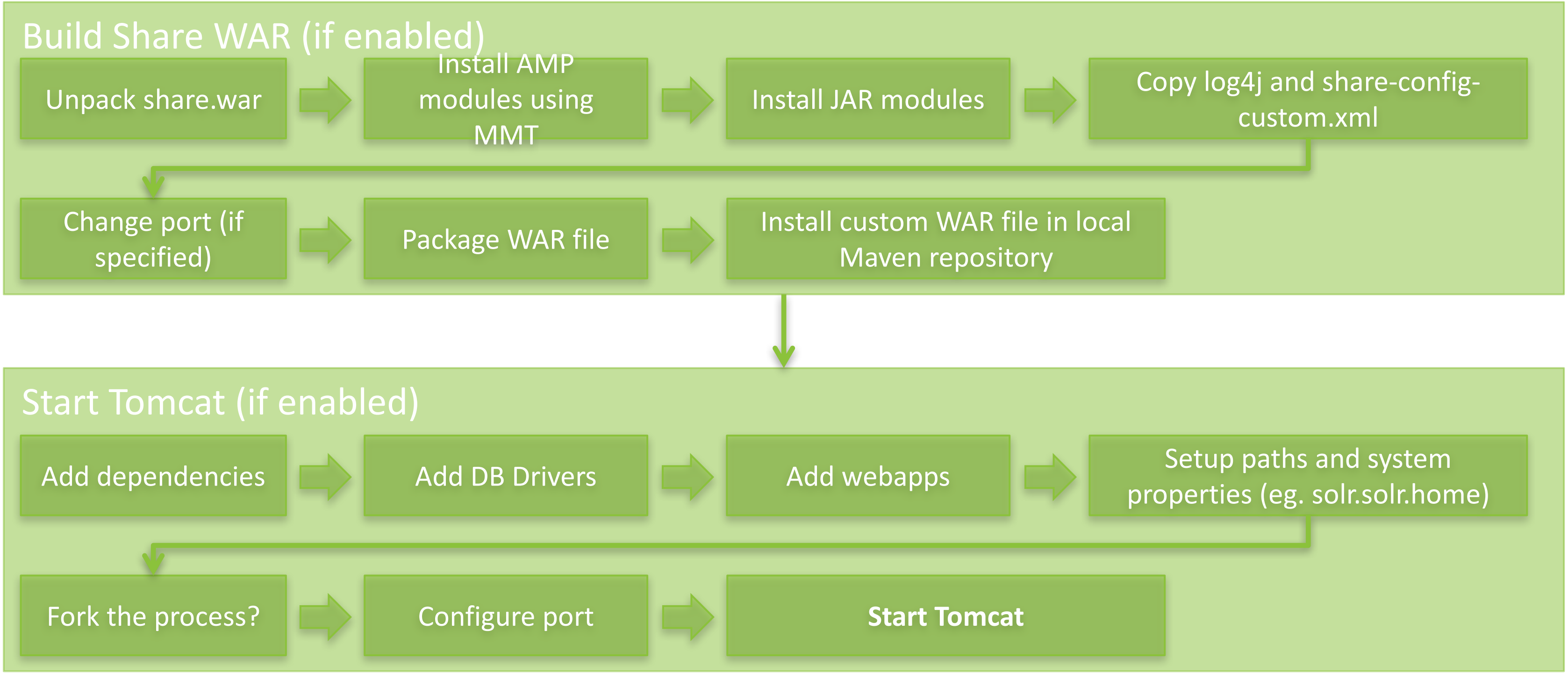
Copy License

Copy alfresco-global.properties

Package WAR file

Install custom WAR file in local Maven repository

Alfresco Maven Plugin – “Run (alfresco:run)”



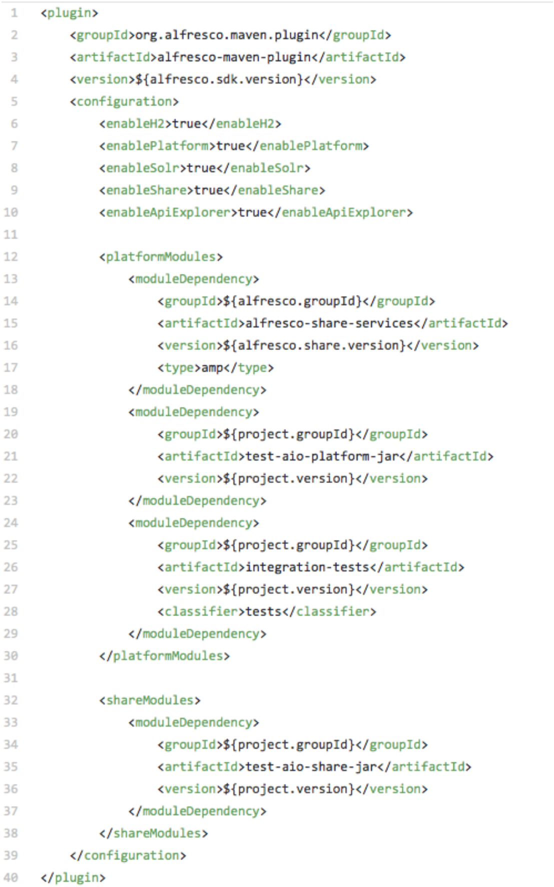
Build Share WAR (if enabled)

- Unpack share.war
- Install AMP modules using MMT
- Install JAR modules
- Copy log4j and share-config-custom.xml
- Change port (if specified)
- Package WAR file
- Install custom WAR file in local Maven repository

Start Tomcat (if enabled)

- Add dependencies
- Add DB Drivers
- Add webapps
- Setup paths and system properties (eg. solr.solr.home)
- Fork the process?
- Configure port
- **Start Tomcat**

Alfresco Maven Plugin Sample Configuration



```xml
<plugin>
    <groupId>org.alfresco.maven.plugin</groupId>
    <artifactId>alfresco-maven-plugin</artifactId>
    <version>${alfresco.sdk.version}</version>
    <configuration>
        <enableH2>true</enableH2>
        <enablePlatform>true</enablePlatform>
        <enableSolr>true</enableSolr>
        <enableShare>true</enableShare>
        <enableApiExplorer>true</enableApiExplorer>

        <platformModules>
            <moduleDependency>
                <groupId>${alfresco.groupId}</groupId>
                <artifactId>alfresco-share-services</artifactId>
                <version>${alfresco.share.version}</version>
                <type>amp</type>
            </moduleDependency>
            <moduleDependency>
                <groupId>${project.groupId}</groupId>
                <artifactId>test-aio-platform-jar</artifactId>
                <version>${project.version}</version>
            </moduleDependency>
            <moduleDependency>
                <groupId>${project.groupId}</groupId>
                <artifactId>integration-tests</artifactId>
                <version>${project.version}</version>
                <classifier>tests</classifier>
            </moduleDependency>
        </platformModules>

        <shareModules>
            <moduleDependency>
                <groupId>${project.groupId}</groupId>
                <artifactId>test-aio-share-jar</artifactId>
                <version>${project.version}</version>
            </moduleDependency>
        </shareModules>
    </configuration>
</plugin>
```

Demo

Questions?

Alfresco®

</presentation>

Alfresco®