Customize Liberty

Contents

1 Customize Liberty

In this lab we will explorer the different options to customize a Liberty environment. Customization is often required as an application is moved between the different stages of a DevOps pipeline. At the completion of this lab, you will be able to make an informed decision about how to fit Liberty into your existing pipeline, or as part of a new infrastructure.

Prerequisite: Completion of Getting Started lab section 3, using the command line.

Please refer to the following table for file and resource location references on different operating systems.

Location Ref.	OS	Absolute Path
{LAB_HOME}	Windows	`C:\WLP_<version>`
	Linux	`~/WLP_<version>` `Or your choice`
	Mac OSX	`~/WLP_<version>` `Or your choice`

2 Create the test server

Start by creating a server for the lab.

__1. Copy the customizeServer directory included with this lab to {LAB_HOME}/wlp/usr/servers directory. After copy, you should see {LAB_HOME}/wlp/usr/servers/customizeServer directory.

__2. Change directory to {LAB_HOME}/wlp/bin

__3. Start the server

server start customizeServer

__4. Ensure server is running correctly by pointing your browser to **http://localhost:9580/Sample1/SimpleServlet**

3 Packaging Liberty server

There are multiple ways you can package a liberty server:

- As a zip file
- As a runnable jar
- As a container image. This is beyond the scope of this lab.

3.1.1 Packaging Liberty server as a zip file

You can package the liberty server as a zip file to be unzipped and run in a destination environment.

__1. Stop the server if it is still running

server stop customizeServer

__2. Run the server package command.

server package customizeServer --include=minify

__3. The minify option creates the minimum zip file that can be used to run your server. Navigate to customizeServer directory and check the size of “customizeServer.zip”

__4. unzip “customizeServer.zip” to a temporary directory. We’ll refer to it as <DEST>. Optionally you can unzip it on a different machine.

__5. Change directory to the destination liberty environment:

cd <DEST>/wlp/bin

__6. Start the liberty server in the destination environment

server start customizeServer

__7. Point your browser to the server (http://localhost:9580/Sample1/SimpleServlet) and check the logs to ensure it runs as expected.

__8. Stop the server

server stop customizeServer

3.1.2 Packaging Liberty server into a runnable jar

To create a runnable jar:

__1. Stop the server if it is still running

server stop customizeServer

__2. Run the server package command.

server package customizeServer --include=minify,runnable

__3. The runnable option creates a runnable jar. Navigate to myServer directory and check the size of “customizeServer.jar”

__4. Copy “customizeServer.jar” to a temporary directory. We’ll refer to it as <DEST>. Optionally you can unzip it on a different machine.

__5. Change directory to the destination liberty environment:

cd <DEST>

__6. Start the liberty server in the destination environment

java -jar customizeServer.jar

__7. Point your browser to the server (http://localhost:9580/Sample1/SimpleServlet) and verify it still works.

__8. Stop the server with ctrl-<C>

__9. When you run “java -jar myServer.jar”, the contents of the jar is first extracted to a default location. The output of the server is also directed to the default location. If you wish to change the default, set the WLP_OUTPT_DIR environment variable:

__a. For windows: “**set WLP_OUTPUT_DIR=<DEST>**”

__b. For Linux: “**export WLP_OUTPUT_DIR=<DEST>**”

__10. Re-run the “**java -jar customizeServer.jar**” command and verify that the logs and workarea directories are now in the <DEST>/customizeServer directory.

3.2 Overriding Configurations

You can use the configDropins directory to provide default configurations, or to override configurations as you pass the Liberty server to different environments from development to production. Default configurations are provided in the <SERVER>/configDropins/defaults directory. Overrides are provided in the <SERVER>/configDropins/overrides directory. For this lab, we will only experiment with overriding configuration.

3.2.1 Overriding Liberty configuration

__1. Stop the server if it is still running

server stop customizeServer

__2. Change directory to {LAB_HOME}/wlp/usr/servers/customizeServer directory

__3. Examine the contents of server.xml and bootstrap.properties. In server.xml, two variables customize.http.port and customize.https.port are used:

```
<httpEndpoint id="defaultHttpEndpoint"
httpPort="${customize.http.port}" httpsPort="${customize.https.port}"
/>
```

In bootstrap.properties, the values for these variables are defined. Note that the http port is defind to be 9580.

```
customize.http.port=9580
customize.https.port=9581
```

__4. Create a new directory named configDropins

mkdir configDropins

__5. Change directory to configDropins

__6. Create a new directory named overrides

mkdir overrides

__7. Copy the file overrides.xml from the lab's directory to the overrides directory.

__8. Start the server

server start customizeServer

__9. Point the browser to the original URL (http://localhost:9580/Sample1/SimpleServlet) and check it no longer works.

__10. Point the browser to the new URL ((http://localhost:9080/Sample1/SimpleServlet) and check it works.

__11. Examine overrides.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="Liberty server for labs">

    <!-- variables for port numbers -->
    <variable name="customize.http.port" value="9080" />
    <variable name="customize.https.port" value="9443" />

    <!-- Disable reading configuration after server starts -->
    <config updateTrigger="disabled"/>
</server>
```

Note that:

__a. It changes the values of the variables customize.http.port and customize.https.port

__b. It disables configuration update trigger so that configuration updates are no longer read by the server automatically. From here on, you need to restart the server after each configuration change.

3.2.2 Variable Substitution Precedence and environment variables

We have seen in the last section that variable definitions in server.xml override the values in bootstrap.properties. The precedence for variable substitution in Liberty is as follows:

- Server configuration files
- Java system properties
- bootstrap.properties
- Environment variables

If you are working in a containerized environment, it is not uncommon for some variables to be resolved through environment variables. There are two ways to ensure environment variables are used:

- Do not define values for these variables somewhere else, as environment variable take the least precedence.

- Use ${env.<variable name>} in the configuration.

Let's explore these options.

__1. Stop the server if it is still running

server stop customizeServer

__2. Edit overrides.xml, remove the following line, and save the changes.

<variable name=”customize.http.port” value=”9080” />

__3. Delete the following line from bootstrap.properties and save the changes:

customize.http.port=9580

__4. In the window where you start the Liberty server, set the following environment variable:

__a. On windows: “**set CUSTOMIZE_HTTP_PORT=9087**”

__b. On Linux, “**export CUSTOMIZE_HTTP_PORT=9087**”

__5. Start the server

server start customizeServer

__6. Point the browser to new URL (http://localhost:9087/Sample1/SimpleServlet) and check it works.

__7. Stop the server

server stop customizeServer

__8. To ensure that the environment variable is always used, add the following line to overrides.xml:

<variable name="customize.http.port" value="${env.CUSTOMIZE_HTTP_PORT}" />

__9. Change the environment to specify a different port:

__a. On windows: “**set CUSTOMIZE_HTTP_PORT=9187**”

__b. On Linux, “**export CUSTOMIZE_HTTP_PORT=9187**”

__10. Start the server and ensure the application is reachable on the new port: http://localhost:9187/Sample1/SimpleServlet

3.2.3 Environment Specific Configuration

The Liberty server configuration provides an “**include**” element to reference external configuration files. Included files are considered external to the server, and not packaged when using the “server package” tool. This allows you to create a single package that will work in multiple environments. Examples of external configuration files include:

- Common configurations shared by multiple servers in one environment, e.g., staging or test environment.
- Security related configuration such as key/trust stores, and password encryption keys

For physical hardware or VMs, these files are deployed separately, and are assumed to be present before you start the Liberty server. For containerized environments, these files may be mounted during container startup.

__1. Stop the server if it is still running

server stop customizeServer

__2. Open an editor and create a new file outside of the {LAB_HOME} environment. For example, c:\external.xml. Add the following content to the file and save the changes:

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="Liberty server for labs">
    <variable name=”customize.http.port” value=”9180” />
</server>
```

__3. Edit overrides.xml

__a. Add the following line:

<include optional="false" location="c:/external.xml"/>

__b. Remove this line:

<variable name="customize.http.port" value="…" />

__4. Start the server and ensure the application is reachable on the new port: http://localhost:9180/Sample1/SimpleServlet

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES

Please Recycle

WebSphere® software