

# Local plugin library isolation – vSphere 8.0

Third-party libraries deployed and utilized by the vCenter Server appliance (VCSA) for its own needs that are currently exposed to partner plugins will be restricted and no longer available, effective with vSphere 8.0.

In the versions of vSphere up to 7.0, local plugins can import packages from third-party libraries deployed for the needs of the vSphere Client platform. This is problematic for multiple reasons, such as:

- Changes to internal vSphere Client APIs could break plugin compatibility.
- Changes to a particular vSphere Client dependency (e.g. to consume security updates) could impact plugin compatibility.
- Unnecessary 3<sup>rd</sup>-party dependencies decrease performance of the vSphere Client. (Most plugins need only libraries related to the Spring framework.)
- Partner plugins can fail when VMware updates third-party libraries in response to security vulnerabilities.

For these reasons, VMware will require plugin developers to supply their own OSGi Java dependencies. Any Spring Framework, JSON/XML serialization, Apache utilities, etc. will have to be provided as part of the plugin: either as separate OSGi bundles or included in an existing plugin bundle's class path.

The library isolation will be in effect from vSphere 8.0.

Following are detailed guides on how to comply with the new OSGi requirements. A local setup refers to when a developer runs the vSphere Client process on their developer machine in contrast to CloudVM setup, where the vSphere Client process runs inside the vCenter Server Appliance.

## Step-by-step guide on how to move to full isolated plugin (local setup)

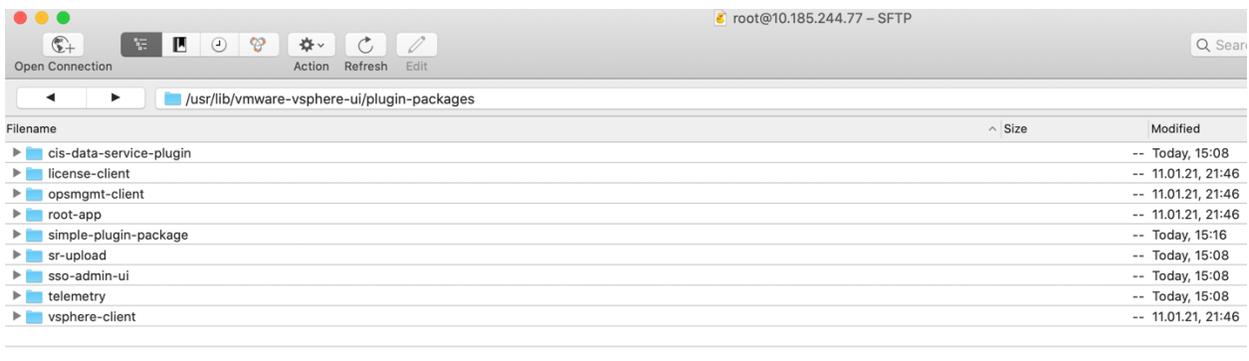
### Prerequisites:

- zip edit tool
- telnet client
- registered plugin
- knowledge about using the OSGi Console: see Using the OSGi console section here: <https://www.vogella.com/tutorials/OSGi/article.html#access-to-the-osgi-console-for-your-eclipse-application>

### Steps:

1. Place `osgi.fullIsolation.debug=true` in `webclient.properties`





2. Make sure that the files of your plugin are owned by the vsphere-ui OS user and the directories are secured as shown in the following illustration.

```
cd /usr/lib/vmware-vmware-UI/plugin-packages/
```

```
chown vsphere-ui simple-plugin-package/ -R
```

The resulting file permissions and ownership should be as follows:

```
root@sc2-10-185-244-77 [ /usr/lib/vmware-vmware-UI/plugin-packages ]# ls -lah
total 44K
drwxrwx--- 11 vsphere-ui root 4.0K Jan 15 13:16 .
drwxrwx---  7 vsphere-ui cis  4.0K Jan 15 13:27 ..
drwxrwx---  3 vsphere-ui root 4.0K Jan 15 13:26 cis-data-service-plugin
drwxrwx---  3 vsphere-ui root 4.0K Jan 11 19:46 license-client
drwxrwx---  3 vsphere-ui root 4.0K Jan 11 19:46 opsmgmt-client
drwxrwx---  3 vsphere-ui root 4.0K Jan 11 19:46 root-app
drwxr-xr-x  3 vsphere-ui root 4.0K Jan 15 13:26 simple-plugin-package
drwxrwx---  3 vsphere-ui root 4.0K Jan 15 13:26 sr-upload
drwxrwx---  3 vsphere-ui root 4.0K Jan 15 13:26 sso-admin-ui
drwxrwx---  3 vsphere-ui root 4.0K Jan 15 13:26 telemetry
drwxrwx---  3 vsphere-ui root 4.0K Jan 11 19:46 vsphere-client
```

3. Add `osgi.fullIsolation.debug=true` in `webclient.properties`

```
vi /etc/vmware/vsphere-ui/webclient.properties
```

```
live.updates.navtree.enabled = true
live.updates.lists.enabled = true
live.updates.objectstate.enabled = true

osgi.fullIsolation.debug=true

# Web client session timeout in minutes, default is 120, i.e. 2 hours
```

4. Enable the OSGi Console, as follows:

4.1: **cd /usr/lib/vmware-vmphere-ui/server/webapps**

4.2: **unzip -j "h5-bridge-webapp.war" "WEB-INF/web.xml" -d "/usr/lib/vmware-vmphere-ui/server/webapps/WEB-INF"** (this will unzip only the /WEB-INF/web.xml file)

4.3. Edit the WEB-INF/web.xml by removing the the XML comments around the init-param "commandline"

**vi WEB-INF/web.xml**

The web.xml should look like this:

```
<servlet id="bridge">
  <servlet-name>equinoxbridgeservlet</servlet-name>
  <servlet-class>com.vmware.vsphere.bridge.BridgeServletEx</servlet-class>
  <!-- Telnet to port 2401 for osgi console -->

  <init-param>
    <param-name>commandline</param-name>
    <param-value>-console 2401</param-value>
  </init-param>
```

Save the file.

4.4. Modify the h5-bridge-webapp.war with the modified web.xml

**zip -u h5-bridge-webapp.war WEB-INF/web.xml**

```
root@sc2-10-185-244-77 [ /usr/lib/vmware-vmphere-ui/server/webapps ]# zip -u h5-bridge-webapp.war WEB-INF/web.xml
updating: WEB-INF/web.xml (deflated 76%)
```

Be sure that the output of the zip command is "updating", not adding.

4.5. Make sure that the files of the archive are owned by the vmphere-ui OS user

**chown vmphere-ui h5-bridge-webapp.war**

Check filesystem permissions, using **ls -lah**

```

root@sc1-10-168-218-96 [ /usr/lib/vmware-vmphere-ui/server/webapps ]# chown vsphere-ui h5-bridge-webapp.war
root@sc1-10-168-218-96 [ /usr/lib/vmware-vmphere-ui/server/webapps ]# ls -lah
total 98M
drwxr-xr-x  3 root      root 4.0K Jun 10 10:12 .
drwxr-xr-x 10 vsphere-ui root 4.0K Jun 10 08:22 ..
-rwxr-xr-x  1 vsphere-ui root  98M Jun 10 10:12 h5-bridge-webapp.war

```

5. Open ports on the VCSA:

```
iptables -A INPUT --dport 2401 -j ACCEPT
```

6. Restart the vsphere-ui:

```
service-control --restart vsphere-ui
```

7. Telnet to <VCSA IP/hostname> port 2401

```
telnet <vc_ip> 2401
```

8. Use the **ss** command to list all bundles and look for the bundles belonging to your plugin. (You can make them out by their symbolic names)

```

272    ACTIVE    org.osgi.service.subsystem.region.context.1_1.0.0
273    RESOLVED   com.vmware.o6jia.simple-lib_6.1.0.SNAPSHOT
274    INSTALLED  com.vmware.o6jia.spring-xml-consumer_7.0.2.00000
275    INSTALLED  org.eclipse.virgo.web.dm_1.0.0
276    INSTALLED  com.vmware.o6jia.simple-front-end_7.0.2.00000

```

We can observe that the bundles of your plugin are in the INSTALLED state, meaning they weren't resolved and could not be started.

9. Use **diag <bundle id>** to diagnose missing dependencies

```

g! diag 274
com.vmware.o6jia.spring-xml-consumer [274]
  Unresolved requirement: Import-Package: org.aopalliance.aop
  Unresolved requirement: Import-Package: com.vmware.o6jia.context
    -> Export-Package: com.vmware.o6jia.context; bundle-symbolic-name="org.eclipse.virgo.web.dm"; bundle-version="1.0.0"; version="0.0.0"
    org.eclipse.virgo.web.dm [275]
      Unresolved requirement: Import-Package: org.springframework.aop

```

The command will list all unresolved Import-Package/Require-Bundle requirements.

10. If you're not sure which bundle/jar provides a package that your plugin requires you can use the following OSGi console command:

```
packages <required package>
```

```
g! packages com.fasterxml.jackson.core
osgi.wiring.package; bundle-symbolic-name="com.fasterxml.jackson.core.jackson-core"; bundle-version:Version="2.13.2";
```

Typically, the bundle-symbolic-name is going to be a good orientation point to bundle that provides a required package. Use the bundle-symbolic-name to search maven.org for a matching jar name. For instance, if we make a search for "jackson-core", we'll likely hit the jar that is required. <https://search.maven.org/search?q=jackson-core>

## 11. Add missing OSGi dependencies

**Dependencies should be placed in a new folder "libraries", a sibling to the "plugins" folder. These dependencies in the "libraries" folder are going to be skipped in 6.7 and 7.0, but picked up in 8.0, making it possible to run the same plugin release in 6.7, 7.0 and 8.0 without any changes to the plugin.**

A note about Spring MVC:

If you use Spring MVC controllers in your plugin the following list provides a latest picture (at time of writing) of the bundles required to resolve Spring MVC dependencies:

- org.apache.servicemix.bundles.aopalliance-1.0\_6
- org.apache.servicemix.bundles.spring-aop-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-beans-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-context-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-context-support-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-core-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-expression-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-web-5.2.22.RELEASE\_1
- org.apache.servicemix.bundles.spring-webmvc-5.2.22.RELEASE\_1

**The "plugins" and "libraries" folders will look like:**



12. Restart vsphere-ui, so we can make another iteration and see any other missing dependencies

**service-control --restart vsphere-ui**

13. Telnet will disconnect. Connect again to the OSGi console:

**telnet <vc\_ip> 2401**

This time our plugin looks like the following:

```

272  ACTIVE      org.osgi.service.subsystem.region.context.1_1.0.0
273  RESOLVED    org.apache.servicemix.bundles.spring-core_5.2.6.RELEASE_1
274  RESOLVED    org.apache.servicemix.bundles.spring-context-support_5.2.6.RELEASE_1
275  RESOLVED    com.vmware.o6jia.simple-lib_6.1.0.SNAPSHOT
276  RESOLVED    com.vmware.o6jia.spring-xml-consumer_7.0.2.00000
277  INSTALLED   com.vmware.o6jia.simple-front-end_7.0.2.00000
278  RESOLVED    org.eclipse.virgo.web.dm_1.0.0
279  RESOLVED    org.apache.servicemix.bundles.aopalliance_1.0.0.6
280  RESOLVED    org.apache.servicemix.bundles.spring-web_5.2.6.RELEASE_1
281  RESOLVED    org.apache.servicemix.bundles.spring-beans_5.2.6.RELEASE_1
282  RESOLVED    org.apache.servicemix.bundles.spring-context_5.2.6.RELEASE_1
283  RESOLVED    org.apache.servicemix.bundles.spring-expression_5.2.6.RELEASE_1
284  RESOLVED    org.apache.servicemix.bundles.spring-aop_5.2.6.RELEASE_1
285  RESOLVED    org.apache.servicemix.bundles.spring-webmvc_5.2.6.RELEASE_1

```

We can observe that only one war is not resolved. We'll repeat steps 9-12

14. Use **diag <bundle id>** to diagnose remaining missing dependencies

```
g! diag 276
com.vmware.o6jia.simple-front-end [276]
  Unresolved requirement: Import-Package: com.fasterxml.jackson.core
```

Note: that the id of the simple-front-end bundle is now different after the last restart of the vsphere-ui. That is due to the random nature in which plugin bundles are installed in the OSGi runtime.

16. Continue adding required bundles in your "libraries" directory until all your plugin bundles are resolved.

17. Make the necessary changes to your build to include the OSGi bundles identified as missing dependencies in the "libraries" folder of your plugin package.

---

### Note 1:

#### Troubleshooting

If runtime requests to your plugin endpoints fail with: **"org.springframework.http.converter.HttpMessageNotWritableException: No converter found for return value of type: class XXX"**, then your plugin code is probably expecting HTTP responses with Content-Type: application/json. Spring Framework supports application/json serialization if you include the following jars in your **libraries** folder:

<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations/2.13.2>  
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind/2.13.2>  
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations/2.13.2>

These are the latest versions at the time of writing of this document. Always use the latest versions to consume the latest security fixes.

### Note 2:

Some jars posted in the Maven Central repository are not OSGi bundles. If you happen to require such a jar, use Spring's bundlor or BND to convert the jar to an OSGi bundle:

<https://docs.spring.io/s2-bundlor/1.0.x/user-guide/htmlsingle/user-guide.html>  
<https://github.com/bndtools/bnd/tree/master/maven/bnd-maven-plugin>

Some jars have different artifact IDs to denote that they are OSGi bundles. For instance, Apache's http client:

<https://search.maven.org/artifact/org.apache.httpcomponents/httpclient-osgi>

is an OSGI bundle, but

<https://search.maven.org/artifact/org.apache.httpcomponents/httpclient>

is not an OSGi bundle.

**Note 3:**

If your plugins require the following packages:

com.vmware.vise.messaging.endpoints

com.vmware.vise.messaging.remoting

you should remove them. Those packages are not exported in full isolation mode and are no longer needed.

**Note 4:**

Apache ServiceMix provides OSGi versions of Spring and other popular libraries. See:

<https://mvnrepository.com/artifact/org.apache.servicemix.bundles>