Course Exercises Guide

Administration of IBM DataPower Gateway V7.6

Course code WE761 / ZE761 ERC 1.0

IBM Training

February 2018 edition

Notices

This information was developed for products and services offered in the US.

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, MD-NC119
Armonk, NY 10504-1785
United States of America

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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.

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 actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Approach®	Bluemix®	DataPower®
DB™	DB2 Connect™	DB2®
Express®	IBM Business Partner®	IMS™
Notes®	Power®	Rational®
Tivoli®	WebSphere®	z/OS®

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Lenovo and ThinkPad are trademarks or registered trademarks of Lenovo in the United States, other countries, or both.

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

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

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

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

VMware is a registered trademark or trademark of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

CloudLayer® and SoftLayer® are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

This course includes the following exercises:

- Exercise 1, "Upgrading image firmware"
- Exercise 2, "Using the CLI and the XML Management Interface to manage DataPower appliances"
- Exercise 3, "Using the troubleshooting tools to debug errors"
- Exercise 4, "Securing connections with SSL"
- Exercise 5, "Logging to an external system"

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

Hint

If you are unable to complete an exercise, you can copy the model solution application from the **lab files** directory (`/home/student/labfiles/dp/Solutions`). See Appendix B, "Exercise solutions" for instructions on how to access the solution code and application for each exercise.

Important

Online course material updates might exist for this course. To check for updates, see the Instructor wiki at: http://ibm.biz/CloudEduCourses

Exercise 1. Upgrading image firmware

Estimated time

00:30

Overview

In this exercise, you upgrade the firmware level of the gateway by using the Blueprint Console capabilities. You also practice switching the installation image firmware to the other level on the gateway.

Objectives

After completing this exercise, you should be able to:

- Identify the current firmware level on the gateway
- Upgrade the firmware level on the gateway
- Switch the installation image between the current and the previous version of the firmware.

Introduction

In this exercise, you upgrade the firmware level of the gateway by using the Blueprint Console capabilities. You also practice switching the installation image back the firmware to the previous level.

Requirements

To complete this exercise, you need:

- Access to an individual IBM DataPower Gateway at a 7.6.0 firmware level later than the most current level
- Access to the most current level of firmware file in the `/home/localuser/firmware` directory

Attention

This exercise depends on the following student requirements:

- Each student has their own gateway.
- The student gateway must be an IBM DataPower Gateway (IDG) at the 7.6.0.0 or newer level, but not the current level.
- Having access to the current 7.6.0 firmware on the student image. The firmware file must be consistent with the actual gateway. The exercise steps mention file names and firmware levels that depend on the stated levels.

General exercise information

This section provides general information about the exercises in this course. Review this section before starting the exercises.

User accounts

An Ubuntu user ID was created for you. You use this ID to log on to the image.

- User ID: `localuser`
- Password: `passw0rd` (replace the o with a zero 0)

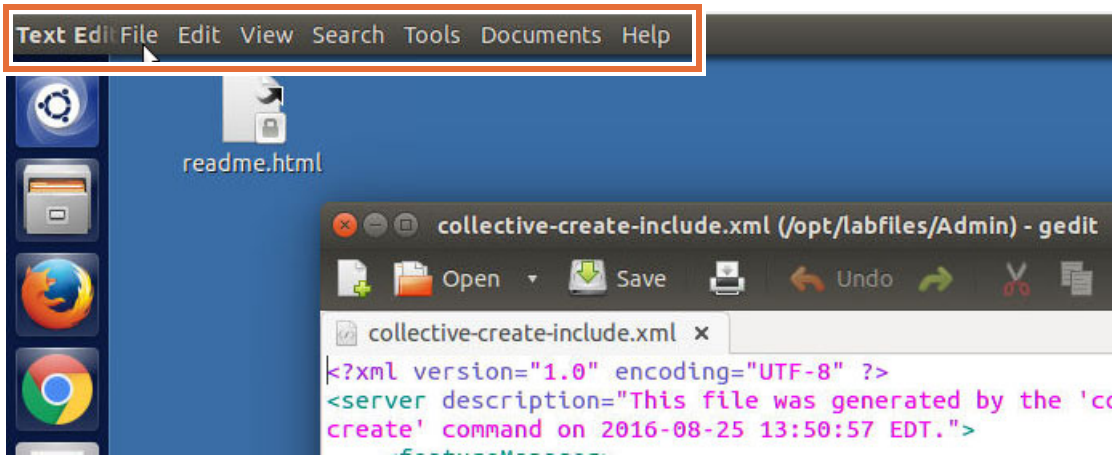
Information

The supplied course image is Ubuntu 14.04 LTS. The desktop uses Unity, which is different than the common Gnome desktop. Some hints on using Unity are at:

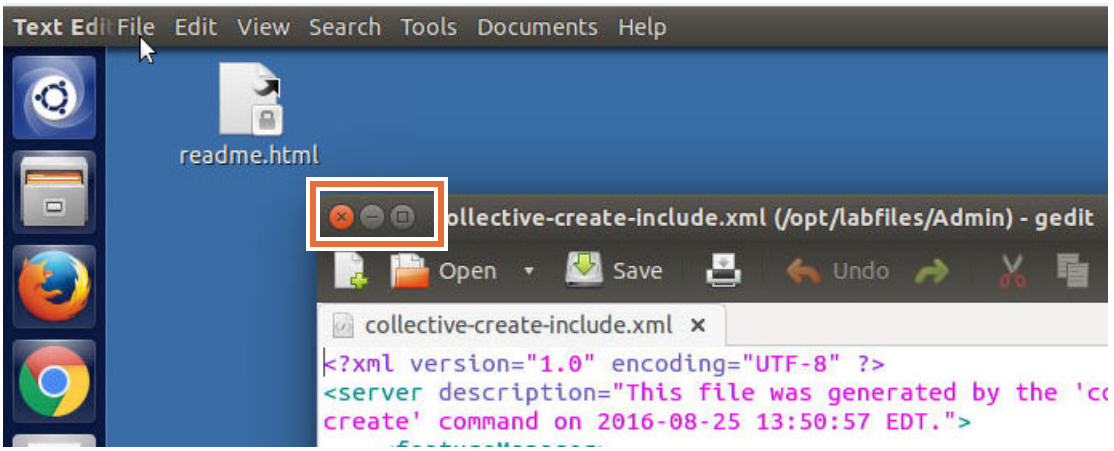
http://www.howtogeek.com/113330/how-to-master-ubuntus-unity-desktop-8-things-you-need-to-know/

Information

The desktop for Ubuntu is the Unity desktop. Unity uses a global menu. Which means application menus are not located in the window for the application. They are on the top pane. When a window is the active window, that window does not have any menu items, but the application type is displayed in the black bar that spans the top of the desktop. You cannot see the menu for the application until you hover your mouse over the top pane. When you hover your move over the black bar, the menu items for the active window are displayed.



The close, minimize, and full screen options are in the top corner of the application.



When you maximize the application window, the close, minimize, and full screen options also appear in the top pane.

Course labfiles

The exercises in this course use a set of lab files that might include scripts, applications, files, solution files, and others. The course lab files can be found in the following directory:

- `/home/localuser/labfiles/dp` for the Linux operating system

The exercises point you to the lab files as you need them.

Stop

Course updates and corrections

A course corrections document might be available for this course.

If you are taking the class with an instructor, the instructor can provide this document to you.

If your are taking the course in a self-paced environment, the course corrections document is provided with the other manuals.

To check whether a course corrections document exists for this course:

1. Go to the following URL: http://ibm.biz/CloudEduCourses
2. On the web page, locate, and click the **Course information** category.
3. Find your course in the list and click the link.
4. Click the **Attachments** tab to see whether a course corrections document exists with updated instructions.
5. To save the file to your computer, click the document link and follow the prompts.

Exercise instructions

Preface

- The references in exercise instructions use the following value:
  - `<dp_internal_ip>`: IP address of the DataPower gateway development and administrative functions
  - `<dp_WebGUI_port>`: Port number for the web management interface. Both the WebGUI and Blueprint Console use the same port. Default is **9090**
  - `<dp_admin_login>`: Secondary administrative account. The default for the IBM Remote Lab Platform (IRLP) is **sysadmin**
  - `<dp_admin_password>`: Password for secondary administrative account. The default for the IRLP is **sysadminpassw0rd**

1.1. Terminology

In this exercise, you work with two different levels of firmware. The version in `/home/localuser/firmware/` should be the most current level of 7.6.0 firmware. This level of firmware is referred to as the **current** level.

The version that is running on the gateway should be an older version of the 7.6.0 firmware. This level is referred to as the **current-1** level.

For the example in this exercise, and for the IRLP, the **current-1** version of the firmware on the gateway is 7.6.0.3. The **current** version that is in `/home/localuser/firmware/` is at the 7.6.0.5 level. The name of **current-1** does not imply that firmware updates must jump only one level. For example, the **current-1** version in this example is 7.6.0.3 and the **current** version is 7.6.0.5.

1.2. Determine the current firmware level

In this section, you determine the current firmware version on the gateway.

__ 1. Log on to the Blueprint Console by using the sysadmin account.

__ a. Access the gateway by opening the web browser and by using `https://<dp_internal_ip>:<dp_WebGUI_port>`

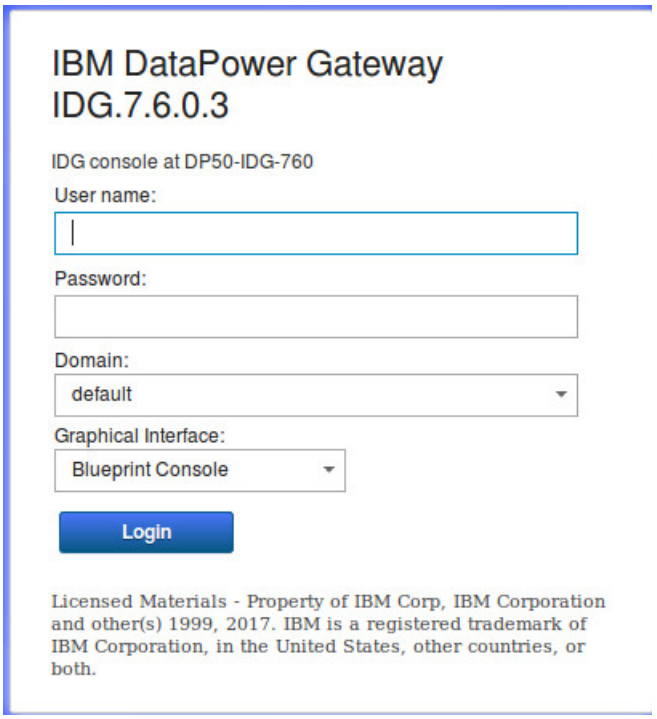
Information

A bookmark that is named DataPowerGateway is set on the Firefox browser that you can use to access the DataPower web management console. You might need to update the bookmark if the IP address for your DataPower is different.

__ b. The URL redirects to the initial value for the Blueprint Console: `https://<dp_internal_ip>:<dp_WebGUI_port>/dp/login.xml`. The first time that the browser connects to the DataPower gateway, a warning is received. Accept the certificate.

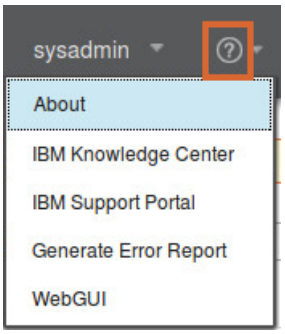
__ c. Enter the secondary administrative account and password: `<dp_admin_login>/<dp_admin_password>`

__ d. Set the **Graphical Interface** to **Blueprint Console**.



__ e. Click **Login**.

__ 2. After the login, check the firmware version in the Blueprint Console. It is located under the **Help** menu icon (circled question mark) on the menu bar in the **About** selection. This level is the **current-1** level. It should be 7.6.0.0 or newer.



Information

For other ways to determine the current firmware version. In the WebGUI, you can use **Control Panel (Open menu) > Status > System > Firmware Information**. In the CLI, you can use **show firmware-version**.

1.3. Where you download the firmware

Information

Since the firmware that you use in the exercise is already downloaded, this section provides you with the information on where you get the most recent firmware. You can read through this section without needing to perform the steps.
You DO NOT need to download the firmware from IBM Fix Central as described in this part to complete this exercise. The firmware is already downloaded onto the student image.

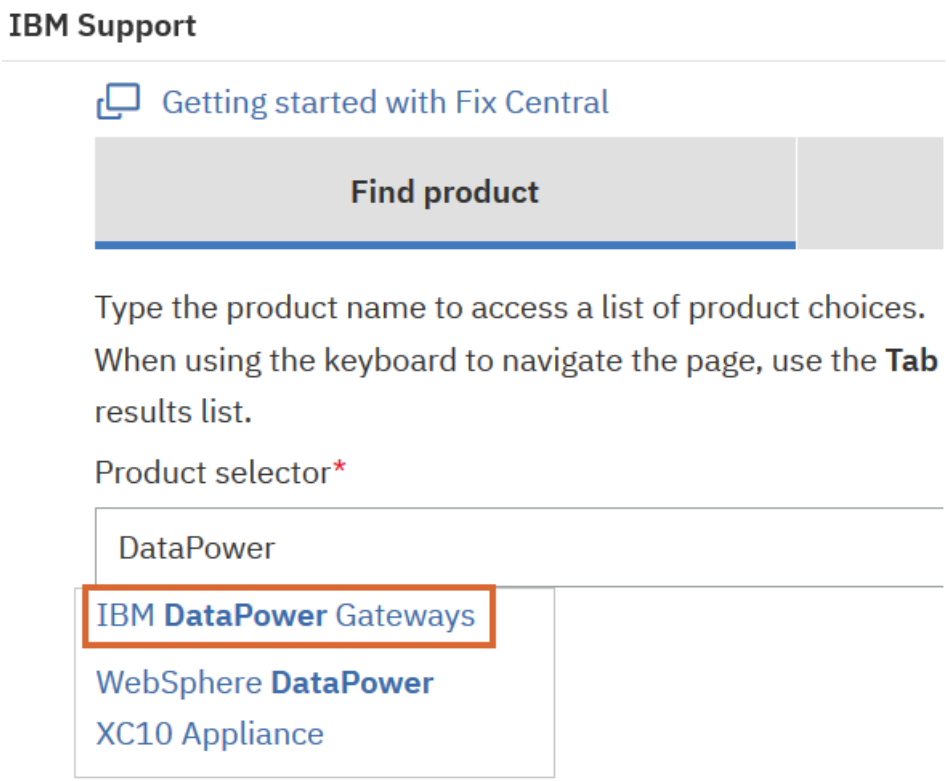
This section describes how the DataPower firmware can be downloaded from IBM Fix Central.

IBM Fix Central is the IBM support site that provides fixes and updates for your system's software, hardware, and operating system.

__ 1. Locate the fixes for your product.

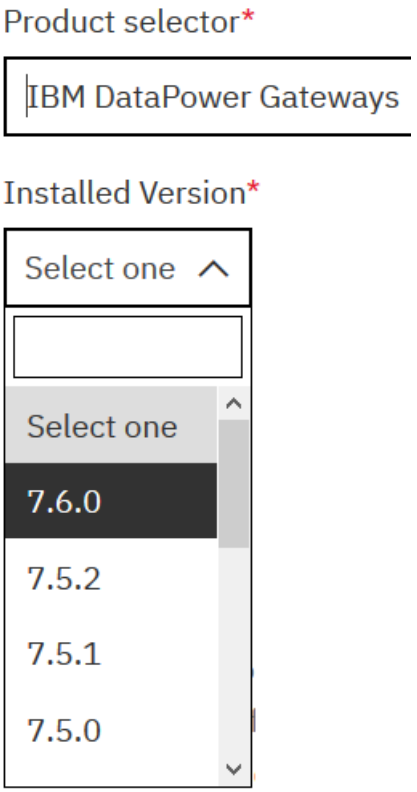
__ a. Open a browser and type `https://www.ibm.com/support/fixcentral/` in the address area.

__ b. Go to the Product selector field. Type `DataPower` in the selector field.

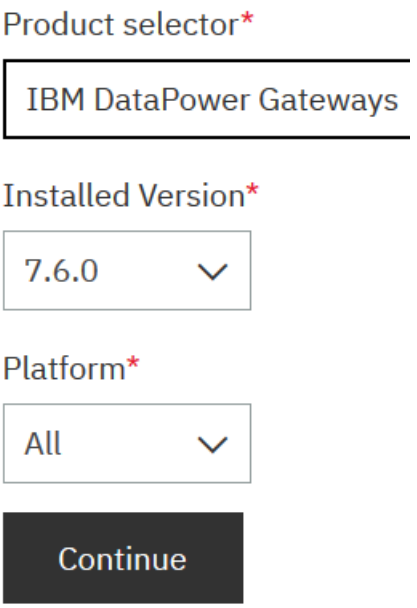


Then, select **IBM DataPower Gateways** from the list.

__ c. Next, you select the currently installed version from the list. Here the version is **7.6.0**.

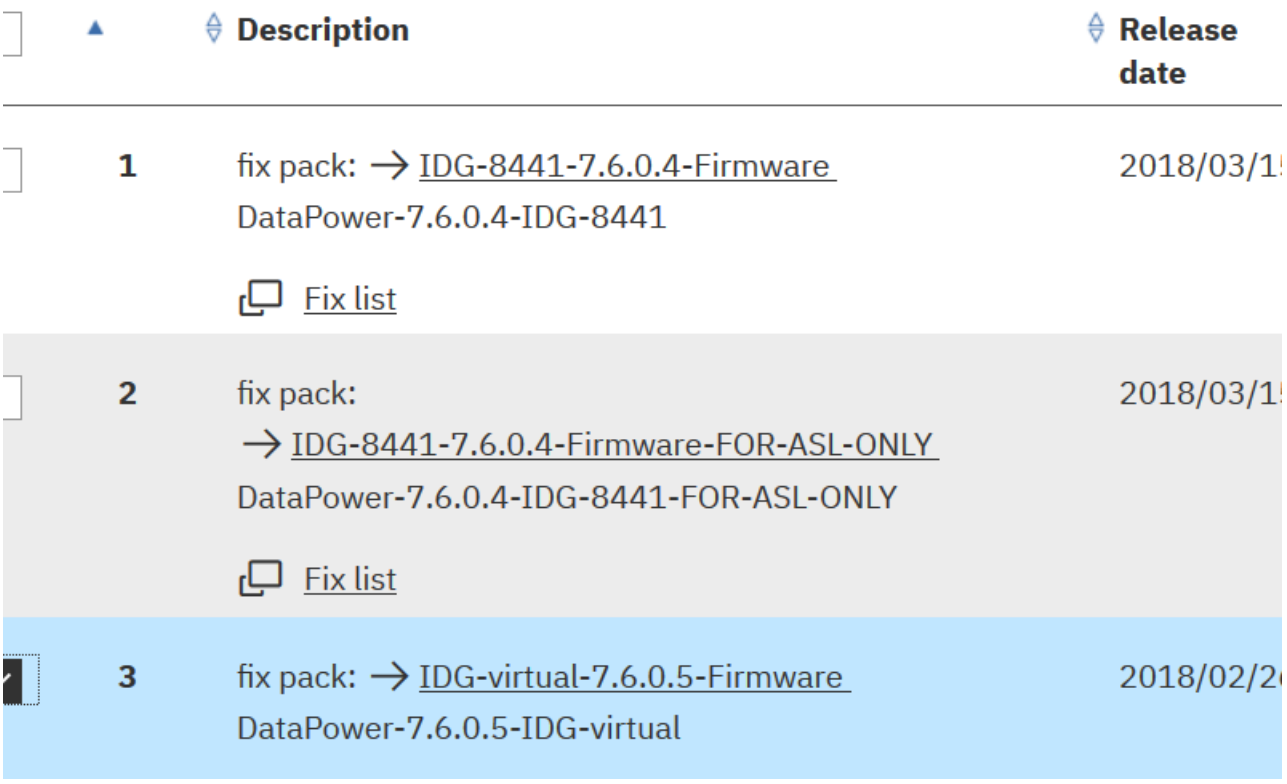


__ d. When you have selected the installed version, click **Continue**.



__ e. Select **Browse for fixes** and then, click **Continue**.

__ f. A page with a title “Select fixes” is displayed. Since you are working with a virtual appliance in the exercises, you select the IDG-virtual firmware option for the version that you want to upgrade to.



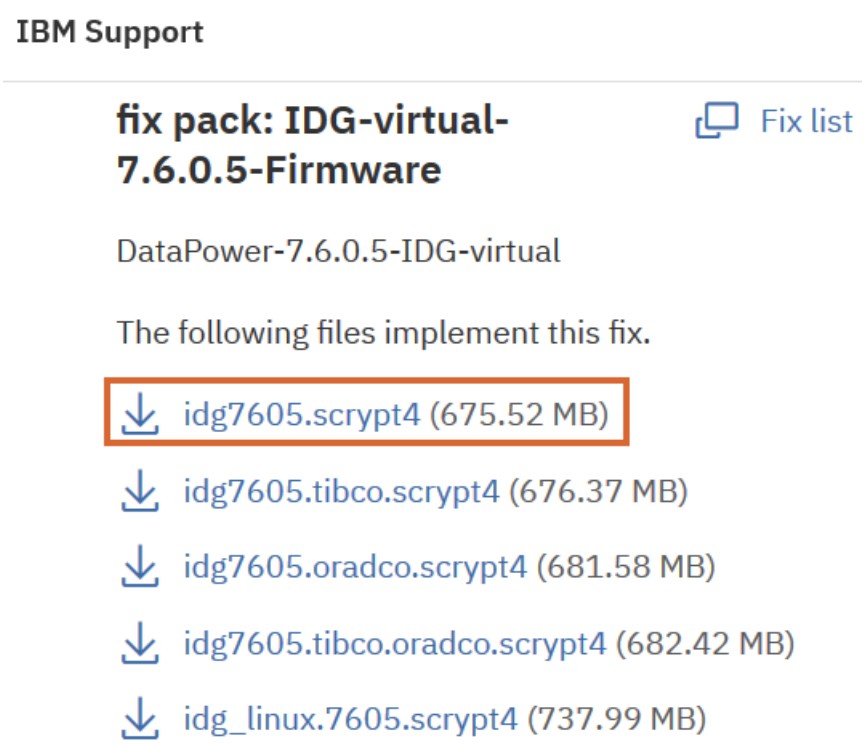
__ g. Click **Continue**.

__ h. You are prompted to sign on with your IBM ID.

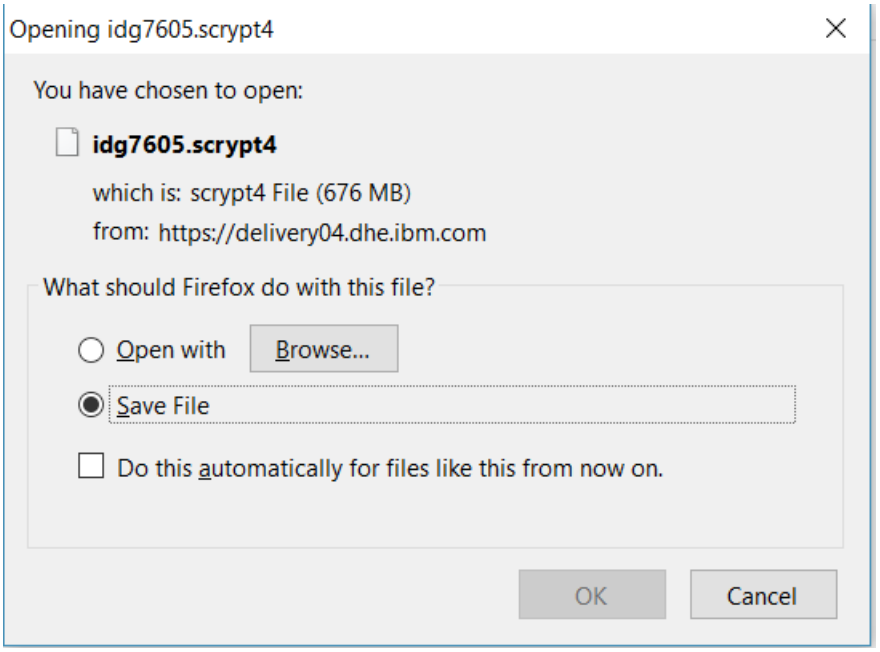
__ i. When you have signed on, you are prompted for the download method. The choices are Download Director (requires Java), FTPS, or HTTPS. Select an option and click **Continue**.

__ j. Click **I agree** in the “View and accept terms” window.

__ k. Select the file for your environment.



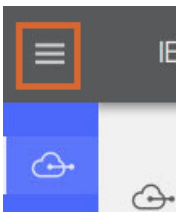
__ l. Save the file to a folder on your workstation.



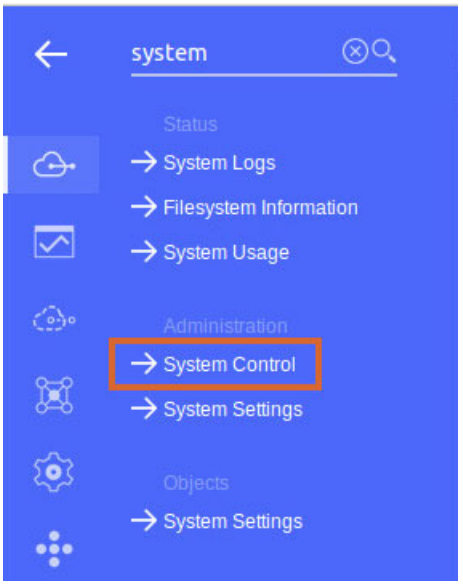
1.4. Upload and upgrade the firmware

In this section, you upload the firmware file from the local file system, and then upgrade the firmware on the gateway to the new level.

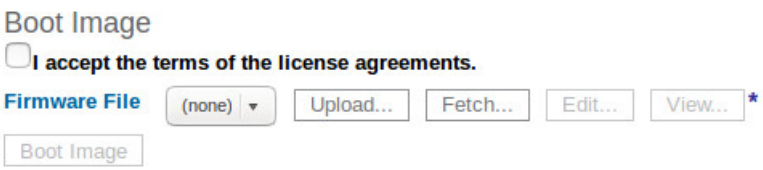
__ 1. Click the **Open menu** icon (“hamburger” icon) of the Blueprint Console.



__ 2. Enter `system` in the Search field and click **System Control.**



__ 3. You use the “Boot Image” section of the page.

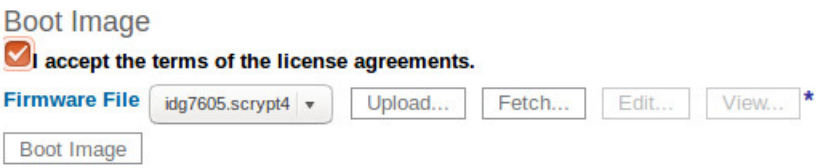


__ 4. Click **Upload** and then click **Browse** to upload the firmware file **<current>** from the directory **/home/localuser/firmware** to the gateway. An example is the **idg7605.scrypt4** file.

Information

After you click **Upload**, a pop-up window appears. In that pop-up, use **Browse** to go to the **/home/localuser/firmware** directory. Select the **idg7605.scrypt4** file there. Click **Upload** in this window. The window refreshes and indicates that the upload is occurring. Because the firmware file is large, this step might take a few minutes. When the upload completes, click **Continue** to close the pop-up window.

__ 5. Select the **I accept the terms of the license agreements** check box, and click **Boot Image**.



__ 6. Click **Confirm** to restart.

__ 7. You might get a dialog box about the boot process taking some time. The firmware validation and loading takes several minutes. If you get a Warning dialog box about a supported feature not in the firmware, click **Continue**.

__ 8. When you click in the Blueprint Console to make it switch web pages, the DataPower login page appears after the successful restart instead of the selected page. Log in to the gateway as before.

__ 9. The restart begins. The restart might take 4 - 5 minutes. When the DataPower Login page appears, you see the new level of the DataPower Gateway on the login dialog box.

__ 10. Log in again.

__ 11. Use the **Help** icon and the **About** selection to verify the new firmware level.

Information

The CLI command **boot image** also initiates the firmware upgrade. This command does not have the “upload” capability, so other CLI or Blueprint Console approaches must be used to get the new firmware level into the image: directory on the gateway.

1.5. Switch the installation image

The firmware provides an option to return to the earlier level of the firmware (switch installation image). In previous versions of the firmware, this action was called “firmware rollback”.

__ 1. Enter `system` in the **Search** field under the Open icon. Select **System Control** from the resulting list.

__ 2. Look for the **Switch Installation Image** section.

__ 3. Click **Switch Installation Image**.

__ 4. Click **Confirm** in the dialog box.

__ 5. The gateway restarts to the earlier level of the firmware. This operation takes a few minutes.

__ 6. When you attempt to change the Blueprint Console page, the DataPower Login page appears instead due to the restart.

__ 7. Log in again.

__ 8. Use the **About** selection under the **Help** icon to verify the older firmware level. It should be at the **current-1** level, the earlier level.

__ 9. Change to the System Control page.

__ 10. Perform the image switch process again to get the gateway back to the **current** level.

__ 11. Check the firmware level under the **Help** icon. It should be at the **current** level. This level is the firmware level that is used for the remaining lab exercises.

Information

Notice that the roll-back alternates the primary and secondary installation of firmware. It cannot roll back to any other earlier level.

The CLI command **boot switch** can also run the image switch.

__ 12. Log out of the DataPower Blueprint Console by clicking **Logout** in the upper-right corner of the page, under the logged-in user ID menu item. You might need to click **OK** in the confirmation dialog box that is displayed.

End of exercise

Exercise review and wrap-up

In this exercise, you upgraded the virtual gateway to a newer firmware level. You also used the installation image switch function to alternate the active firmware level.

Exercise 2. Using the CLI and the XML Management Interface to manage DataPower appliances

Estimated time

01:30

Overview

In this exercise, you learn how to manage user resources and domain configuration, run simple network testing, and retrieve appliance status information. You use the CLI, SOMA, and AMP administrative interfaces.

Objectives

After completing this exercise, you should be able to:

- Create DataPower resources by using the CLI
- Create DataPower resources by using SOMA requests
- Send appliance management requests by using AMP

Introduction

The DataPower gateway supports several ways to administer the gateway. The primary ways for non-GUI interfaces are the command-line interface (CLI), XML Management Interface (supports SOAP management (SOMA) and Appliance Management Protocol (AMP) requests), and REST Management Interface. Almost any request that can be run in the web management interface can also be run by using CLI, XMI, or RMI. The particular approach an administrator might use depends on many factors: familiarity with the syntax, automation requirements, configuration management procedures, and others.

This exercise contains six activities:

- In the *first* activity, you use CLI commands to create a student administrative account and student administrative user group. Then, you use the new administrative account to create a developer domain, user group, and user account by using XML Management Interface requests.
- In the *second* activity, you use your developer account and CLI commands to import a service configuration that is stored in an HTTP server. More CLI commands are used to modify the protocol handler in the service to use the correct port.

- In the *third* activity, you test the service by calling an HTML page on the HTTP server.
- In the *fourth* activity, you create domains that you are going to need for later exercises. You then modify the user group to include the new domains.
- In the *fifth* activity, you send AMP requests to interrogate the gateway.
- In the *sixth* and final activity, you use RMI to create, update, and delete a domain, and save the configuration.

Requirements

To complete this exercise, you need:

- *Administrative* access to the DataPower gateway
- Linux *terminal window* to log on to the DataPower gateway CLI interface
- An *HTTP server* (in your student image)
- The RESTClient add on installed in Firefox

Exercise instructions

Preface

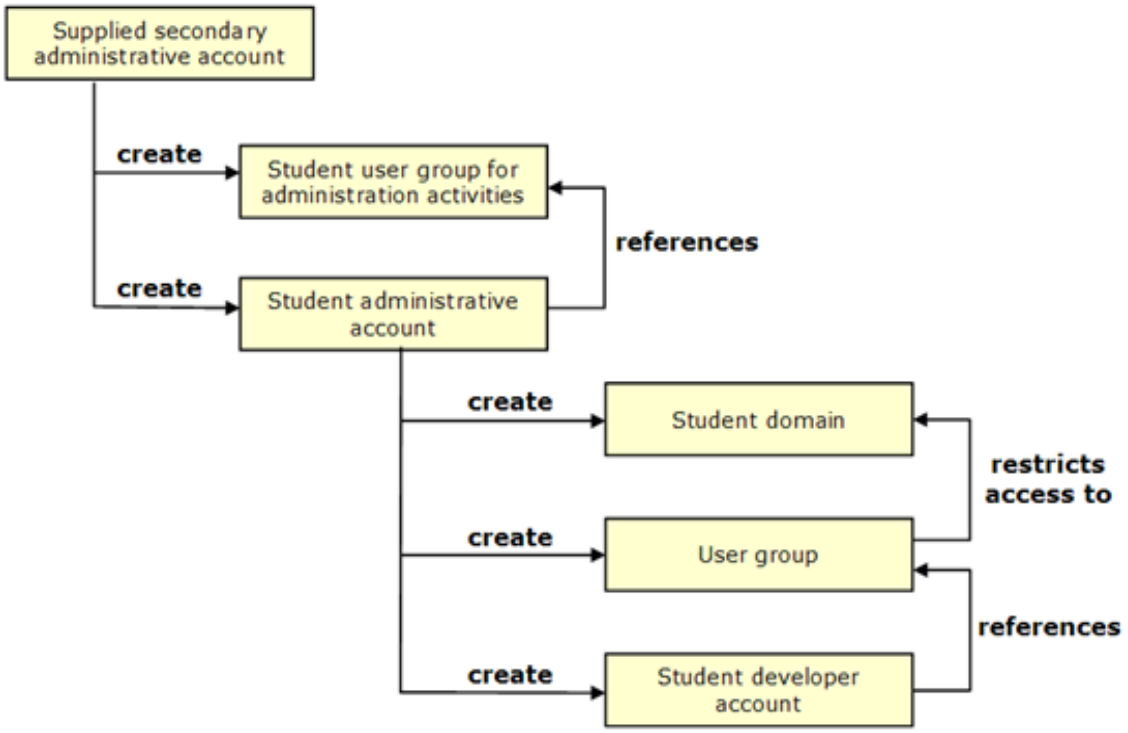
- All exercises in this chapter depend on the availability of specific equipment in your classroom (physical or virtual).
- The following variables are used in this exercise:
  - *<lab_files>*: **/home/localuser/labfiles/dp**
  - <image_ip>: IP address of the image
  - *<dp_admin_login>*: DataPower secondary administrator user name
  - *<dp_admin_password>*: DataPower secondary administrator password
  - *<dp_internal_ip>*: IP address of the gateway's management interfaces
  - *<dp_public_ip>*: IP address of the public services on the gateway
  - *<dp_WebGUI_port>*: Web management port number of the DataPower gateway; the default port is 9090
  - *<dp_xml_mgmt_port>*: The port number of the DataPower XML Management interface: 5550
  - *<dp_rest_mgmt_port>*: The port number of the DataPower REST Management interface; the default is 5554
  - *<mpgw_dpadmin>*: DPAdmin multi-protocol gateway service port: 10*nn*1
  - <student*nn*_admin_group>: User group for administrative activities
  - <student*nn*_admin>: Student administrator account that is created by using the CLI interface
  - <student*nn*_admin_password>: Password for the student administrator account
  - <student*nn*>: Student developer account that is created by using the CLI interface
  - <student*nn*_password>: Password for the student account
  - <student*nn*_domain>: Student application domain that is created by using the CLI
  - <student*nn*_developer_group>: User group object that is associated with the student*nn* account

Hint

In this exercise, you supply and change a number of student passwords. Write the passwords *immediately* in Appendix A, "Master exercise variable table," on page A-1 when you add or change them so that you can refer to them later.

Create user account, user group, and domain objects

In this section, you use a supplied general administrative account to create a user group for administrative activities, and to create your student administrative account that references that user group. Using this new student administrative account, you then create an application domain and a user group. Finally, you create a student developer account that references the new user group, and is restricted to the new application domain. These resources are displayed in the following graphic.

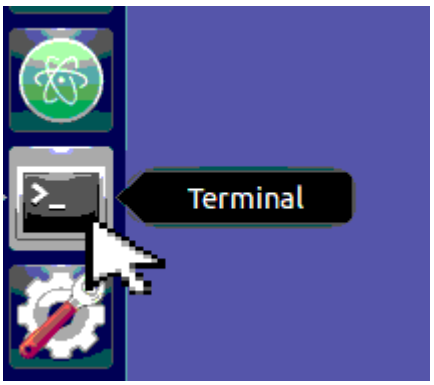


2.1. Create a student administrative user account by using CLI

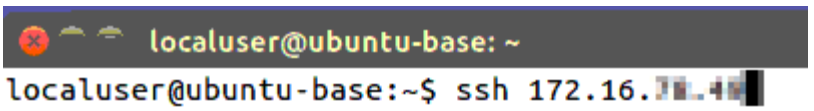
In this section, you create an administrative user account by using the CLI interface. Using the supplied general administrative account `<dp_admin_login>`, log on to the CLI with SSH. Create your own administrative user group, and then an administrative account object that references the user group.

__ 1. Log on to the CLI interface.

__ a. On your Linux desktop, open a command-line terminal by clicking the **Terminal** icon from the launcher.



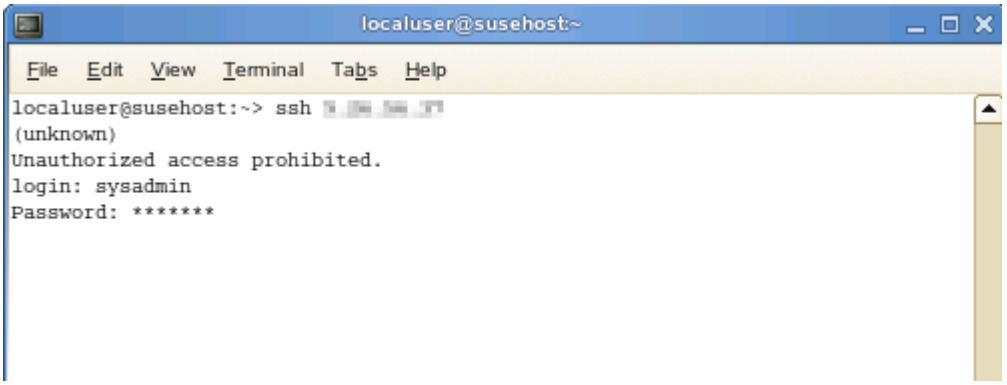
__ b. In the terminal window, run the `ssh` command by using the IP address: `<dp_internal_ip>`



Note

If during the execution of the `ssh` command, you are prompted that the authenticity of the host cannot be established because of the RSA key fingerprint, then continue with the connection by typing `yes`. The request for the RSA fingerprint can occur when connecting for the first time.

__ c. A login prompt is displayed. Enter the user name *<dp_admin_login>* and the admin password *<dp_admin_password>*



__ d. If prompted to select a domain, enter `default`.

```
DP50-IDG-760
Unauthorized access prohibited.
login: sysadmin
Password: ****************
Domain (? for all): default

Welcome to IBM DataPower Gateway console configuration.
Copyright IBM Corporation 1999-2018

Version: IDG.7.6.0.5 build 296276 on Feb 11, 2018 11:01:40 AM
Serial number: 0000000

idg#
```

Note

The value `DP50-IDG-760` that displays after the `ssh` command is the **Appliance Name** that is defined in the System Settings. This value is customizable by the administrator.

The command-line prompt depends on which gateway model is accessed. For example, it shows `idg` when you are using `IBM DataPower Gateway`.

__ 2. Verify that user group access profiles apply to CLI access and to web management access.

__ a. Check that RBM Settings are enabled. Query the RBM Settings object status.

```
idg# show rbm
```

__ b. Examine the results of the query for the **admin-state** property. The correct state is **enabled**.

__ c. Examine the results of the query for the **apply-cli** property. Verify that it is **on**.

Attention

The following instructions apply only if the RBM Settings are incorrect.

If either of these properties is set incorrectly, you must change these values. The exercise does not work properly without these values. The values can be changed in this CLI session.

These instructions use the web management interface because it does not take you out of the current CLI session state.

Configure the RBM settings for CLI

a. While you are still logged in to web management in the browser, you must also set a property for RBM Settings. Click **Administration > Access > RBM Settings**.

b. From **Main**, select the **Enable administrative state** check box.

c. From **Account policy**, select the **Enforce RBM on CLI** check box.

d. Click **Apply**.

e. Click **Save**.

These settings specify that an access profile, as defined in a user group, controls the CLI access of an account that is associated with the user group.

__ 3. Open `<lab_files>/mgmtInterface/createAdminUserGroup.txt`. The content that is in this file can be copied to the CLI interface by using copy-and-paste to save you from having to enter the command and reduce the chance of syntax errors. Be sure to replace the *nn* in `student`***nn***`_admin_group` with your actual student number before pressing Enter.

__ a. Click the **Files** icon in the launcher (shortcut bar on the left of the desktop) to open the Ubuntu File Manager.

__ b. Double-click the **labfiles** folder.

__ c. Double-click the **dp** folder.

__ d. Double-click the **mgmtInterface** folder. You can now see the files and folders within the mgmtInterface directory.

__ e. Right-click the **createAdminUserGroup.txt** file and click **Open with gedit**.

__ f. Replace all occurrences of the *nn* in `student`***nn***`_admin_group` with your actual student number.

__ g. Leave the file open in the editor. You copy the individual lines of text into the terminal in the next step.

__ 4. Create a user group for student administrative access.

You can search for the appropriate command to create a user group object.

__ a. In the terminal window, enter global configuration mode.

```
idg# configure terminal
```

Information

You can also use the shortcut `co` instead. Modifying configuration on the DataPower gateway is only possible in the global configuration mode.

__ b. Enter the user group configuration mode to create a user group and populate it with the appropriate fields. Each line can be copy-and-pasted from the `createAdminUserGroup.txt` file, which is open in gedit, or the contents can be manually entered.

Important

If you choose to copy and paste, be sure to replace the *nn* in `student`***nn***`_admin_group` with your actual student number before pressing Enter. The process applies to the first two commands that you enter. Be sure to press Enter after you paste each command.

```
idg(config)#usergroup studentnn_admin_group

idg(config usergroup studentnn_admin_group)#summary "studentnn
administrative user group"

idg(config usergroup studentnn_admin_group)#access-policy
"*/*/*?Access=rwadx"

idg(config usergroup studentnn_admin_group)#access-policy
"*/*/file/store?Access=r"

idg(config usergroup studentnn_admin_group)#access-policy
"*/*/network/interface?Access=r"
```

The general purpose of these access policies is to allow an administrative user to manipulate resources on the gateway, except for the network settings and the `store:///` directory.

__ c. Enter `exit` to leave the user group configuration mode and create the object.

```
idg(config usergroup studentnn_admin_group)# exit
```

__ 5. Create an administrative user account that uses the new user group.

__ a. In the terminal window, verify that you are at the following prompt:

```
idg (config)#
```

__ b. You can use the `help` command to find the user account object. Alternatively, you can search the DataPower CLI documentation. The appropriate command to create a user account is the `user` command. Test the help feature by typing `user` and pressing enter.

```
idg(config)# user
Usage: user <user-name>
```

__ c. The Usage prompt indicates that the command was not complete. Enter the command with the user name `<student`*`nn`*`_admin>` again. Be sure to replace the *`nn`* you see in the instructions with your student number.

```
idg(config)# user student nn _admin
New User configuration
idg(config user student nn _admin)#
```

__ d. Enter `help` again to search for the command to modify permissions.

__ e. After reading through the list of command descriptions, notice that the `access-level` command is the appropriate command.

__ f. Modify the permissions of this user by using the `access-level` command.

```
idg(config user student nn _admin)#access-level group-defined
```

__ g. Specify the user group that you created in a previous step as the group with which this account is associated.

```
idg(config user student nn _admin)#group student nn _admin_group
```

Note

Because the user account references the user group, the user group had to be created *before* the student account.

__ h. Enter a password for the account.

```
idg(config user student nn _admin)# password student<nn>
Re-enter new password: *******
```

Note

Ensure that you type the same password for “Re-enter new password”, in this case it is `student<nn>`.

__ i. Enter `show` to verify the user account settings. The password is not displayed. The following parameters are displayed:

```
admin-state enabled
summary ""
access-level group-defined
group student nn _admin_group [up]
```

__ j. Enter `exit` to leave the user configuration mode and apply the changes.

__ k. Enter `write mem` and answer `y` (yes) to overwrite the previously saved configuration. This command saves the new user and user group configuration to the startup configuration of the gateway.

__ l. Enter `exit` to leave the configuration mode.

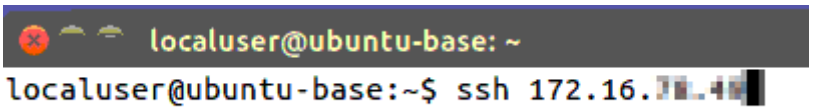
__ m. Enter `exit` to exit `ssh` mode.

__ 6. Log on to the SSH terminal again by using the `student`*nn*`_admin` user account and verify that you can run CLI commands in the default domain.

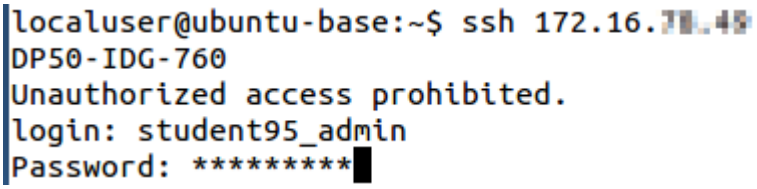
__ a. In the terminal window, run the `ssh` command by using the IP address `<dp_internal_ip>` and press Enter.

Note

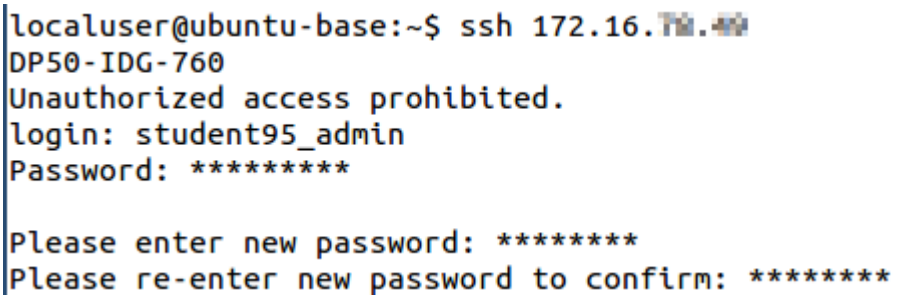
If you did not close the terminal window, the Up arrow can be used to toggle through previously entered commands.



__ b. A login prompt now is shown. Enter the student administrative user name `<student`*nn*`_admin>` and the password as `student`*nn*.



__ c. You might be prompted to enter a new password. Enter a new password and write down the new password in the table in Appendix A.



Information

The command `suppress-password-change` can be used to control whether the password for this account must be changed after the initial login by the account owner.

__ d. Enter `default` to identify the default domain as the choice.

If other domains to which you have access are defined on the gateway, you might be prompted to select which domain you want to use. This process is similar to the domain choice on the web management login page.

__ e. Enter the command `configure terminal` to enter global configuration mode.

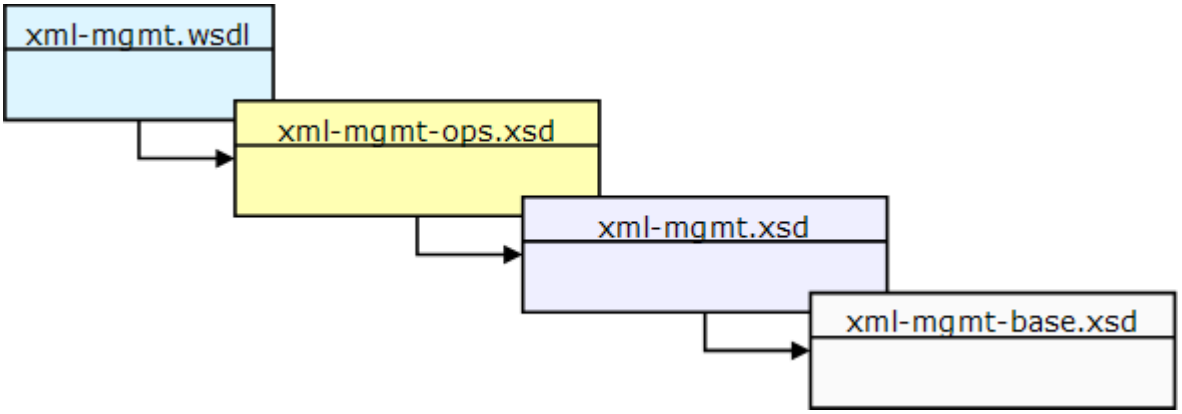
__ f. You verified that the student administrative user account is accessible. Enter `exit` to leave the global configuration mode.

__ g. Enter `exit` to leave the CLI session.

2.2. Review the XML Management Interface WSDL (optional)

In this section, you can review the DataPower XML Management Interface WSDL and the associated XSD files. You do *not* need to do this section to complete any of the other sections in this lab. These files contain the information DataPower uses to process XML Management Interface requests coming in through the XML Management Interface. These WSDL and XSD files are opened for viewing only.

The WSDL file (`xml-mgmt.wsdl`) describes the XML Management Interface. This WSDL file references more XML schema files (`xml-mgmt-ops.xsd`, `xml-mgmt.xsd`, and `xml-mgmt-base.xsd`). These files describe the operations that an administrative client can start to create, retrieve, update, or delete DataPower objects.



Files that are on the DataPower gateway in the `store:///` subdirectory describe the XML Management Interface interface. The XML Management Interface interface files are included with the DataPower gateway and can change between releases.

- `xml-mgmt.wsdl`: The WSDL file that defines the services that are available through the XML Management Interface interface.
- `xml-mgmt-ops.xsd`: The schema file that defines the operations that can be sent in XML Management Interface requests.
- `xml-mgmt-base.xsd`: The schema file that defines the primitive management types in XML Management Interface messages.
- `xml-mgmt.xsd`: The schema file that defines the non-primitive management types in XML Management Interface messages.

Attention

Files that are included with the DataPower gateway that are in the `store:///` subdirectory should *not* be edited, updated, or deleted. These files help control DataPower processing. If you want to work with these files, the good practice recommendation is to make a copy of the files in another subdirectory and work with the copy, leaving the original files intact.

__ 1. You have a choice in how you want to review the files:

- Use the file browser and the **gedit** editor to examine the files in a text editor. This view is not structured.
- Use the file browser and the **Atom** editor to examine the files in a text editor. This view is slightly more structured.
- Use **Firefox** to view the files. You get a structured view, plus the ability to collapse and expand the hierarchy.

__ 2. The files are in the `<lab_files>/mgmtInterface/DP` directory. Use your viewing choice to open the files to view the structure and elements that are available in the WSDL and schema. Most of the details are in the XSD files. A suggested approach to analyze the WSDL is by using the following steps:

__ a. Go to `<lab_files>/mgmtInterface/DP` and open the `xml-mgmt-ops.xsd` file.

__ b. Search for the `request` element definition. You can find the following text:

```
<xsd:element name="request">
   <xsd:complexType>
      <xsd:choice>
      ....
```

__ c. The child tag of `Body` is `request`, and it has one of the following child tags as described in the XML schema definition:

- `get-config`
- `get-samlart`
- `get-log`
- `get-filestore`
- `get-file`
- `set-file`
- `do-export`
- `do-backup`
- `do-restore`

Other operations also exist in this file, and more can be added in future releases.

__ d. Look near the end of the `request` element definition. Notice that it also defines a `domain` attribute. The `request` element can be used for retrieving and modifying DataPower configurations. The `domain` attribute filters the request to a specific domain.

You can continue to examine the XML Management Interface WSDL at your discretion.

This approach can also be used to review the AMP WSDL that is used in later sections of this exercise. The appropriate WSDL and XSD file are in the same location as the XML Management Interface files.

2.3. Create developer resources by using XML Management Interface

In this section, you use the SOAP Management (SOMA) part of the XML management interface to create a domain, user group, and user account for development activities. You send the XML Management Interface requests by using cURL from a terminal window.

__ 1. Create the XML Management Interface request to define a specific application domain, user group, and user for development.

__ a. Using File Browser, go to the `<lab_files>/mgmtInterface` directory.

__ b. Right-click the `createDeveloperResources.xml` file and click **Open With > gedit**.

__ c. Notice that the `set-config` request configures three different resources: a domain, a user group, and a user. Because the user object refers to the user group, and the user group refers to the domain, the particular order of the configuration is necessary.

The string "`*/default/*?Access=r`" enables read access to the file system of the default domain.

The string "`*/student`*nn*`_domain/*?Access=r+w+a+d+x+`" enables full access to the `student`*nn*`_domain`.

__ d. Replace the *nn* in the specifications with your actual student number in each of the nine locations.

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <dp:request xmlns:dp="http://www.datapower.com/schemas/management">
      <dp:set-config>
        <Domain name="studentnn_domain">
          <UserSummary>Test domain for student account nn.</UserSummary>
          <NeighborDomain class="domain">default</NeighborDomain>
        </Domain>
        <UserGroup name="studentnn_developer_group">
          <UserSummary>Developer group for the student nn domain.</UserSummary>
          <AccessPolicies>*/studentnn_domain/*?Access=r+w+a+d+x</AccessPolicies>
          <AccessPolicies>*/default/*?Access=r</AccessPolicies>
        </UserGroup>
        <User name="studentnn">
          <Password>studentnn</Password>
          <GroupName>studentnn_developer_group</GroupName>
          <AccessLevel>group-defined</AccessLevel>
          <UserSummary>Developer account on the student nn domain.</UserSummary>
        </User>
      </dp:set-config>
    </dp:request>
  </env:Body>
</env:Envelope>
```

__ e. Save the changes to the file, and close it.

__ 2. Send the `createDeveloperResources.xml` file to the XML Management Interface on the DataPower gateway.

__ a. In a terminal window, use the `cd` command to change to the *<lab_files>*`/mgmtInterface` directory:
`# cd /home/localuser/labfiles/dp/mgmtinterface`

__ b. Enter the following cURL command:

```
curl -H "Content-Type: text/xml" --data-binary
@createDeveloperResources.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

Note

The <*dp_xml_mgmt_port*> is 5550.

__ c. Verify that you get a valid response with three *OK* responses, one for each resource:

```
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <dp:response
         xmlns:dp="http://www.datapower.com/schemas/management">
         <dp:timestamp>2018-01-12T13:22:27-05:00</dp:timestamp>
         <dp:result>OK</dp:result>
         <dp:result>OK</dp:result>
         <dp:result>OK</dp:result>
      </dp:response>
   </env:Body>
</env:Envelope>
```

__ d. The `<dp:result>OK</dp:result>` confirms the creation of a DataPower object.

__ e. Edit `saveConfig.xml`. This XML Management Interface request specifies that the current running configuration is saved, or persisted. Notice that its request is a *do-action*.

__ f. Close the file.

__ g. Send the XML Management Interface request to save the configuration:

```
curl -H "Content-Type: text/xml" --data-binary @saveConfig.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

Note

The <*dp_xml_mgmt_port*> is 5550.

__ h. Verify that you get a valid response: an OK. You now have a domain, user group, and user that are defined for a developer.

Import and update a service configuration by using CLI

In this section, you perform several activities to import the **DPAdmin** service, update it, and test the service.

Since file upload from the CLI session to the gateway is not supported, you use an HTTP server as a technique to load files into the file system.

The file that contains the exported service definition is stored on the HTTP server. The exported service does not contain the private key and certificate files. You must load those files as well. A CLI script is used to accomplish several of these actions.

Last, you test the **DPAdmin** service to verify that it works.

2.4. Import a domain configuration from an HTTP server

In this section, you copy the contents of a compressed file, which contains an application service, from an HTTP server to the DataPower gateway and your developer domain.

__ 1. Verify that the HTTP Server on the image is running.

__ a. Open the web browser by clicking the Firefox icon from the Ubuntu launcher.

__ b. Enter the URL `http://localhost` in the browser. You see the text from the HTTP server, *Welcome to WE761 - DataPower V7.6 Administration web server*, confirming the web server is up and running.

__ 2. Log in to the CLI as a student developer.

__ a. Log to the CLI again by using the **<student*nn*>** ID and **<student*nn*>** password that was created in the previous section. You might be prompted to select which domain to log in to. Enter `<student*nn*_domain>`

__ b. If you are prompted to enter a new password, be sure to write it down.

Attention

Ensure that you sign on to the **<student*nn*>** ID, not the **<student*nn*_*admin*>** ID that you signed on with earlier.

__ 3. Use the network utilities to verify connectivity with the HTTP server.

Verify network access to the HTTP server before importing the domain configuration.

__ a. The `.zip` file is imported into the new domain you created in the previous section; ensure that you are logged on to the `student*nn*` domain.

```
idg[student<i>nn</i>_domain] #
```

__ b. Use the `ping` command to verify access to the HTTP server at `<image_ip>`.

```
idg[student<i>nn</i>_domain] # ping <image_ip>
PING image_ip_address (192.168.XX.XX) with 56 data bytes of data
64 bytes from 192.168.XX.XX: seq=0, ttl=127, rtt=2.0 ms
64 bytes from 192.168.XX.XX: seq=1, ttl=127, rtt=2.0 ms
64 bytes from 192.168.XX.XX: seq=2, ttl=127, rtt=1.0 ms
64 bytes from 192.168.XX.XX: seq=3, ttl=127, rtt=1.0 ms
64 bytes from 192.168.XX.XX: seq=4, ttl=127, rtt=1.0 ms
64 bytes from 192.168.XX.XX: seq=5, ttl=127, rtt=0.0 ms
6 packets transmitted, 6 received, 0% loss, time 5999 ms
```

Information

The `<image_ip>` is the IP address of your VMware image. Use the `/sbin/ifconfig` command in a terminal window to obtain the IP address.

The `ping` command (ICMP echo request) might not complete even though the IP address is reachable. Configurations of firewalls and routers can block this command.

__ c. Use the `test tcp-connection` command to verify that you can access the HTTP service on port 80.

```
test tcp-connection <image_ip> 80
TCP connection successful
```

__ 4. Examine other helpful networking commands.

__ a. View the list of active TCP connections on the gateway by using the command: `show tcp`

The list displays all of the TCP connections in their various states of the TCP handshake.

__ b. View the list of currently logged-in users by using the command: `show users`

Your ID should display.

__ 5. Copy the security key-related files `DPCert.cer`, `Alice-sscert.pem`, and `Alice-privkey.pem` from the HTTP server to the `cert:///` directory on the gateway. The files are in the `dp` subdirectory of the HTTP server. The key and certificate objects that are within the DPAdmin service to be imported use these files.

__ a. After entering configuration mode, enter the CLI copy command for each of the three files:

```
co
copy http://<image_ip>/dp/DPCert.cer cert:///DPCert.cer
copy http://<image_ip>/dp/Alice-sscert.pem cert:///Alice-sscert.pem
copy http://<image_ip>/dp/Alice-privkey.pem cert:///Alice-privkey.pem
```

__ b. Type `exit`.

__ 6. Import the application service **DPAdmin** from the HTTP server.

Use the `import-package` command to create an object configuration that specifies how to obtain the multi-protocol gateway configuration that is stored on the HTTP server. The `import-execute` command initiates the import.

__ a. In the CLI session, if required, enter global configuration mode with the `co` command.

```
idg[studentnn_domain](config)# co
```

__ b. Create an import package configuration object that is called **DPAdminPackage** by using the `import-package` command.

```
idg[studentnn_domain](config)# import-package DPAdminPackage
New Import Configuration File configuration
```

__ c. Specify the `source-url` to be: `<image_ip>`

```
idg[studentnn_domain](config import-package DPAdminPackage)#
source-url http://<image_ip>/dp/DPAdmin.zip
```

__ d. Specify the format of the imported file as: `ZIP`

```
idg[studentnn_domain](config import-package DPAdminPackage)#
import-format ZIP
```

__ e. Enter `exit` to finish completion of the **DPAdminPackage** object.

__ f. Use the `import-execute` command to import the **DPAdminPackage** object you created.

```
idg[studentnn_domain](config)# import-execute DPAdminPackage
Loading import-package 'DPAdminPackage'.
Import package is complete.
```

Attention

If you get an error when running the `import-execute` command, check that the `source-url` value is entered correctly.

__ 7. The name of the service you imported is `DPAdmin`. Check the status of this multi-protocol gateway service you imported.

__ a. View the status of the **DPAdmin** multi-protocol gateway by entering the command:

```
idg[studentnn_domain](config)# show mpgw DPAdmin
```

__ b. Verify that you see the following text at the top:

```
mpgw: DPAdmin [down]
...
```

The **DPAdmin** multi-protocol gateway configuration was imported successfully as indicated by the message in the CLI prompt; however, the **DPAdmin** multi-protocol gateway status is down.

Information

The **DPAdmin** multi-protocol gateway status is down because the port number that is used conflicts with another port number on the gateway. If the status is **up**, then the port number was unassigned at the time. You still must configure the correct port number. In the next step, you fix this problem.

__ c. Use the `show log` command to examine the system logs. Scroll through the list to view any errors about the **DPAdmin** multi-protocol gateway service. Notice at least one error about referenced objects that are down or about port assignments.

__ 8. A multi-protocol gateway receives requests by using a front-side protocol handler. The "show mpgw DPAdmin" command displays the name of the front-side protocol handler. The DPAdmin service uses a handler that is named `https_fsh_dpadmin`. Update its definition to point to the correct port.

__ a. Enter the HTTP Front Side Handler configuration mode to change the port to your student-specific value: `<mpgw_dpadmin>`

```
idg[studentnn_domain](config)# source-https https_fsh_dpadmin
Modify HTTP Front Side Handler configuration
idg[studentnn_domain](config source-http https_fsh_dpadmin)#
port <mpgw_dpadmin>
```

Note

The <*mpgw_dpadmin*> port is 10nn1.

__ b. Enter `exit` to leave this configuration mode.

__ c. Check whether the gateway service is now up:

```
idg[studentnn_domain](config)# show mpgw DPAdmin
```

__ d. It should now report its status as **up**.

__ e. Enter `write mem` and answer `y` (yes) to overwrite the previously saved configuration.

__ f. Enter `exit` to exit out of ssh.

Information

Importing configurations from an external HTTP or source control server is a common technique that is used when deploying configurations into production.

A typical development environment consists of developers that are creating configurations on a *development* DataPower gateway and moving it to either an HTTP or a source control server. A production DataPower gateway can point to a server that contains the configuration.

Test the DPAdmin service

In this section, you test the operation of the DPAdmin multi-protocol gateway. You use an HTML page under the HTTP server to call the service and send the appropriate parameters.

2.5. Test the DPAdmin multi-protocol gateway service

In this section, you test the operation of the **DPAdmin** service.

__ 1. Open a web browser and enter the URL (a shortcut in the toolbar that is named **XML Interface** might be available): `http://<image_ip>/dp/searchConfig.html`

__ 2. In the web page that opens, enter the following information:

- DP XML Management IP Address: *<dp_internal_ip>*
- DP XML Management Port: *<dp_xml_mgmt_port>*
- DP Service Host: *<dp_internal_ip>*
- DP Service Port: *<mpgw_dpadmin>*
- Domain: `default`
- Search for: `User`
- Object name: `student`*nn*

Information

The DP XML Management Host Name is the gateway that is receiving the XML management request from the back side of the DPAdmin service. In the lab scenario, this gateway is the same gateway that is hosting the DPAdmin service itself.

The DP XML Management Port is the port on the DP XML Management Host that the XML Management Interface is listening on (default is 5550).

The DP Service Host is the gateway that is hosting the DPAdmin service itself.

The DP Service Port is your assigned port for the DPAdmin service.

The DPAdmin service takes the parameters from the web page form and constructs an XML Management Interface request. The service then forwards the request to the gateway that contains the XML Management Interface and the DataPower object that is being queried for.

__ a. Click **Submit**.

__ b. You are notified of a certificate that is presented. Click **I Understand the Risk**.

__ c. Click **Add Exception** to add the certificate acceptance to the browser.

__ d. Click **Confirm**.

__ e. A login dialog box is shown; you are being challenged for an administrative login to the XML management interface. Enter `<student`*nn*`_admin>` and `<student`*nn*`_admin_password>`.

__ f. Verify that the web page returns the configuration information for your user account. The student 95 example is shown:

```
. . .
-<User read-only="true" name="student95
    <mAdminState read-only="true">enabled</mAdminState>
    <UserSummary read-only="true"/>
    <AccessLevel read-only="true">group-defined</AccessLevel>
    <GroupName read-only="true" class="UserGroup">student95_developer_group</GroupName>
</User>
. . .
```

This XML file does not appear to have any style information associated with it. The document tree is shown

```
−<env:Envelope>
  −<env:Body>
    −<dp:response>
       <dp:timestamp>2018-01-16T14:23:47-05:00</dp:timestamp>
      −<dp:config>
        −<User read-only="true" name="student95">
            <mAdminState read-only="true">enabled</mAdminState>
            <UserSummary read-only="true">Developer account on the student 95 domain.</UserSumm
            <AccessLevel read-only="true">group-defined</AccessLevel>
            <GroupName read-only="true" class="UserGroup">student95_developer_group</GroupNa
          </User>
        </dp:config>
      </dp:response>
    </env:Body>
  </env:Envelope>
```

Create and modify more resources

In this section, you use an XML Management Interface request to create more application domains that are used in later exercises. You also modify the developer user group to include the new domains.

2.6. Create more developer domains by using XML Management Interface requests

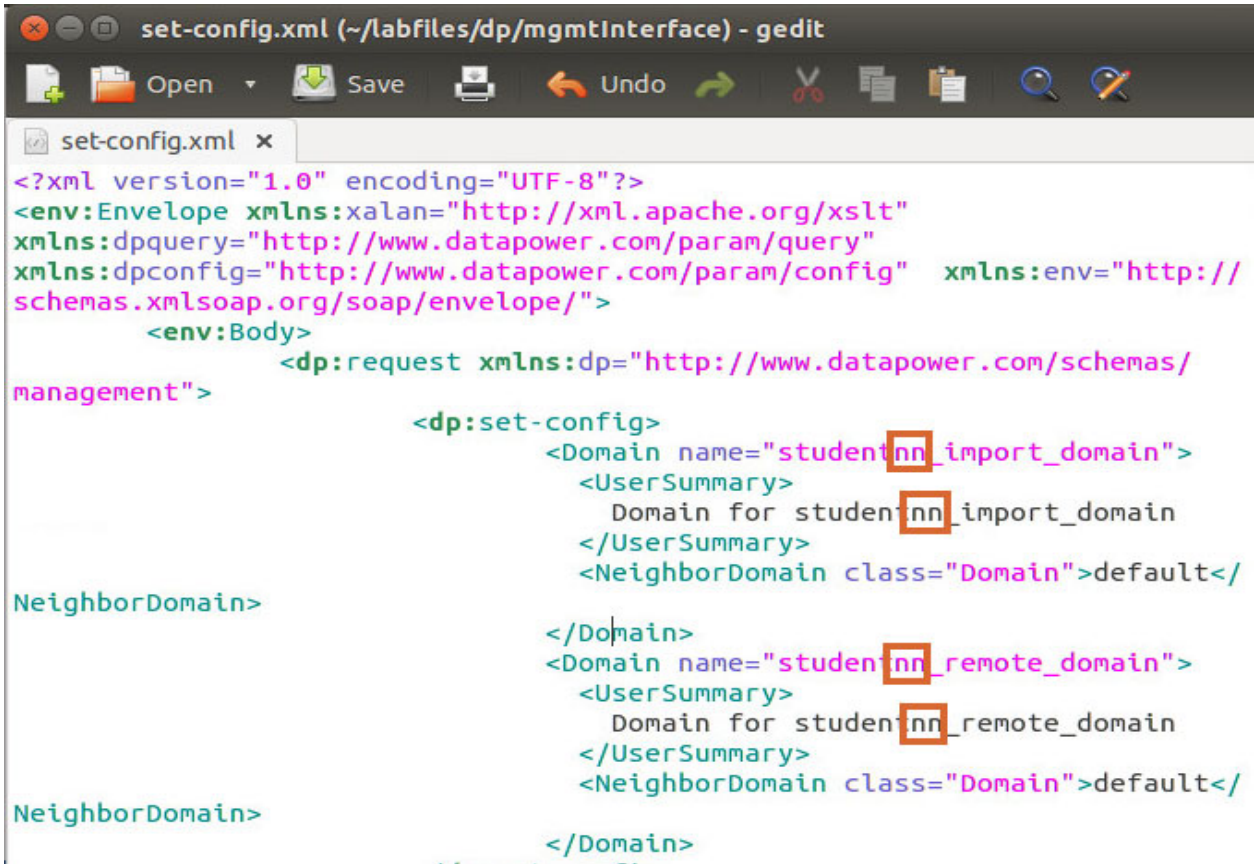
In this section, an XML Management Interface request is built that can create multiple application domains on the DataPower gateway. The domains that are created in this section are used in subsequent exercises throughout this course.

__ 1. Build the XML Management Interface request to create two more developer domains: `student`***`nn`***`_import_domain` and `student`***`nn`***`_remote_domain`

__ a. Using File Manager, go to the *<lab_files>*`/mgmtInterface` directory.

__ b. Right-click the **set-config.xml** file and click **Open With > gedit**.

__ c. Replace the four occurrences of `student`***`nn`*** in the `set-config.xml` file with your student number.



__ d. In the `set-config.xml` file, the general structure is:

```
<Domain name="student nn _import_domain">
   <UserSummary>
      Domain for student nn _import_domain
   </UserSummary>
      <NeighborDomain class="Domain">default</NeighborDomain>
</Domain>
<Domain name="student nn _remote_domain">
   <UserSummary>
      Domain for student nn _remote_domain
   </UserSummary>
      <NeighborDomain class="Domain">default</NeighborDomain>
</Domain>
```

__ 2. Save and close the `set-config.xml` file.

__ 3. Send the `set-config.xml` file to the XML management interface on the DataPower gateway.

__ a. In the terminal window, go to the *<lab_files>*/mgmtInterface directory.

__ b. Enter the following cURL command:

```
curl -H "Content-Type: text/xml" --data-binary @set-config.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

__ c. Verify that you get a valid response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <dp:response
         xmlns:dp="http://www.datapower.com/schemas/management">
         <dp:timestamp>2018-03-26T19:54:05-04:00</dp:timestamp>
         <dp:result>OK</dp:result>
         <dp:result>OK</dp:result>
      </dp:response>
   </env:Body>
</env:Envelope>
```

__ d. The two `<dp:result>OK</dp:result>` elements confirm the creation of the DataPower objects.

__ 4. Use the `saveConfig.xml` file again to save the configuration:

```
curl -H "Content-Type: text/xml" --data-binary @saveConfig.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

Verify that you get a valid response: an OK. You now have a two new domains that are defined for a developer.

__ 5. Log on to the DataPower Blueprint Console to confirm creation of the application domains.

__ a. Enter the following URL into a web browser:

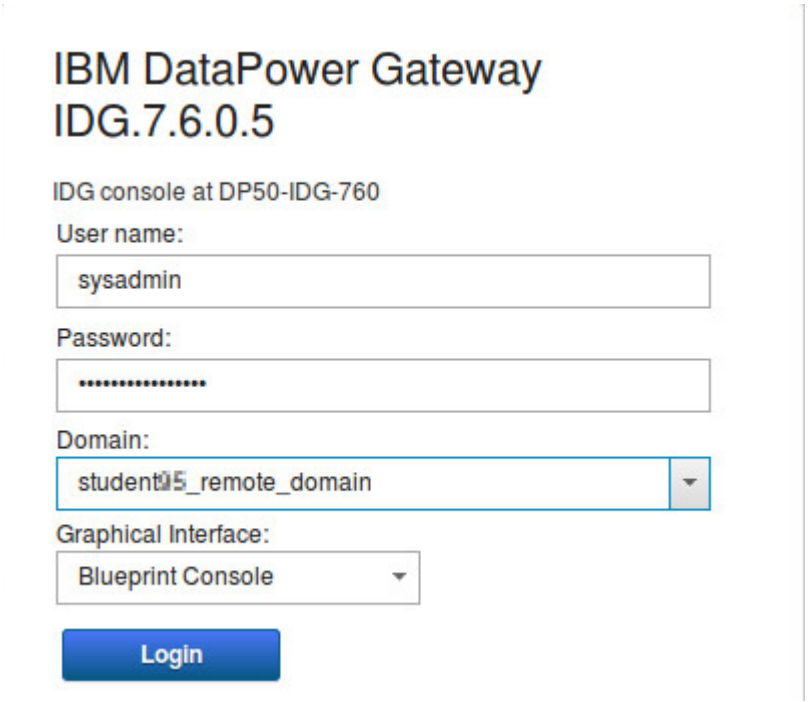
`https://<dp_internal_ip>:<dp_WebGUI_port>/dp/login.xml`

__ b. Enter the `<dp_admin_login>` user name and password.

Note

You use the administrative account because the associated user group restricts the developer account to the `studentnn_domain`. Access to the new domains is granted in a few more steps.

__ c. Select the domain that was created, `studentnn_remote_domain`, and click **Login**.



__ d. You are successfully logged in to a properly created domain.

__ e. Log out of DataPower.

__ f. Log on to DataPower again, specifying the other domain that was recently created, `studentnn_import_domain`, and click **Login**.

__ g. If the domain was created properly, you are now successfully logged in.

__ h. Log out of DataPower.

2.7. Modify the developer user group by using an XML Management Interface request

In this section, you modify your user group object, `student`*`nn`*`_developer_group`, to allow the `student`*`nn`* user account access to the domains created in the previous section: `student`*`nn`*`_remote_domain` and `student`*`nn`*`_import_domain`

You build an XML Management Interface request to accomplish this task. You can view the `student`*`nn`*`_developer_group` user group object in the web management interface to obtain an idea of the required fields in the SOAP message.

__ 1. Log on to the DataPower web management page.

__ a. If required, log in again by entering the administrative user name, `<student`*`nn`*`_admin>`, and password.

__ b. Make sure that the `default` domain is selected and click **Login**.

__ 2. Examine the steps and information that is required to define a user group object.

You do *not* change the definition here; instead, you are examining the information that is required to help build your XML Management Interface request.

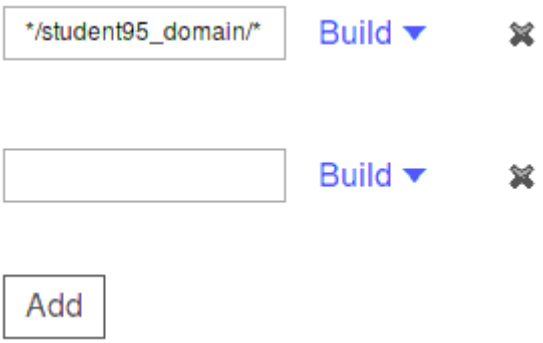
__ a. In the Blueprint Console menu, select the **Open menu** icon.

__ b. In the **Search** field, start to enter **user group**, and select **User Group** under Administration.

__ c. In the list of user group objects, click the **student*nn*_developer_group** user group object.

The **Access Profile** field is the key field that you must modify to enable the `student`*`nn`* account access to the other domains created in the previous steps.

__ d. Click **Add** to get a new entry field, and then click **Build** to create the access profile string.



__ e. A dialog box opens. Select your application domain from the Application Domain list, `student`*`nn`*`_import_domain`, and select all of the privileges.

__ f. Click **Apply**. The dialog box closes.

__ g. In the page, the string is inserted in an **Access profile** field. Copy and paste this value into a text file. You use this string when you build the XML Management Interface request. An example of the string for student 01 is:

```
*/student01_import_domain/*?Access=r+w+a+d+x
```

__ h. Click **Cancel** and then click **Leave Page** to exit the `<student`*`nn`*`_developer_group>` user group object page. You build an XML Management Interface request to complete this task in a following step.

__ 3. Use the browser to submit a request to the DPAdmin multi-protocol gateway service to view the user group configuration.

__ a. Enter the URL `http://<image_ip>/dp/searchConfig.html` into a web browser. This action opens a web page that you can use to search for the DataPower configuration.

__ b. Enter the following information:

- DP XML Management IP Address: *<dp_internal_ip>*
- DP XML Management Port: *<dp_xml_mgmt_port>*
- DP Service Host: *<dp_internal_ip>*
- DP Service Port: *<mpgw_dpadmin>*
- Domain: `default`
- Search for: `UserGroup`
- Object name: `student`*`nn`*`_developer_group`

__ c. Click **Submit**.

__ d. A login dialog box might prompt for an administrative login to the XML management interface. Enter `<student`*`nn`*`_admin>` and `<student`*`nn`*`_admin_password>`.

__ e. Examine the response that is returned. The example response here is for student95

```
-<env:Envelope>
 -<env:Body>
  -<dp:response>
    <dp:timestamp>2018-01-16T20:52:20-05:00</dp:timestamp>
   -<dp:config>
    -<UserGroup read-only="true" name="student95_developer_group">
      <mAdminState read-only="true">enabled</mAdminState>
      <UserSummary read-only="true">Developer group for the student 95 domain.</UserSummary>
      <AccessPolicies read-only="true">*/default/*?Access=r</AccessPolicies>
      <AccessPolicies read-only="true">*/student95_domain/*?Access=r+w+a+d+x</AccessPolicies>
     </UserGroup>
    </dp:config>
   </dp:response>
  </env:Body>
 </env:Envelope>
```

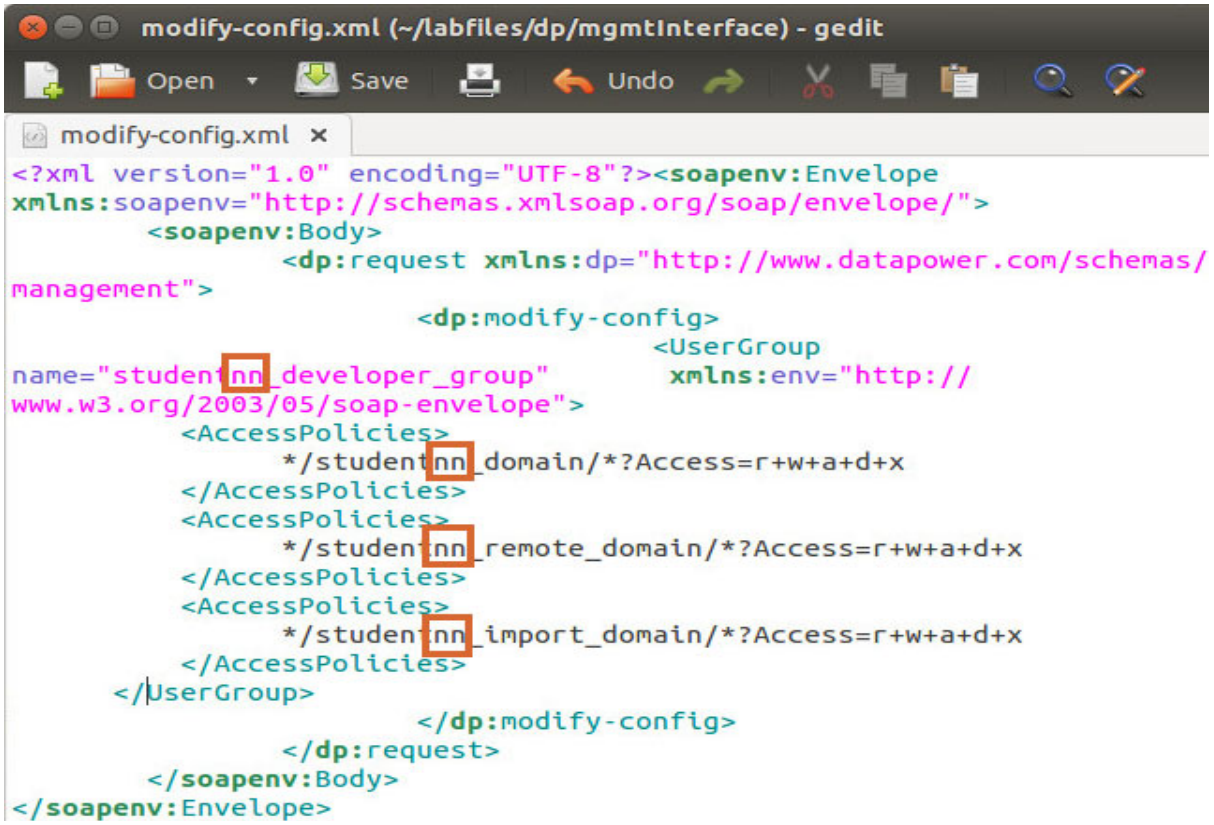
In this `studentnn_developer_group` user group response, notice that the `AccessPolicies` element contains the access policy string. You can add another access policy string by using the `AccessPolicies` element.

__ 4. Build an XML Management Interface request to modify the `studentnn_developer_group` user group object. You add another access policy string to allow the `studentnn` user account to access the new domains.

__ a. Using File Manager, go to the *<lab_files>*`/mgmtInterface` directory.

__ b. Right-click the `modify-config.xml` file and click **Open With > gedit**.

__ c. In the `modify-config.xml` file, replace the four occurrences of `student`*`nn`* with your student number.



__ d. The `modify-config.xml` file contains *(student95 example):*

```
. . .
<dp:modify-config>
      <UserGroup name="student95_developer_group"
      xmlns:env="http://www.w3.org/2003/05/soap-envelope">
          <AccessPolicies>
                */student95_domain/*?Access=r+w+a+d+x
          </AccessPolicies>
          <AccessPolicies>
                */student95_remote_domain/*?Access=r+w+a+d+x
          </AccessPolicies>
          <AccessPolicies>
                */student95_import_domain/*?Access=r+w+a+d+x
          </AccessPolicies>
      </UserGroup>
</dp:modify-config>
. . .
```

__ e. Save the file.

__ 5. Send the `modify-config.xml` file to the XML Management Interface on the DataPower gateway.

__ a. Open a terminal and go to the `<lab_files>/mgmtInterface` directory.

__ b. Enter the following cURL command:

```
curl -H "Content-Type: text/xml" --data-binary
@modify-config.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

__ c. Verify that you get a valid response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <dp:response
         xmlns:dp="http://www.datapower.com/schemas/management">
         <dp:timestamp>2018-03-27T14:14:57-04:00</dp:timestamp>
         <dp:result>OK</dp:result>
      </dp:response>
   </env:Body>
</env:Envelope>
```

__ d. The `<dp:result>OK</dp:result>` confirms the modification of the DataPower object.

__ 6. Log on to the DataPower web management page to confirm that you can now log on to the additional application domains by using the `student`*nn* account.

__ a. Enter the following URL into a web browser:

```
https://<dp_internal_ip>:<dp_WebGUI_port>/dp/login.xml
```

__ b. Enter the user name, `student`*nn*, and your password.

__ c. Select the appropriate domain (`student`*nn*`_import_domain` or `student`*nn*`_remote_domain`) and click **Login**.

__ d. Verify that you are successfully logged in.

__ 7. Save the configuration on the DataPower gateway.

__ a. Send the `saveConfig.xml` file by using cURL.

```
curl -H "Content-Type: text/xml" --data-binary @saveConfig.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

__ 8. Verify that you get a successful response.

__ 9. Create a backup of your `student`*nn* domain. A backup creates an archive of the specified configuration for the domain. The import process that was used in earlier steps is the reverse of a backup.

__ a. Using gedit, edit the supplied template file that is called: *<lab_files>*`/mgmtInterface/backup-domain.xml`

__ b. Change the **nn** in the file to the correct student number.

__ c. Save and close the file.

__ d. Send the `backup-domain.xml` file by using cURL.

```
curl –H "Content-Type: text/xml" --data-binary
@backup-domain.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/current -u
<dp_admin_login>:<dp_admin_password> -k
```

__ e. Examine the response message that is returned from the backup domain operation. The SOAP body of the message contains the contents, but it is base64-encoded. You must use a tool to decode the message and save the file with an XML extension.

Use the Appliance Management Protocol (AMP)

In this section, you use a different part of the XML management interface, the Appliance Management Protocol (AMP). Although it is a SOAP protocol like XML Management Interface, it has a different WSDL and a different set of operations that it supports.

2.8. Use AMP to interrogate the gateway

In this section, you work with another SOAP interface for the XML Management Interface: Appliance Management Protocol (AMP). Only one WSDL file and one XSD file describe the AMP interface:

- `app-mgmt-protocol-v3.wsdl`: The WSDL file that defines the services that are available through the AMP interface.
- `app-mgmt-protocol-v3.xsd`: The schema file that defines the operations that can be sent in XML Management Interface requests.

These files are also contained in the `store:///` directory of the gateway.

The AMP WSDL and XSD are considered to be compatible across firmware releases, but different versions of the WSDL and XSD exist. The name of the file indicates the version. For DataPower Version 7.6, the AMP version is 3.0.

__ 1. OPTIONAL: Open the WSDL and XSD files to review the types of AMP requests that are defined.

As before with the XML Management Interface WSDL and XSD files, you can use a text editor or a web browser to review the files. The files are in the *<lab_files>*/mgmtInterface/DP/ directory.

__ 2. Create an AMP request to retrieve the device information.

__ a. Using File Manager, go to the *<lab_files>*/mgmtInterface directory.

__ b. Right-click the **getdeviceinfo.xml** file and click **Open With > gedit**.

__ c. The DataPower get device information request is provided for you:

```
<dp:GetDeviceInfoRequest
xmlns:dp="http://www.datapower.com/schemas/appliance/management/3.0"/>
```

__ d. Save and close the `getdeviceinfo.xml` file.

__ 3. Send the `getdeviceinfo.xml` file to the AMP interface on the DataPower gateway.

__ a. In a terminal window, go to the `<lab_files>/mgmtInterface` directory.

__ b. In the terminal, run the cURL command:

```
curl -H "Content-Type: text/xml" --data-binary
@getdeviceinfo.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/amp/3.0 -u
<dp_admin_login>:<dp_admin_password> -k
```

Information

The URI for AMP requests is different from the URI for XML Management Interface requests. The URI for AMP requests also varies slightly between the different versions of AMP.

__ c. Verify that you get a response that contains information about the gateway and its features. The specific response that you get depends on the particular gateway you send the request to. However, if you get an error response, look at the information pane on the next page.

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Body>
<amp:GetDeviceInfoResponse
xmlns:amp="http://www.datapower.com/schemas/appliance/management/3.0">
<amp:DeviceName>DP50-IDG-760</amp:DeviceName>
<amp:DeviceSerialNo>0000000</amp:DeviceSerialNo><
amp:DeviceID>5725</amp:DeviceID>
<amp:DeviceType>IDG</amp:DeviceType>
<amp:FirmwareVersion>IDG.7.6.0.5</amp:FirmwareVersion>
<amp:FailureDetected>false</amp:FailureDetected>
<amp:CurrentAMPVersion>3.0</amp:CurrentAMPVersion>
<amp:ManagementInterface
type="web-mgmt">9090</amp:ManagementInterface>
<amp:SecureBackup>enabled</amp:SecureBackup>
<amp:DeviceFeature>MQ</amp:DeviceFeature>
<amp:DeviceFeature>TAM</amp:DeviceFeature>
<amp:DeviceFeature>DataGlue</amp:DeviceFeature>
...
...
<amp:DeviceOperation>Ping</amp:DeviceOperation>
<amp:DeviceOperation>GetToken</amp:DeviceOperation>
<amp:DeviceOperation>Reboot</amp:DeviceOperation>
<amp:DeviceOperation>SetFirmware</amp:DeviceOperation>
<amp:DeviceOperation>Reinitialize</amp:DeviceOperation>
...
...
</amp:GetDeviceInfoResponse>
</env:Body>
</env:Envelope>
```

__ 4. Build an AMP request for the status of your `student`*`nn`* domain.

__ a. Using File Manager, go to the *<lab_files>*`/mgmtInterface` directory.

__ b. Right-click the **getdomainstatus.xml** file and click **Open With > gedit**.

__ c. You see the DataPower get domain status request:

```
<dp:GetDomainStatusRequest
xmlns:dp="http://www.datapower.com/schemas/appliance/management/3.0">
   <dp:Domain>student_nn_domain</dp:Domain>
    </dp:GetDomainStatusRequest>
```

__ d. Change the **nn** in the domain name to the correct student number.

__ e. Save and close the `getdomainstatus.xml` file.

__ 5. Send the `getdomainstatus.xml` file to the AMP interface on the DataPower gateway.

__ a. In the terminal, run the cURL command:

```
curl –H "Content-Type: text/xml" --data-binary
@getdomainstatus.xml
https://<dp_internal_ip>:<dp_xml_mgmt_port>/service/mgmt/amp/3.0 -u
<dp_admin_login>:<dp_admin_password> -k
```

__ b. Verify that you get a response that contains information about your domain. (Sample for student95 is provided here.)

```
<<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Body>
<amp:GetDomainStatusResponse
xmlns:amp="http://www.datapower.com/schemas/appliance/management/3.0">
<amp:Domain name="student95_domain">
<amp:AdminState>enabled</amp:AdminState>
<amp:OpState>up</amp:OpState>
<amp:ConfigState>saved</amp:ConfigState>
<amp:DebugState>true</amp:DebugState>
<amp:CurrentCommand></amp:CurrentCommand>
<amp:QuiesceState></amp:QuiesceState>
</amp:Domain>
</amp:GetDomainStatusResponse>
</env:Body>
</env:Envelope>
```

2.9. REST management

This selection demonstrates some of the capabilities of the REST Management Interface. The exercise uses a Firefox add-on called **RESTClient** to send the REST request and observe the response.

__ 1. First, verify that the REST Management Interface is enabled, and determine which port it is using.

__ a. Use a CLI session (either a session that is still open, or start a new SSH session in a terminal window) to connect to the gateway by using sysadmin access. Log in to the `default` domain.

__ b. Enter `co` to get into configuration mode.

__ c. Enter `rest-mgmt` to get into the mode to work with REST management settings.

__ d. Enter `show` to see the current property values.

__ e. Verify that the **admin-state** is `enabled`, and note the listening **port**.

Important

If the admin-state is `disabled`, redefine it to `enabled`.

__ f. Enter `exit` to leave the REST management configuration mode.

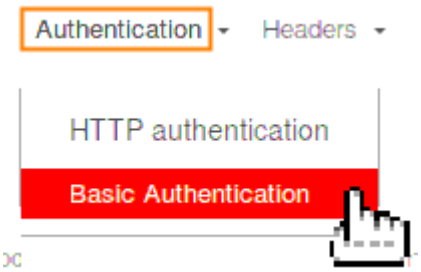
__ g. If you made any changes, enter `write mem` to save the configuration.

__ h. Enter `exit` to leave configuration mode.

__ 2. Issue a REST request to retrieve a list of the currently connected users.

__ a. In the upper-right area of the Firefox browser window, click the **RESTClient** icon.

__ b. A RESTClient tab opens in the browser. You need to set up the user and password to access the REST interface. On the upper left of the RESTClient page, select **Authentication > Basic Authentication**.

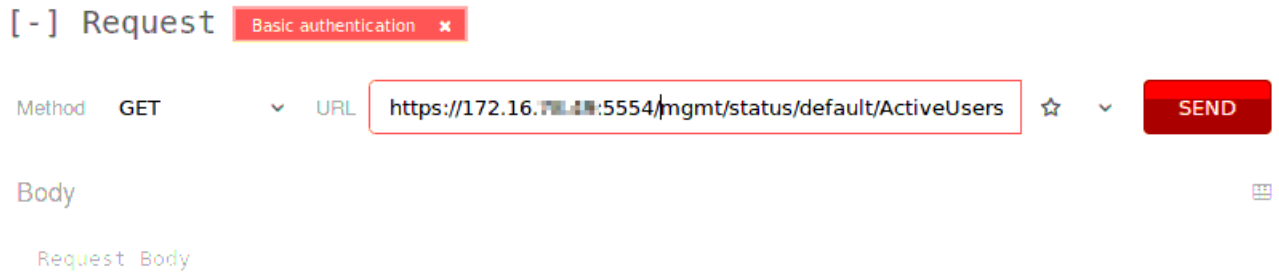


You might need to maximize the browser window to see this option.

__ c. Enter a **Username** of *<dp_admin_login>* and a **Password** of *<dp_admin_password>*

__ d. Click **Okay**. The user and password is sent for you by RESTClient in the HTTP request.

__ e. In the **Request** section of the page, select a **Method** of `GET`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/status/default/ActiveUsers`



Important

Note that the protocol is **HTTPS.**

Be aware of the presence or absence of the trailing slash (“/”) in the URL. It does make a difference.

__ f. Beneath the Method and URL fields, a **Body** field displays for HTTP requests that need a payload. In this case, none is needed.

__ g. On the far right of the page, click **SEND**.

__ h. A “RESTClient call in progress” page temporarily displays in the browser during the request.

__ i. The first time that you call this URL, you might get an invalid security certificate. Click the **Open in a new tab** button, then click the **Advanced** button. Click **Add Exception** in the your connection is not secure dialog box. Finally, click **Confirm Security Exception**.

__ j. In the authentication required prompt, type a **user name** of *<dp_admin_login>* and a **password** of *<dp_admin_password>* and click **OK**.

__ k. Reissue the GET method from the RESTClient tab.

__ l. The RESTClient page refreshes with the response. Be sure to click **Header** to see the returned HTTP headers and HTTP status code.

__ m. You should see a Status Code of **200**. Ignore the x-backside-transport header value of **FAIL FAIL**. It reflects the fact that this request does not call any backside servers to complete its processing.

__ n. Click the **Response** link.

__ o. Instead of the HTTP headers, you now see the JSON response payload. It contains a JSON array of the active users on the gateway, and the user properties.

__ p. Click the **Preview** link.

__ q. You see the same JSON response payload, but slightly better formatted.

__ 3. Issue a REST request to get a list of domains in the gateway

__ a. In the **Request** section of the page, select a **Method** of `GET`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/domains/config/`

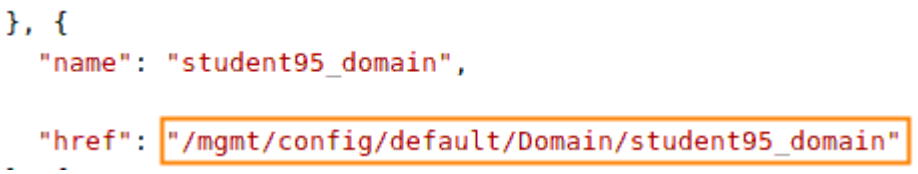
__ b. Click **Send**.

__ c. Select the **Headers** link in the Response section.

__ d. You should see a Status Code of **200**.

__ e. Select the **Preview** link in the Response section.

__ f. You should see a JSON response that contains an array of domains that are defined on your gateway. Search the array for your student domain. Copy the **href** value for use in the next step. The following screen capture shows an example for student95.



__ 4. Retrieve the configuration for your student domain `student`*nn*`_domain`.

__ a. In the **Request** section of the page, select a **Method** of `GET`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/config/default/Domain/student`*nn*`_domain`

Note that the path for this URL matches the **href** that was returned in the “all domains” request previously.

__ b. On the far right of the page, click **SEND**.

__ c. In the **Response** section, click the **Headers** link to see the Status Code, which should be **200**.

__ d. Click the **Preview** link.

__ e. Look at the Domain element to see the properties that are used to define a domain.

__ f. Copy these elements into the text editor to use in the next step.

__ 5. Create a domain

__ a. Edit the Domain properties to make the new domain **name** as `student`*nn*`_REST_domain`,

__ b. Change the **UserSummary** to something different than before.

__ c. The following code snippet shows an example for student95:

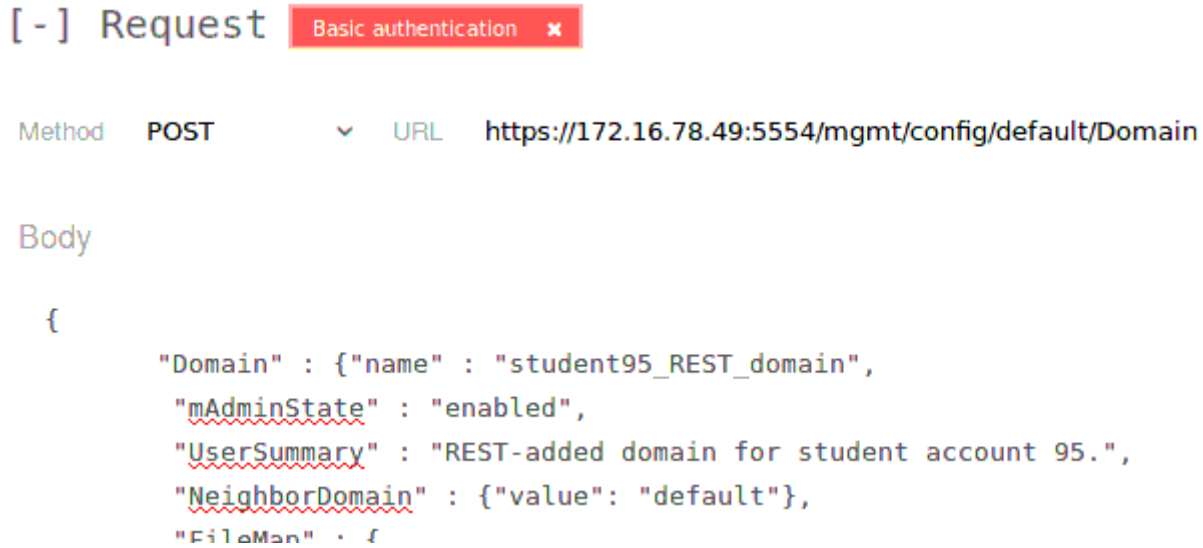
```
{
   "Domain" : {
      "name" : "student95_REST_domain",
      "mAdminState" : "enabled",
      "UserSummary" : "REST-added domain for student account 95.",
      "NeighborDomain" : {"value": "default"},
      "FileMap" : {
            "CopyFrom" : "on",
            "CopyTo" : "on",
            "Delete" : "on",
            "Display" : "on",
            "Exec" : "on",
            "Subdir" : "on"},
      "ConfigMode" : "local",
      "ImportFormat" : "ZIP",
      "LocalIPRewrite" : "on",
      "MaxChkpoints" : 3}
}
```

__ d. In the **Request** section of the page, select a **Method** of `POST`, and a **URL** of `https://`*<dp_internal_ip>*`:`*<dp_rest_mgmt_port>*`/mgmt/config/default/Domain`

__ e. In the **Body** field, paste the Domain element from the text editor to define the new domain.

Important

Because you want to pass the JSON payload, you use the POST HTTP method.



__ f. On the far right of the page, click **SEND**.

__ g. In the **Response** section, the Status Code shows `201 Created`.

__ h. Click the **Preview** link to see the JSON response. You see an href to the new domain, The message `student95_REST_domain": "Configuration was created`.

__ 6. Reenter the URL to return the list of domains in your gateway that you used in a previous step. You should see your new domain.

__ 7. Modify the UserSummary for your new REST domain.

__ a. In the **Request** section of the page, select a **Method** of `PUT`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/config/default/Domain/student`*nn*`_REST_domain/UserSummary`

__ b. In the **Body** field, enter a JSON payload that contains the new value. An example for student95 follows:

```
{ "UserSummary" : "Modified summary for student account 95."}
```

__ c. Click **SEND**.

__ d. In the **Response** section, you should see a Status Code of `200`, and a JSON response payload that says `"UserSummary": "Property was updated."`.

__ 8. Verify that the updated summary is part of the REST domain configuration.

__ a. In the **Request** section of the page, select a **Method** of `GET`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/config/default/Domain/student`*nn*`_REST_domain`

__ b. You can delete the contents of the **Body** field.

__ c. Click **SEND**.

__ d. In the **Response** section, verify that the domain configuration in the response payload has the new text for UserSummary.

__ 9. Save the current configuration.

__ a. In the **Request** section of the page, select a **Method** of `POST`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/actionqueue/default`

__ b. In the **Body** field, enter a JSON payload that identifies the operation to perform:

```
{"SaveConfig" : ""}
```

__ c. Click **SEND**.

__ d. In the **Response** section, you can see the Status Code of `200`, and in the response payload see `"SaveConfig": "Operation completed."`

__ 10. If you want to verify that the domain configuration is saved, you can use another REST request.

__ a. In the **Request** section of the page, select a **Method** of `GET`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/status/default/DomainStatus`

__ b. Click **SEND**.

__ c. In the **Response** section, look in the response payload. Find the **default** domain object. The property value of `"SaveNeeded": "off"` indicates that the default domain is saved. Since the default domain contains the configurations for all of the other domains, it means that the domain configurations are persisted.

__ 11. Delete the REST domain

__ a. In the **Request** section of the page, select a **Method** of `DELETE`, and a **URL** of `https://<dp_internal_ip>:<dp_rest_mgmt_port>/mgmt/config/default/Domain/student`*nn*`_REST_domain`

__ b. Click **SEND**.

__ c. In the **Response** section, the response payload should indicate that your REST domain was deleted. An example for student95 is `“student95_REST_domain": "Configuration was deleted."`

__ 12. To remove the REST domain from the persisted configuration, use the URL from an earlier step to “save configuration” again.

End of exercise

Exercise review and wrap-up

In this exercise, you created and modified domain, user group, and user account objects by using CLI commands and XML Management Interface requests. You imported a service configuration from an HTTP server, and called the new service by using an HTML page from that same HTTP server. You also used AMP requests to interrogate the gateway. You sent REST requests to the REST Management Interface to manipulate configurations on the gateway.

Exercise 3. Using the troubleshooting tools to debug errors

Estimated time

01:00

Overview

The exercise introduces you to the most commonly used troubleshooting tools that are available on DataPower appliances.

Objectives

After completing this exercise, you should be able to:

- Set up and analyze the default system logs
- Configure a multi-step probe to conduct message-level process debugging

Introduction

For unexpected problems, the DataPower appliance offers a range of troubleshooting and debugging tools, from raw packet capture to message-level process debugging. This exercise focuses on three specific DataPower tools:

- Default system logs
- Multi-step probes
- Packet capture

You investigate problems and trace the flow of messages through the DataPower appliance.

The exercise is organized into three parts:

1. **Set up default system logs**: The first section of this exercise demonstrates how to set up and use the default system logs. These logs are important, as they usually are the best place to start a troubleshooting exercise.
2. **Configure the multi-step probe**: The second part of this exercise demonstrates how to use the DataPower multi-step probe problem determination tool. This tool helps analyze the flow and transformation of messages as they are operated at each step of a specific flow.
3. **Review a packet capture file**. The third part of this exercise demonstrates how to view a packet capture file. DataPower offers the ability for the administrator to capture packets that result in a pcap file. This exercise supplies a pcap file and uses the openSource tool Wireshark to view the contents of the pcap file.

Requirements

To complete this exercise, you need:

- Access to the *DataPower* appliance
- Complete the previous exercises (see the Preface section in the Exercise Instructions for details)
- *cURL*, to send requests to the DataPower appliance
- Access to the `<lab_files>` directory
- The Wireshark tool (which is preinstalled on the student image)

Exercise instructions

Preface

- Remember to use the domain and port address that are assigned to you in the exercise setup. *Do not* use the default domain.
- The references in exercise instructions are:
  - *<lab_files>*: **/usr/labfiles/dp**
  - <image_ip>: IP address of the image
  - *<dp_admin_login>*: DataPower secondary administrator user name
  - *<dp_admin_password>*: DataPower secondary administrator password
  - *<dp_internal_ip>*: IP address of the appliance's management interfaces
  - *<dp_public_ip>*: IP address of the public services on the appliance
  - *<dp_WebGUI_port>*: WebGUI port number of the DataPower appliance; the default port is 9090
  - <student*nn*>: Student developer user account
  - <student*nn*_domain>: Student application domain
  - *<mpgw_basic_port>*: The port for the basic multi-protocol gateway (MPGW) that handles validation and transformation, port 10*nn*6

3.1. Import the correct configuration

In this section, you import the configuration of the MPG necessary to complete this lab. The MPG, **MyBasicMPG**, transforms an address request message, `AddressReq.xml`, into an HTML format.

__ 1. Log on to the DataPower Blueprint Console by using a web browser.

__ a. Enter the following URL into a web browser:

`https://<dp_internal_ip>:<dp_WebGUI_port>/dp/login.xml`

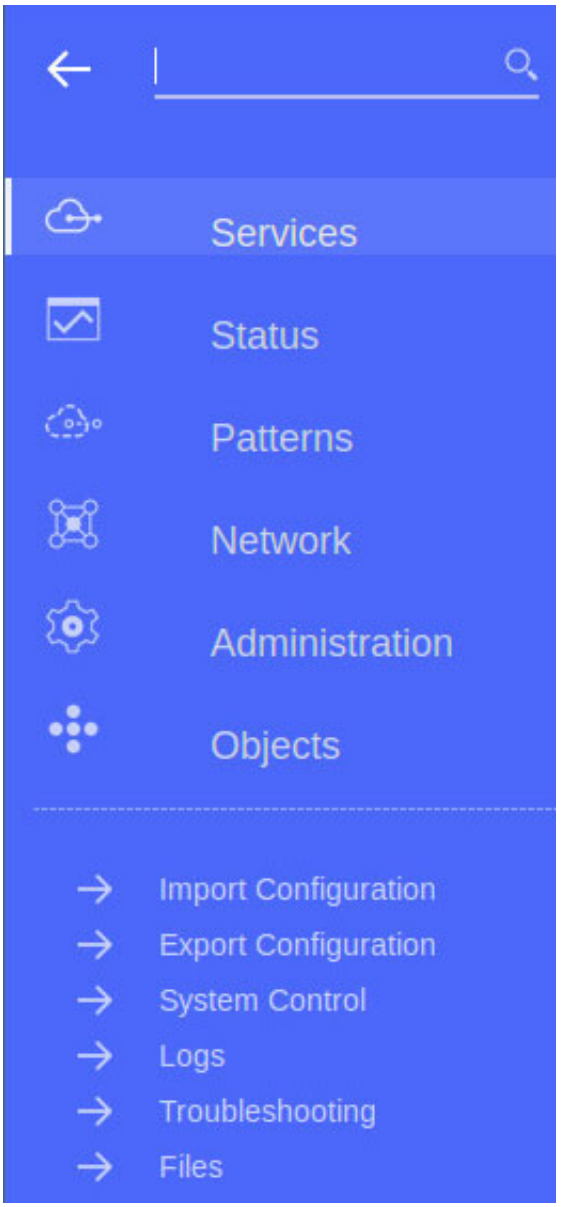
__ b. Enter your `<student<i>nn</i>>` user name, password, and `<student<i>nn</i>_domain>` domain.

__ c. Click **Login**. The DataPower Blueprint Console page is displayed.

__ 2. Import the configuration from the Troubleshooting folder in `<lab_files>`.

__ a. In the Blueprint Console menu, click the **Open menu** icon.

__ b. From the menu, click **Import Configuration**.



__ c. On the Import Configuration page, click **Browse** next to the **File** field.

__ d. Click **Home**, go to *<lab_files>*/`Troubleshooting`, and select `MyBasicMPG.zip`.

__ e. Click **Open**.

__ f. Click **Next**.

__ g. If any objects exist, click **Select All** to ensure that they are all configured properly for this lab.

__ h. Click **Import** at the bottom of the page.

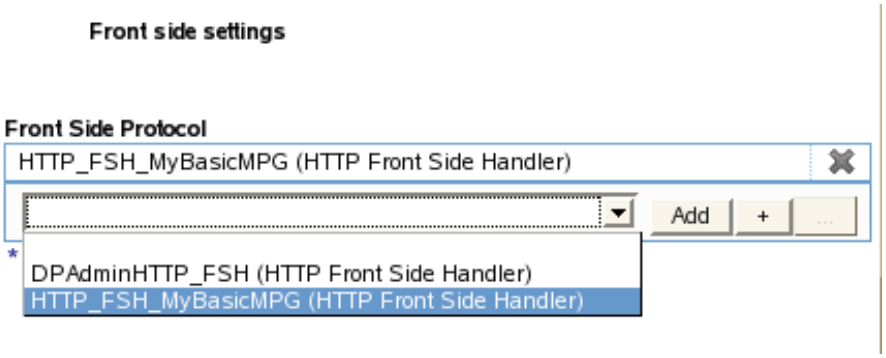
__ i. The results page displays all of the imported objects. Ensure that they are all marked with OK and click **Close**.

__ 3. Change the port number of the MPG to be specific to your student number.

__ a. From the **Services** option, click **Multi-Protocol Gateway**.

__ b. Click **MyBasicMPG**.

__ c. Scroll down to **Front side settings** and select **HTTP_FSH_MyBasicMPG** from the list.

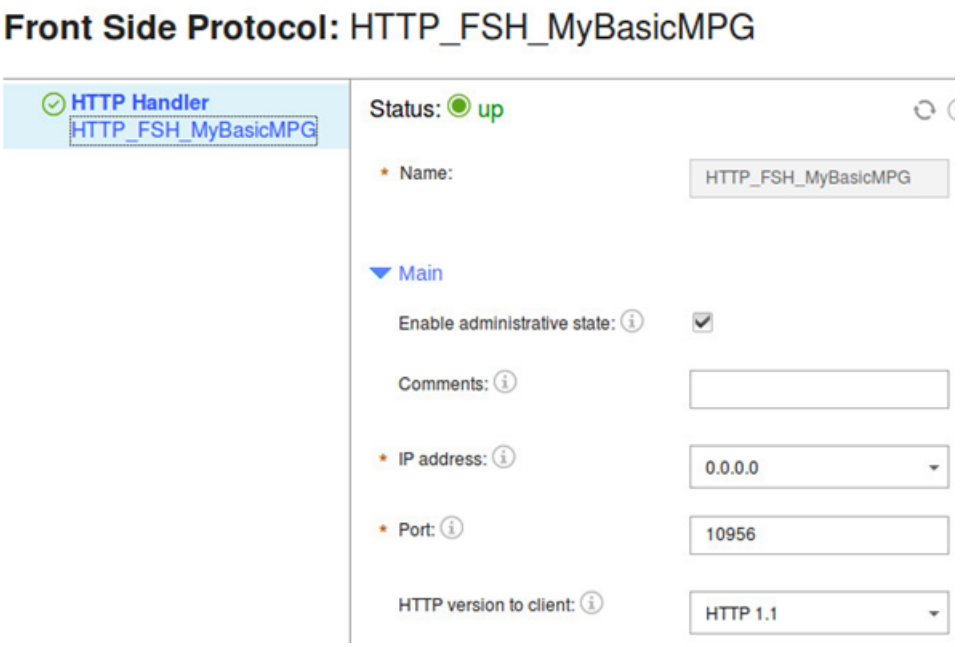


__ d. Click the edit (pencil) button to edit the Front Side Handler.

__ e. Change the Local IP address to *<dp_public_ip>* and the Port Number value to: *<mpgw_basic_port>*

Note

The <*mpgw_basic_port*> is 10nn6.



__ f. Click **Apply** to save the front side handler change, and click **Apply** again to save the changes to the MPG.

__ g. From the top, click **Save changes**.

3.2. Use the default system logs for problem determination

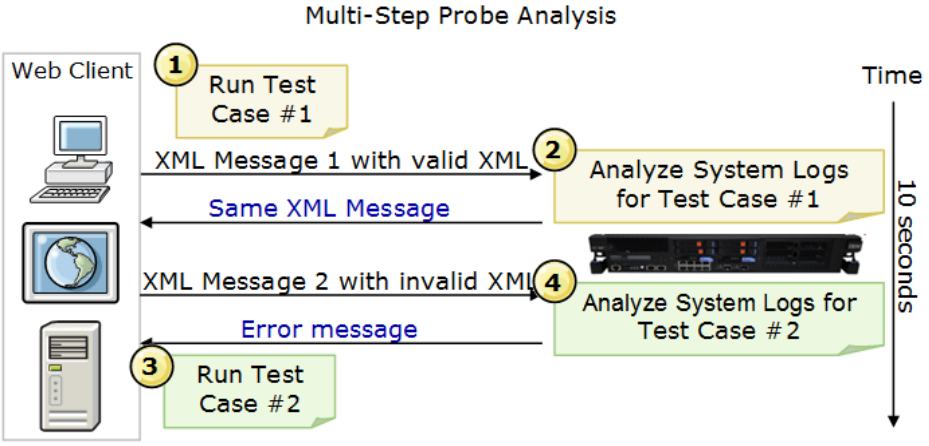
The best debugging tool to use when a problem initially occurs depends on how the appliance is being used then. During development, or during a critical problem discovery phase, changing the default log level to *info* or *debug* on the DataPower appliance can be helpful.

During the development phase, the *default system log*, with log level set to *debug*, is often the best place to start your debugging and troubleshooting process.

In this exercise, you walk through the steps that are required to accomplish this task.

Two test cases are run, and log analysis is conducted thereafter.

- The first test case passes a stable, valid, and well-formed message through the message flow. The logs can be examined to observe the message flow.
- The second test case sends an invalid XML message. The logs must be examined again to trace the message flow to help problem discovery.



This lab implements the scenario that is described in the Multi-Step Probe Analysis figure in the following three sections:

- Configure the system log level to *debug*.
- Run test case 1 and complete log analysis on the system logs.
- Run test case 2 and complete log analysis on the system logs.

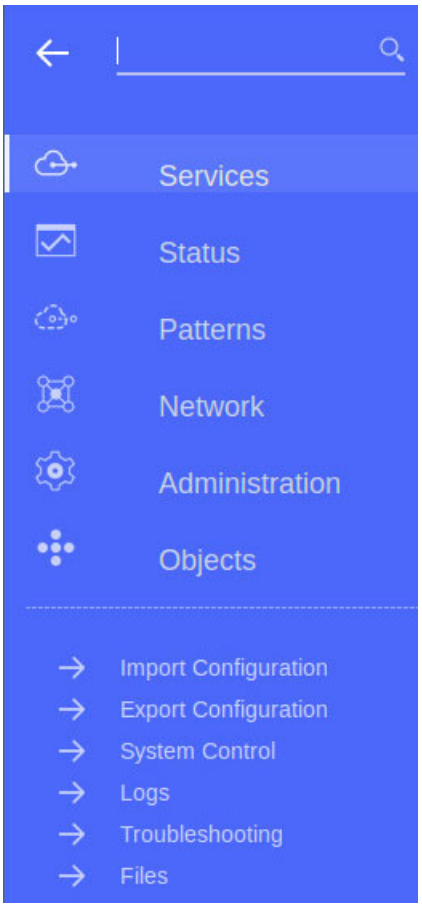
Section 1: Configure the system log level to debug

By default, the system log captures log messages with a severity of error or greater to keep the volume of messages at a minimum. During troubleshooting, it is a good practice to change the log level of the system log to *debug*. This change allows the log to capture more messages and track object processing in greater detail.

__ 1. Change the system log level to *debug*.

__ a. In the Blueprint Console menu, click the **Open menu** icon.

__ b. From the menu, click **Troubleshooting**.



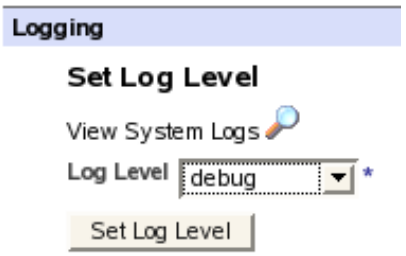
__ c. Make sure that you see the Troubleshooting page; it consists of three tabs:

- **Main** tab: Links to the general high-level debugging panes
- **Debug Probe** tab: Contains the features that are required to set up probes to analyze message processing flows
- **Conformance Validation** tab: Allows a choice of conformance validators to run

In this section, you focus on the system log setup information.

In a later section, you focus on configuring probes to complete a deeper analysis of message processing.

__ d. From the **Main** tab, set the **Log Level** to **debug** and click **Set Log Level**.

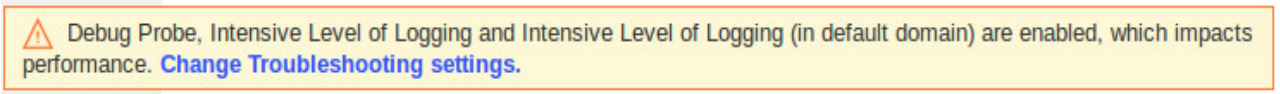


__ e. Click **Confirm** to close the confirmation dialog box.

The system log set to the debug level contains a message for each step of the processing.

__ f. Click **Save changes** to save the configuration.

__ g. A quick way to check whether your log level is set to either **debug** or **info** is to look for a blue “troubleshooting enabled” notice on all DataPower WebGUI pages. Verify that your page has a similar message.



Section 2: Test case 1: Execution and analysis

Use the cURL command line to run the test case.

__ 1. Send a well-formed and valid XML document, `AddressReq.xml`, to the MyBasicMPG multi-protocol gateway.

__ a. Open the `AddressReq.xml` file in a text editor, such as gedit, and review it before you run this test case.

Open a terminal and go to the *`<lab_files>`*`/Troubleshooting` directory.

```
# cd /home/localuser/labfiles/dp/Troubleshooting
```

__ b. Enter the following command in terminal:

```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_internal_ip>:<mpgw_basic_port>
```

__ c. Verify that you get an HTML response similar to what you received when testing earlier. It contains the following text:

```
<address:name>
        <address:title>Mr.</address:title>
        <address:firstName>John</address:firstName>
        <address:lastName>Doe</address:lastName>
</address:name>
```

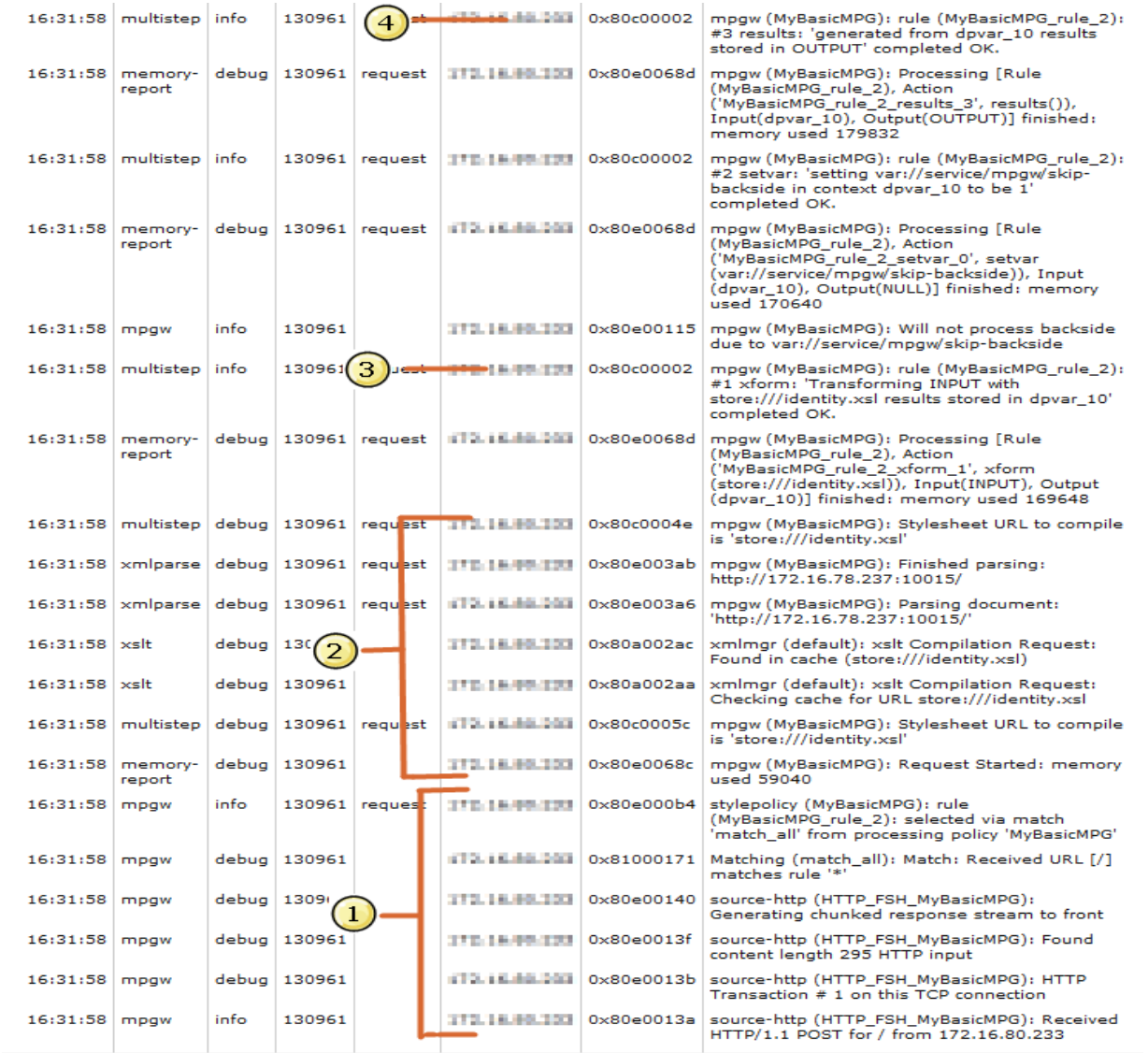
All the service does is return the original XML sent to it. The Transform action uses the `identity.xsl` stylesheet to echo the incoming XML.

__ 2. Review the system logs.

__ a. Click **Status > View Logs > System Logs**.

__ b. Examine the system log that was generated and attempt to reconstruct the path of the message. The file is read from the bottom up. Your message is processed as part of a transaction; it is most likely the transaction at the top (that is, the last one processed). Four key items to look for are:

- Client request
- XML manager classes
- Validation action logging
- Transform action logging



Observations

1) The incoming client POST request to the MyBasicMPG multi-protocol gateway operating on the DataPower appliance address, by using the assigned port number,

is received from the client IP address. If you continue following the trace, observe that a processing match is obtained.

2) The XML manager compiles the `identity.xsl` file.

3) The incoming client request message is transformed by using the identity stylesheet. The action corresponds to the transform action in the multi-step process. The action is completed with a status of OK.

4) The transformed message is then stored in the OUTPUT variable, which is the returned message data, and completed with a status of `OK`.

Information

If you have multiple transactions that run simultaneously, you can have entries from different transactions that are interspersed with each other. A way to see just a list of entries for a single transaction is to click the transaction ID (the `tid` column) for the intended transaction. The process opens another window with just the entries for the selected transaction.

Section 3: Test case 2: Execution and analysis

You now use the cURL command-line to run the test case that fails.

__ 1. Edit the `AddressReq.xml` file to force an XML well-formedness error.

__ a. You can accomplish this task in many ways. A simple way would be to change the `title` element to: `title1`

Be sure to change *only* the start tag and not the end tag. The transmission of this invalidly formatted XML message causes the XML parsing to fail.

__ 2. Send the invalid XML document to the multi-protocol gateway.

__ a. Open a terminal and go to the *<lab_files>*`/Troubleshooting` directory.

__ b. Enter the following command in the terminal:

```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_public_ip>:<mpgw_basic_port>
```

__ c. You get a fault string of `Internal Error` as the response.

```
<faultstring>Internal Error (from client)</faultstring>
```

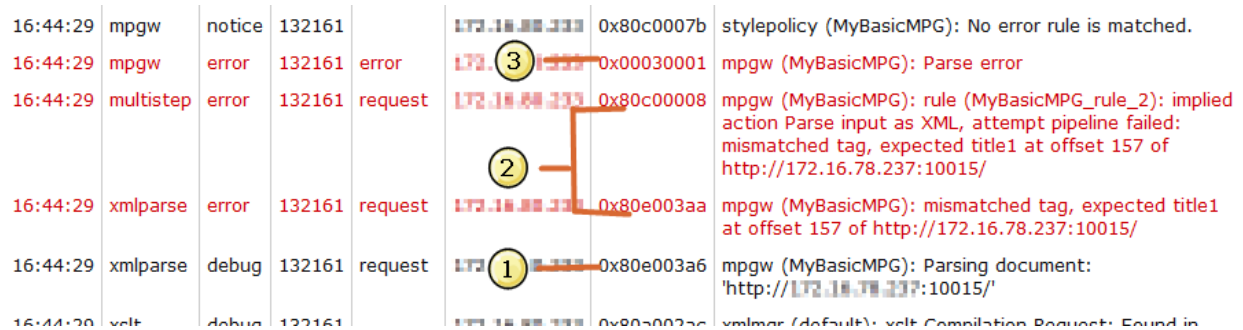
__ 3. Review the system logs.

__ a. Click **Refresh** on the System Log page to refresh the list.

__ b. Attempt to reconstruct the path of the message through the flow. Key items to look for are:

- Client request
- XML manager compilation classes

- Schema validation failure



Note

Your log might be different, depending on the method you chose to invalidate the XML message.

1. Parsing began on the message.
2. Notice that processing failed because of an end tag `title` that is encountered when it expected `title1`.
3. The parse error is generated.

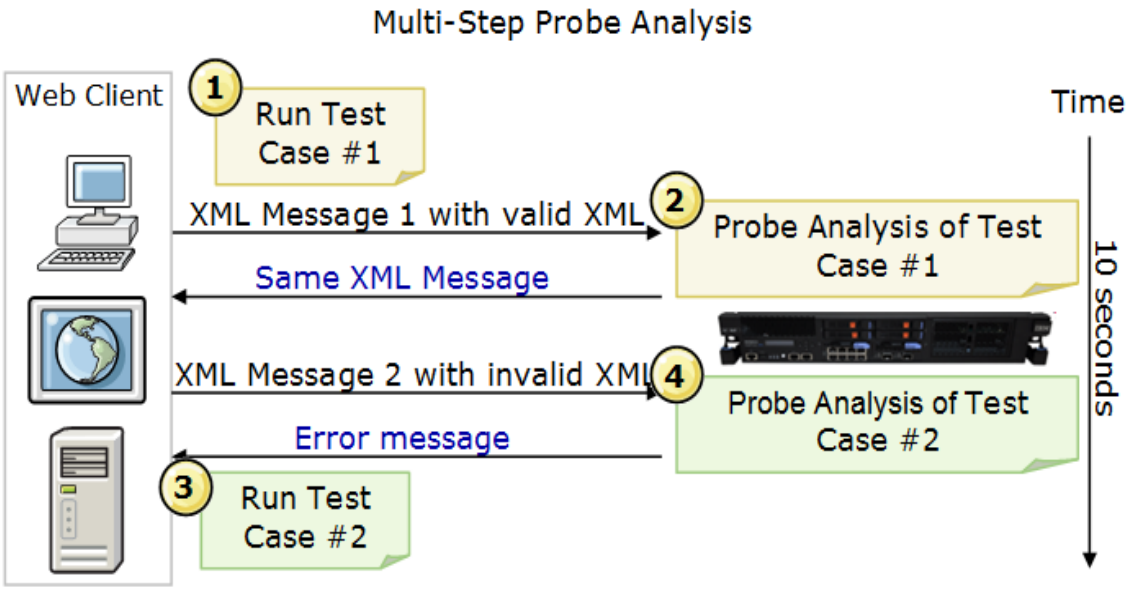
__ c. Remove your changes to `AddressReq.xml`.

__ d. Save and close the file.

3.3. Use the multi-step probe to debug message flows in DataPower

The multi-step probe displays the lifecycle of a message as it passes through the appliance. More specifically, this tool displays the contents of variables at each step of a document processing rule, which usually involves many steps.

The probe must be enabled for a specific service.



This exercise implements the scenario that is described in the Multi-Step Probe Analysis figure in the following three sections:

- Configure the multi-step probe
- Run test case 1 and complete probe analysis
- Run test case 2 and complete probe analysis

Test case 1 sends a valid message to the service.

Test case 2 simulates an invalid message to the service. The message causes an error during message processing. The multi-step probe is used to complete a deep analysis to demonstrate how problem determination can be carried out in a real scenario.

Section 1: Configure the multi-step probe

__ 1. To use the multi-step probe, you must verify whether the multi-step probe is already configured on the service that is under testing. If not, you must enable the multi-step probe before you can send data for testing or debugging purposes. The steps that follow walk you through this verification and configuration process.

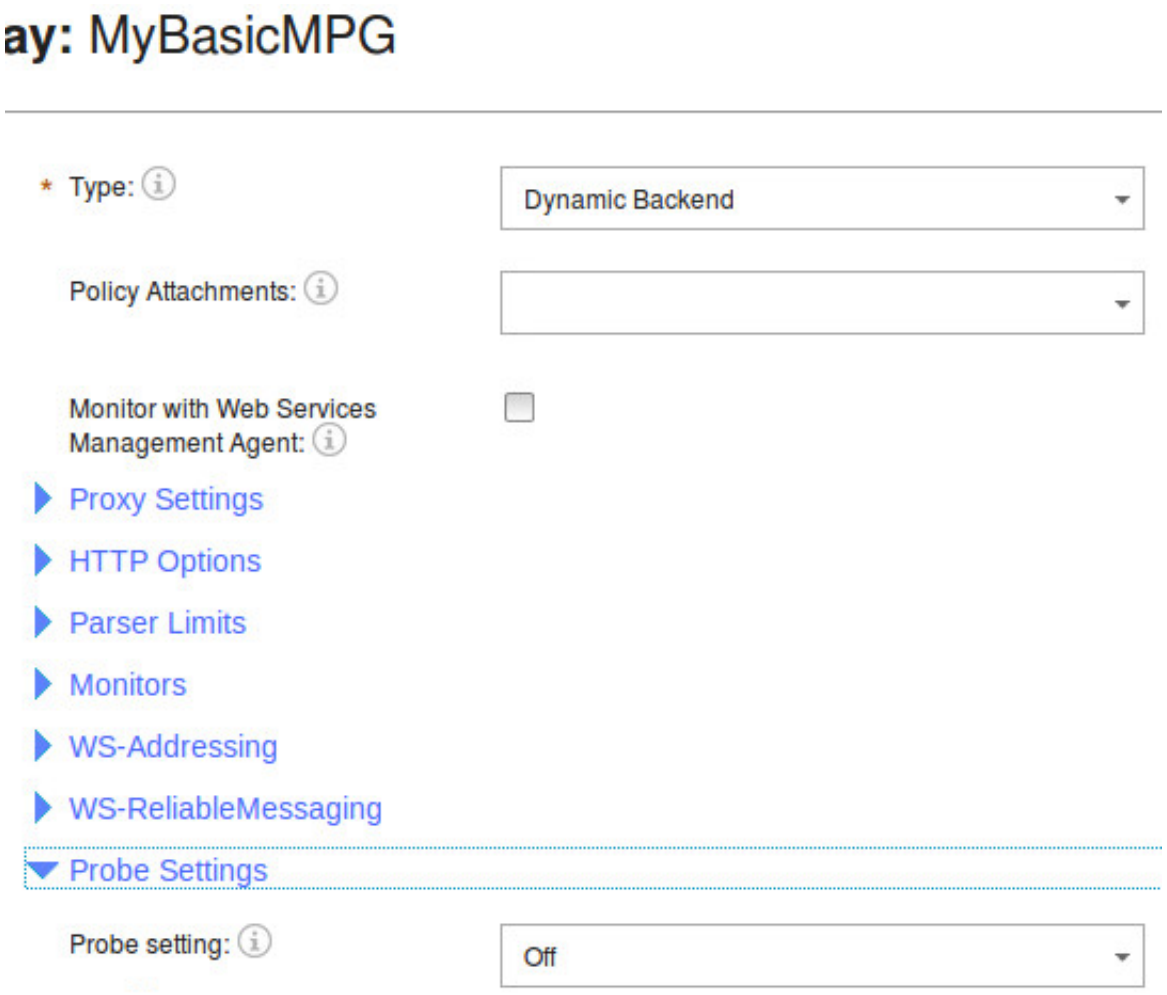
__ a. From the Control Panel, go to **Objects > Service Configuration > Multi-Protocol Gateway**. You are taking a different path to edit the MPG. Definition pages that are shown from the Objects menu sometimes show more detail than from the items that are selected from the Control Panel.

__ b. In the list of configured multi-protocol gateway services, make sure that the **MyBasicMPG** service does not have the **Probe** field that is populated with the magnifying glass.

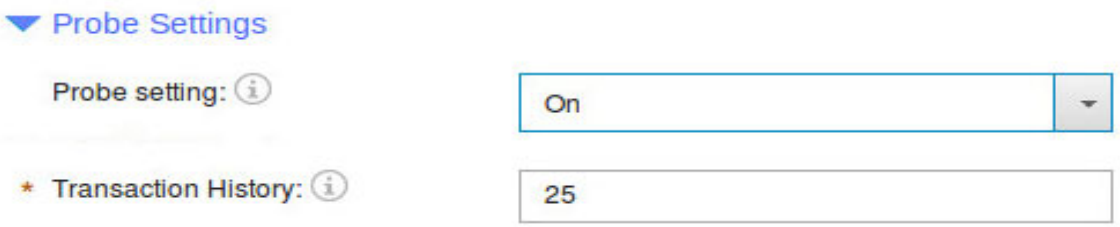
__ c. Click the **MyBasicMPG** service. You now enable a multi-step probe for this proxy.

__ d. In the Configure Multi-Protocol Gateway Service page, complete the following tasks (in this order).

1) Expand the **Probe Settings** link.



2) Select **On** to turn on the probe setting.

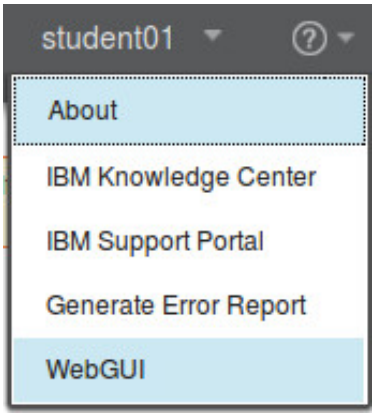


3) Click **Apply** to close the probe settings window. The probe is now enabled.

Note

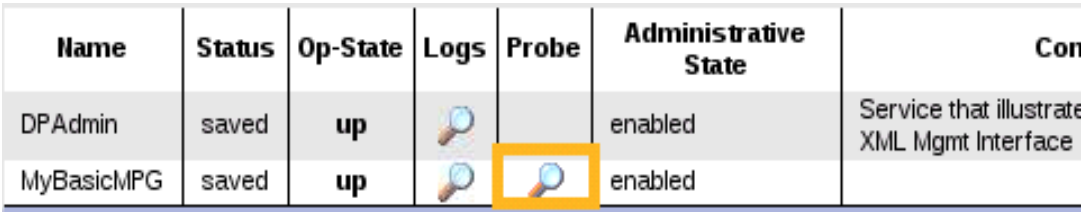
Restarting the appliance automatically disables the probe.

__ e. You can validate that the probe is running by opening the web GUI from the help drop-down.



__ f. The Web GUI opens in a separate tab in the browser.

__ g. From the Web GUI Control Panel, select **Objects > Service Configuration > Multi-Protocol Gateway Service** to display the catalog of service objects defined. Verify whether the probe is operational. You see a magnifying glass beside the **MyBasicMPG** service.



Name	Status	Op-State	Logs	Probe	Administrative State	Con
DPAdmin	saved	up			enabled	Service that illustrate XML Mgmt Interface
MyBasicMPG	saved	up			enabled	

__ h. Close the Web GUI tab and return to the Blueprint Console.

Note

The Web GUI is deprecated in DataPower V7.6.

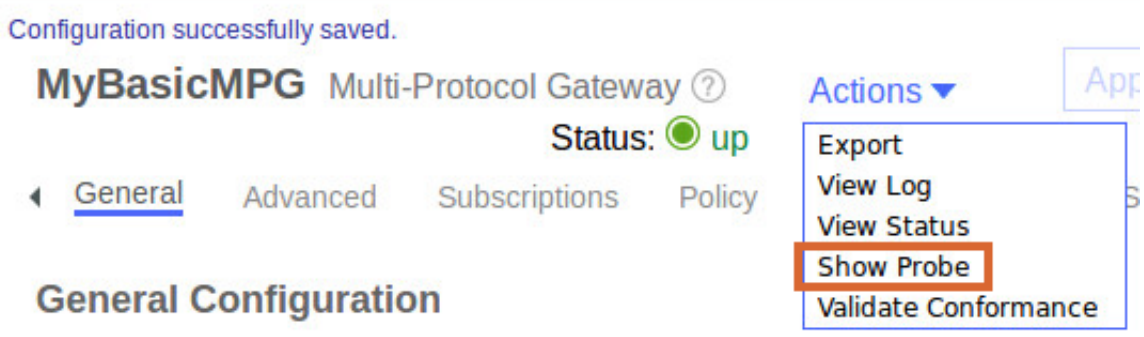
Section 2: Test case 1: Execution and analysis

__ 1. Open the multi-step probe transaction list window.

__ a. From the Blueprint Console, click **Services**, then click **Multi-Protocol Gateway**.

__ b. Click the **MyBasicMPG** gateway link.

__ c. From the MyBasicMPG page, from the **Actions** list, click **Show Probe**.



A new window opens to contain the Multi-Step Probe Transaction List. The list is empty.

__ 2. Using the cURL command-line tool, generate a transaction that you can use to trace.

__ d. Open a terminal and go to the `<lab_files>/Troubleshooting` directory.

__ e. Enter the following command in the terminal:

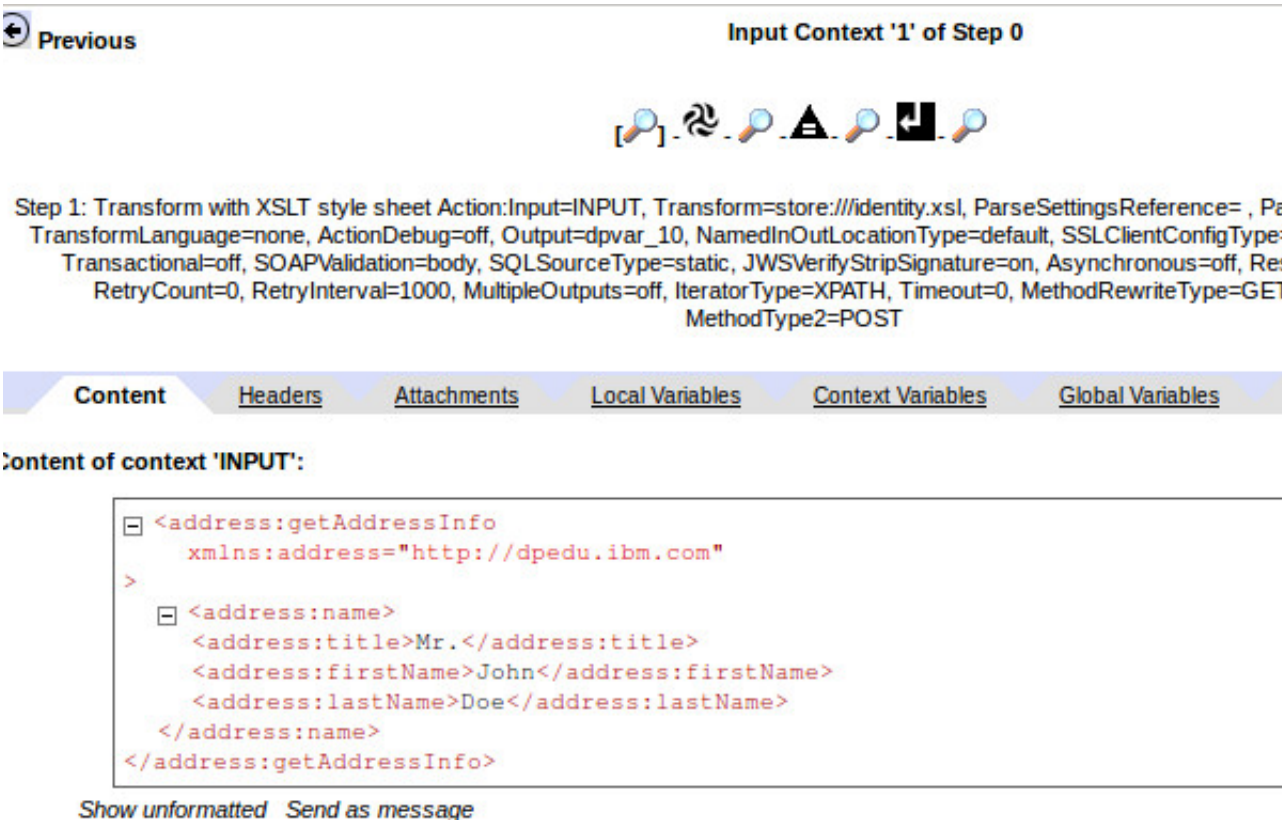
```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_public_ip>:<mpgw_basic_port>
```

__ f. An XML document that contains information for “John Doe” is returned.

__ g. Click **Refresh** in the Multi-Step Probe Transaction window.

__ h. Click the magnifying glass icon of the Probe transaction list to display the details of the transaction.

__ i. The Probe Display window is shown. First, look at the top of the window.



__ j. Note the document processing policy icons in the center of the Transaction Context window. It shows a transform action, then a set variable action, and then a results action. You see these actions when you hover over the icons in the transaction window. The Match is not shown because you would not be in the probe unless the match were not satisfied. In the probe display, you are examining the processing through a specific rule within the policy. These icons are called the input and action detail selector icons.

__ k. The leftmost magnifying glass has brackets that surround it. That indicates where in the rule you are currently positioned. The text beneath (“Step 1”) shows the context of your current position.

__ l. You can move to different positions in the rule by using either the **Next** or the **Previous** icon near the top of the window, or by selecting a specific magnifying glass. In either case, the brackets move to the current magnifying glass icon.

__ m. Now examine the bottom half of the window.

__ n. The **Content** tab is selected. It shows the document context for the current position in the rule. This content is XML-formatted. If the document is not XML, you can click the **Show unformatted** link to open a new window that shows the raw text of the document.

__ o. The **Headers** tab shows the HTTP headers for the transaction. The **Attachments** tab shows information about any attachments to the document. The **Variables** tabs allow you to examine variable settings.

__ p. The left and right arrows are used to scroll within the tabs when they are wider than the window.

__ q. Click the center magnifying glass (input selector icon). Examine the content (context “INPUT”) to see what is being passed to the Transform action.

__ r. Now use the **Next** icon to move to the output phase of the transform. In the **Content** tab, examine the results of the transform. Because this task is a simple identity transform, it looks like the input document. The results are placed in a context variable `dpvar_10`. Feel free to look at the unformatted content.

__ s. Close the Probe Display window.

Section 3: Challenge test cases: Execution and analysis

__ 1. Inject problems into the processing of the message. You alter the input message to cause a parse error, as before. You then use the probe to debug the error.

__ a. Edit the `AddressReq.xml` file so that it invalidates the XML creating a mismatched tag.

A simple way would be to change the `title` element to `title1`.

Be sure to change only the start tag and not the end tag. The transmission of this invalid XML-formatted message causes the schema validation action to fail.

__ b. Resubmit the same command as in the previous test case:

```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_public_ip>:<mpgw_basic_port>
```

__ c. You get a result of `Internal Error`.

__ d. On the Multi-Step Probe Transaction List, click **Refresh**.

__ e. Click the **magnifying glass** icon for the new transaction. The entry is displayed in red to indicate that some problem exists.

__ f. The probe display opens. The transaction stopped in Step 0, so the transform did not occur. Remember that you are looking at what happened, rather than what is configured in the rule. Click the **magnifying glass**.

__ g. Click the **Service Variables** tab (you can click the right arrow to scroll within the tab).

Service Variables:

name	type	value
var://service/client-service-address	string	'9.65.230.170:4300'
var://service/conformance/policyname	string	*(empty string)*
var://service/connection/note	string	*(empty string)*
var://service/current-call-depth	string	'0'
var://service/domain-name	string	'student99_domain'
var://service/error-code	string	0x00030001
var://service/error-headers	string	'HTTP/1.1 500 Internal Server Error Content-Type: text/xml X-Backside-Transport: FAI
var://service/error-ignore	string	'0'
var://service/error-message	string	'mismatched tag, expected title1 at offset 157 of http://9.26.56.37:10996/'
var://service/error-subcode	string	0x00030001

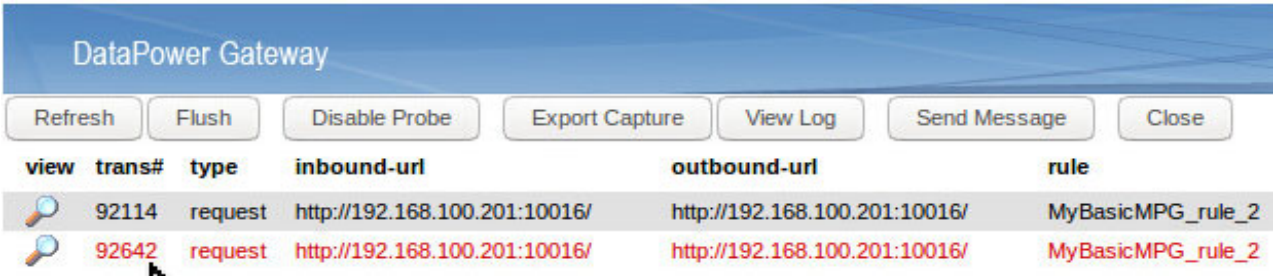
__ h. Look for the variable `var://service/error-message`. Its value is the message that is received in the terminal window.

__ i. Look for the variable `var://service/error-code`. This result corresponds to a “parse error.”

__ j. Look for the variable `var://service/error-subcode`. Sometimes, `error-subcode` indicates the original error, and `error-code` indicates a more general, overriding error. In this case, the codes are the same.

__ k. Look for the variable `var://service/transaction-id` (towards the bottom of the list of variables). Its value is the ID of the failed transaction. You can use this value to correlate to the system log.

__ l. Back on the Multi-Step Probe Transaction List, click the **trans#** of the failed transaction.



__ m. A system log that contains the log entries for just this single transaction is displayed. (You can also open a standard view of the system log and look for the failed transaction number.)

If you do not see any entries in this log, send another `curl` command and click the transaction number of the new message.

__ n. Return `AddressReq.xml` back to normal by removing the changes added to force a failure. Save and close the file.

__ o. Click **Disable Probe** from **Objects > Service Configuration > Multi-Protocol Gateway > MyBasicMPG > Probe Settings** to turn off the multi-step probe.

__ p. Click **Close** to close the action complete window.

__ q. Click **Close** to close the multi-step probe window.

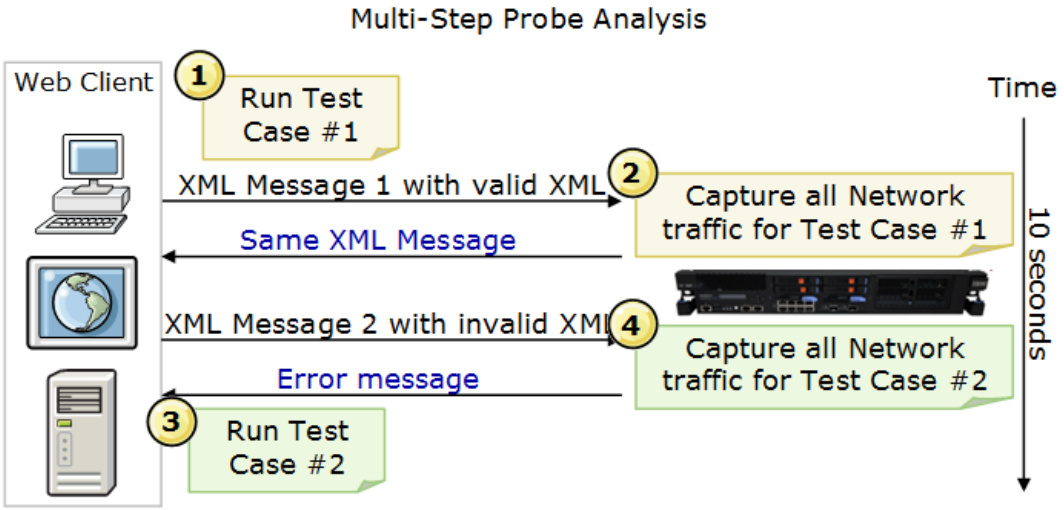
__ r. Log out as `<student`*`nn`*`>` leaving the browser open to the DataPower appliance.

3.4. Use Wireshark to view a packet capture file for debugging purposes

DataPower offers administrators the opportunity to capture packets that come into and leave the DataPower appliance. This ability to monitor traffic is captured in a pcap file. Administrators must supply a tool for viewing the pcap file. In this section of the exercise, students view a precaptured pcap file by using the network protocol analyzer tool, Wireshark.

Wireshark is open source software which can be downloaded and used for free. The license under which Wireshark is issued is the GNU General Public.

For this exercise scenario, the DataPower packet capture tool was turned on for 10 seconds, capturing all network traffic to and from the DataPower Appliance. During this time, test case 1 ran, submitting a valid XML message to the appliance and receiving an expected XML response. Test case 2 then ran, submitting an invalid XML message to the DataPower appliance and receiving a returned error message. Both test cases were completed in the 10-second window and captured in the `capture.pcap` file, which is analyzed during this exercise.



This exercise implements the scenario that is described in the Multi-Step Probe Analysis figure in the following three sections:

- Review how to enable packet capture on DataPower
- Review the basics of the Wireshark tool
- Analyze a precaptured pcap file

Test case 1 sends a valid message to the service.

Test case 2 simulates an invalid message to the service. The message causes an error during message processing. The Wireshark tool is used to complete a packet analysis to demonstrate how problem determination can be carried out in a real scenario.

Attention

Packet capture gathers all network traffic information. User login and password text are also captured. Therefore, many companies have a security policy in place that does not allow the use of packet capture, or analysis of such data. Ensure that customer or company policy is not being violated before using this technology, as it can be grounds for termination.

If the class is taught as a private or custom course, this portion of the exercise does not violate any monitoring network traffic policy. This exercise is using a precapture pcap file, not a live run time.

Section 1: View how to enable package capture

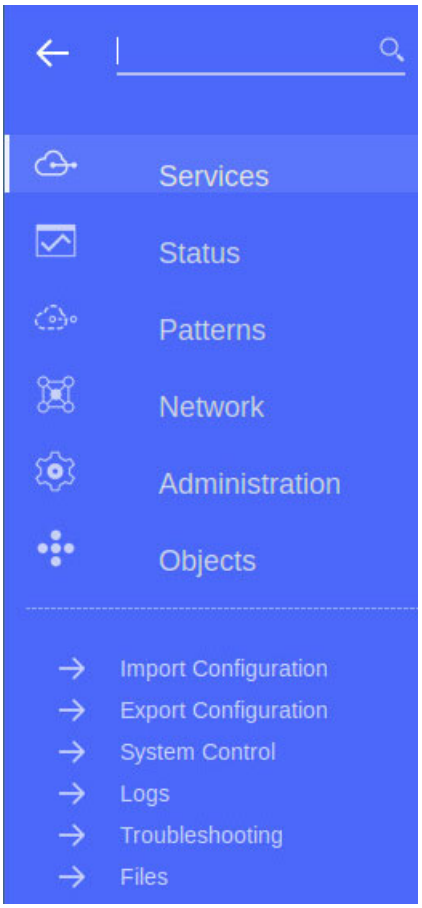
__ 1. Log on to the DataPower appliance as the administrator
Log on to DataPower with `<dp_admin_login>` and select the default domain.

__ 2. View the Packet Capture section.

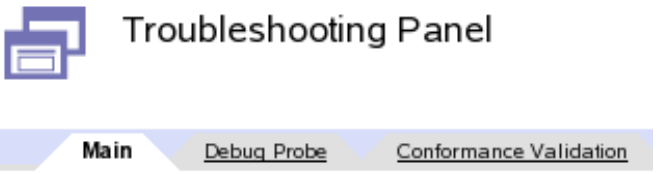
__ 1. Change the system log level to *debug*.

__ a. In the Blueprint Console menu, click the **Open menu** icon.

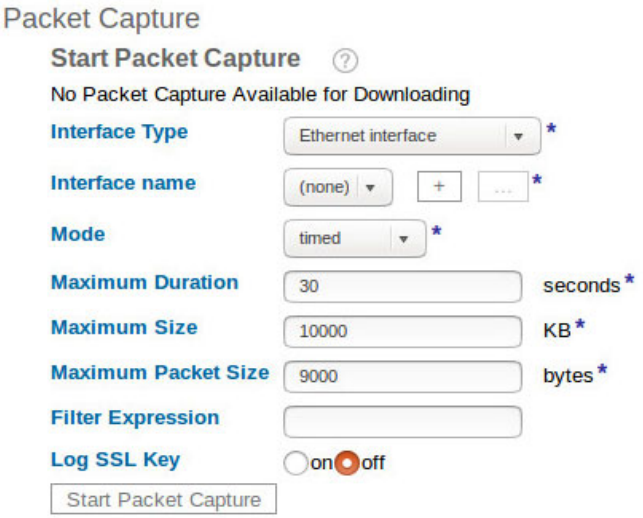
__ b. From the menu, click **Troubleshooting**.



__ c. Make sure that you see the Troubleshooting page, and the **Main** tab is selected.



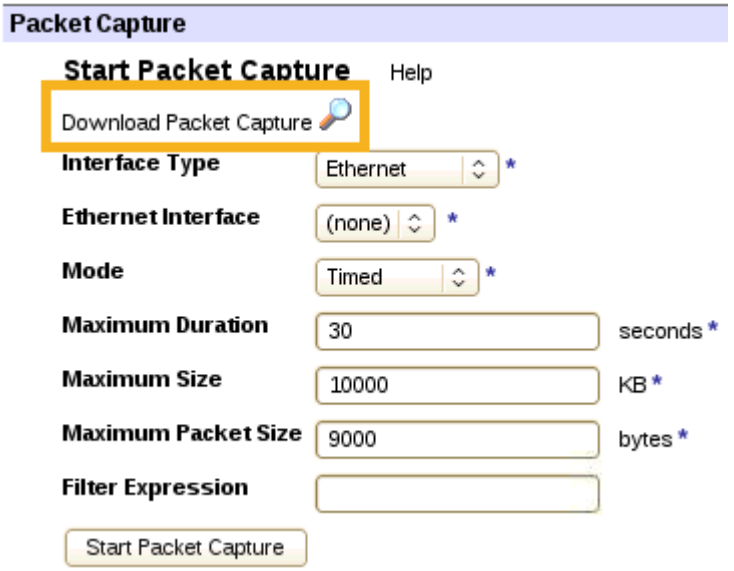
__ d. Scroll down to the **Start Packet Capture** section.



__ e. The Packet Capture consists of the following options:

- **No Packet Capture Available for Downloading**: This text is shown when no packet capture file is available for downloading. If a file is available, then a magnifying glass is shown and not the text. You click the magnifying glass to download the file.
- **Interface Type**: The type of interface to enable Packet Capture on. The options are all interfaces, Ethernet interface, link aggregation interface, Loopback interface, Standalone interface, or VLAN interface.
- **Interface name**: This choice identifies which specific interface object is to be monitored. The choices that are presented depend on the interface type selected.
- **Mode**: The timing mode for the packet capture. The mode can be continuous or timed mode. In timed mode, the packet capture ends after the defined duration.
- **Maximum Duration**: The maximum duration of Packet Capture in timed mode, which is represented in seconds.
- **Maximum Size**: The maximum size of Packet Capture, which is represented in KB.
- **Maximum Packet Size**: The maximum size of each packet that is recorded during a packet capture session. Enter any value of 20 - 9000 bytes. The special value -1 invokes the default 9000 bytes, which correspond to the maximum transmission unit for Ethernet interfaces and therefore allows the entire frame to be saved.
- **Filter Expression**: The packet capture filter expression is a string that defines how the packet capture must be filtered. Full details of the syntax can be found by searching the web for “pcap-filter(7) format”.

__ f. When a packet is captured, the Download Packet Capture becomes visible and is shown with a magnifying glass. To download the pcap file, click the magnifying glass and save the file to a location on the computer that is being used.



__ g. Log out of the DataPower appliance.

Section 2: Load the pcap file in the Wireshark tool

__ 1. Run the Wireshark tool.

__ a. From the Ubuntu desktop, double-click the **Search** icon.

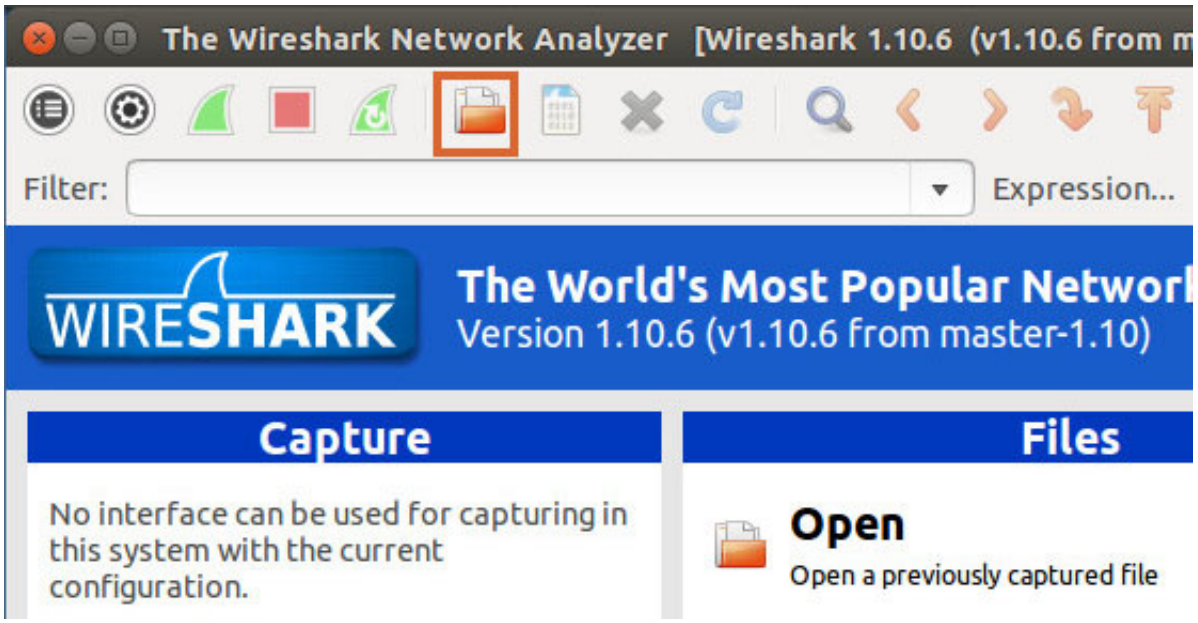
__ b. Type `Wireshark` in the search field.

The icon is displayed in the applications list.

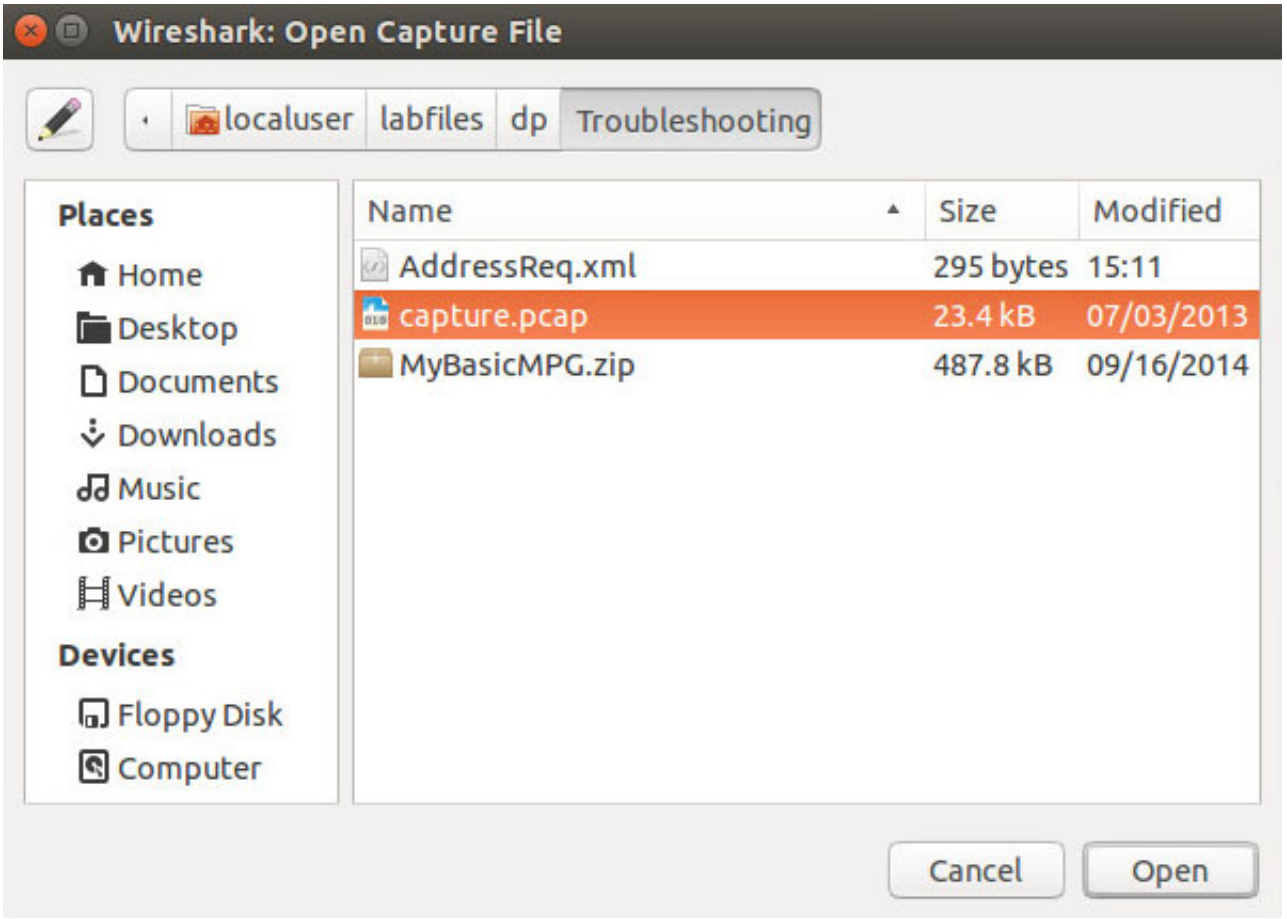
__ c. Double-click the **Wireshark** icon to open Wireshark.

__ 2. Load the pcap file available in `<lab_files>/Troubleshooting`.

__ a. From within the Wireshark tool, click the **Open File** icon.

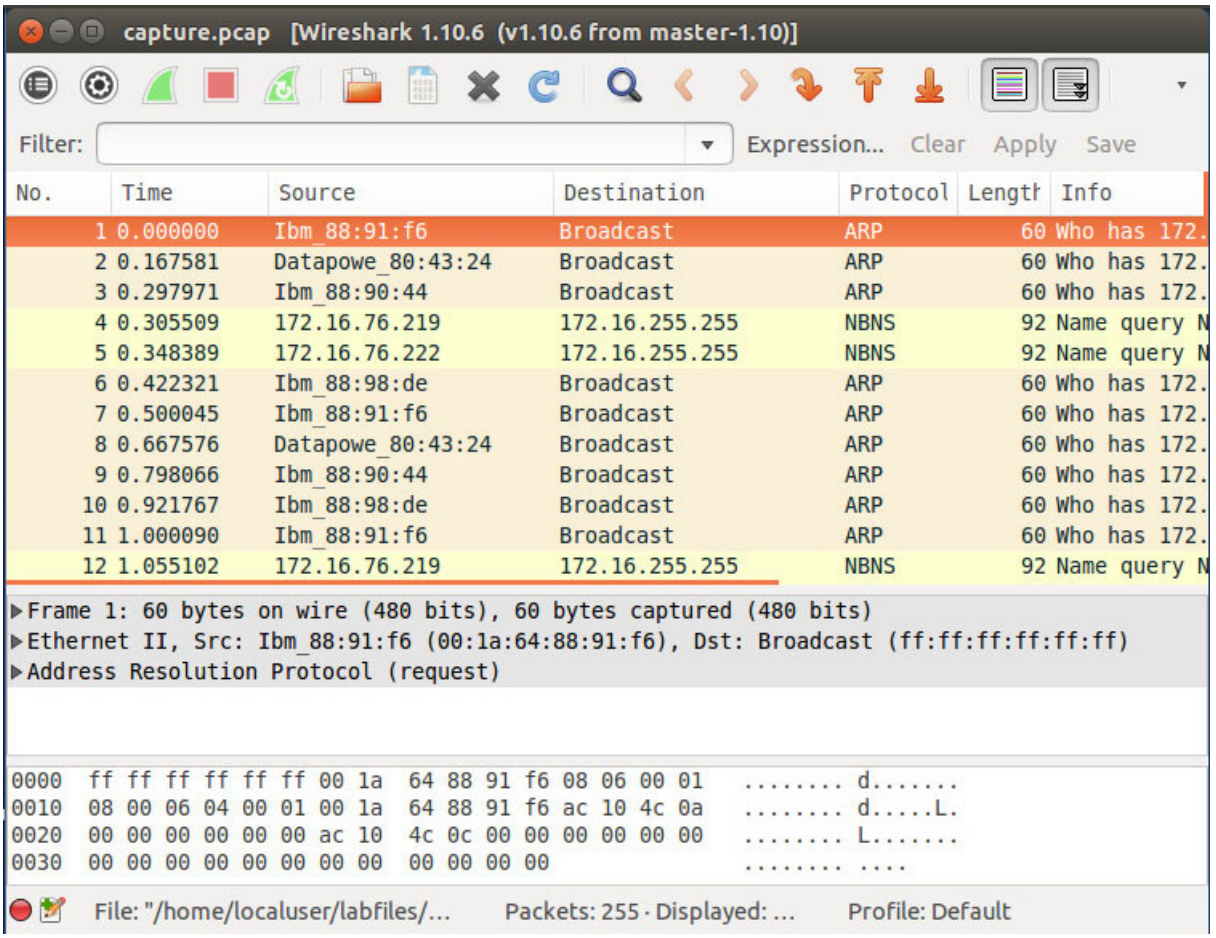


__ b. Using the Open File browser, go to the *<lab_files>*/Troubleshooting subdirectory, and select the **capture.pcap** file.



__ c. Click **Open** to open the capture.pcap file.

__ d. Familiarize yourself with the contents of the Wireshark tool with a loaded pcap file.



- **No**: This column shows the number of the packet that is captured. This number is used to located specific packets during the rest of this exercise.
- **Time**: This column identifies the time that the packet was captured.
- **Source**: This column shows the source location where the packet is coming from.
- **Destination**: This column shows the target destination where the packet is going to.
- **Protocol**: This column shows the protocol of the packet. There are currently 724 supported protocols and media in the version of Wireshark being used.
- **Info**: This column shows the information details about the packet.

Section 3: Test case 1: Analysis of successful XML message

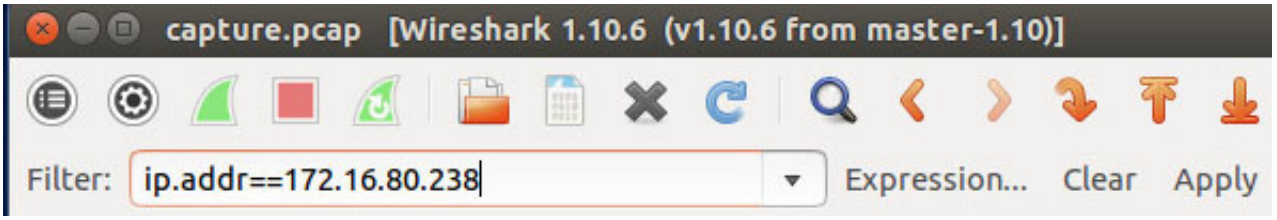
Information

The pcap data file that is used in this section might use a different client IP address and DataPower IP address to your system. Since you are using the Wireshark tool to display a previously captured file, do not worry that these IP addresses are different from the IP addressed on your system.

__ 1. Set a filter to view messages that are related to the IP address of interest.

The pcap data file being used is captured by using a client IP address of 172.16.80.238 and a DataPower IP address of 172.16.78.29. Therefore, to reduce the number of packets in the Wireshark window, a filter is set to display the packets that have a source or destination IP address of the client (172.16.80.238).

__ a. Enter `ip.addr==172.16.80.238` in the **Filter** field.



Click **Apply**.

Notice how the window now displays packets in which either the source or the destination IP address is 172.16.80.238.

No.	Time	Source	Destination	Protocol	Length	Info
51	4.176623	172.16.80.238	172.16.78.29	TCP	74	48921
52	4.176640	172.16.78.29	172.16.80.238	TCP	74	10955
53	4.176821	172.16.80.238	172.16.78.29	TCP	66	48921
54	4.177371	172.16.80.238	172.16.78.29	HTTP/XML	564	POST
55	4.177382	172.16.78.29	172.16.80.238	TCP	66	10955
56	4.178070	172.16.78.29	172.16.80.238	TCP	194	[TCP
57	4.178094	172.16.78.29	172.16.80.238	TCP	109	[TCP
58	4.178098	172.16.78.29	172.16.80.238	TCP	68	[TCP
59	4.178102	172.16.78.29	172.16.80.238	TCP	316	[TCP
60	4.178322	172.16.80.238	172.16.78.29	TCP	66	48921
61	4.178332	172.16.78.29	172.16.80.238	HTTP/XML	73	HTTP/
62	4.178373	172.16.80.238	172.16.78.29	TCP	66	48921

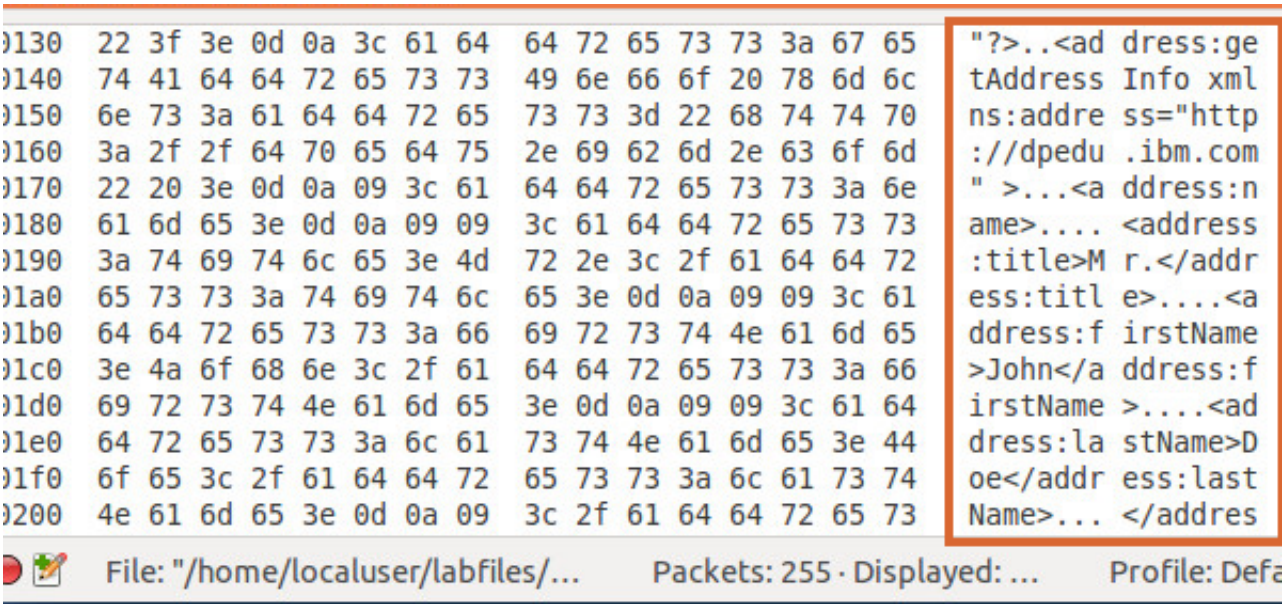
__ 2. Review the contents from test case 1.

__ a. In the top window in Wireshark, scroll down and click packet No 54.

No.	Time	Source	Destination	Protocol	Length	Info
51	4.176623	172.16.80.238	172.16.78.29	TCP	74	48921
52	4.176640	172.16.78.29	172.16.80.238	TCP	74	10955
53	4.176821	172.16.80.238	172.16.78.29	TCP	66	48921
54	4.177371	172.16.80.238	172.16.78.29	HTTP/XML	564	POST
55	4.177382	172.16.78.29	172.16.80.238	TCP	66	10955

Packet 54 contains the source IP address of the client and the destination IP address of the DataPower appliance. Therefore, you know that this packet is traveling to DataPower. There is also interesting information in the Info section; for example, Len = 564. This field informs you that the data payload is 564 bytes, or that there is data in this packet.

__ b. Scroll through the contents of the bottom window where you see the actual data payload.



__ c. Review the data payload content.

__ d. Select **No 59**. This line is the DataPower-to-client return message.

__ e. Notice that the payload length is 316.

__ f. Review the data payload content.

Section 4: Test case 2: Analysis of unsuccessful XML message

__ 1. Review the contents from test case 2.

__ a. In the top window in Wireshark, scroll down and click packet No 196.

__ b. Review the data payload content. This content is the POST request message.

- **Len**: The length parameter in the Info field can be the first indicator that there is a problem with this message, especially when a message with a length of 564 is expected. This message has a length of 565.
- **Data Payload Review**: From within the contents of the message payload section, you can see that the tag `<title>` has the beginning tag of `<title1>`. This section allows you to look at the data and see whether there is a problem with the data that comes in. Here, you are able to see the discrepancy.

__ c. **Select No 199**. This line is the DataPower-to-client return message.

__ d. Review the data payload content. This content is the return DataPower SOAP Fault error message.

__ e. Close Wireshark by clicking the **Close** icon.

End of exercise

Exercise review and wrap-up

In this exercise, you examined the system log to see the types of entries that are recorded for both successful and unsuccessful messages. You used the multi-step probe to provide a comprehensive view of each step of the processing policy that is started by the service as it handles a message. You used a network analysis tool to review packets that are captured from DataPower.

The probe presents the message contents, the value of system and processing variables, the execution path, and any error information available for each step of processing.

Do not leave the multi-step probe in the enabled state for production conditions because it slows down the system. Restarting the appliance automatically disables the probe.

Exercise 4. Securing connections with SSL

Estimated time

01:00

Overview

This exercise shows you how to create cryptographic keys by using the DataPower crypto tools. You create a crypto identification credential that stores certificate-key pairs that are used in securing SSL connections. You also create a validation credential object for validating certificates. These objects are used as part of a crypto profile. Finally, you modify a crypto profile to use the new key and certificate.

Objectives

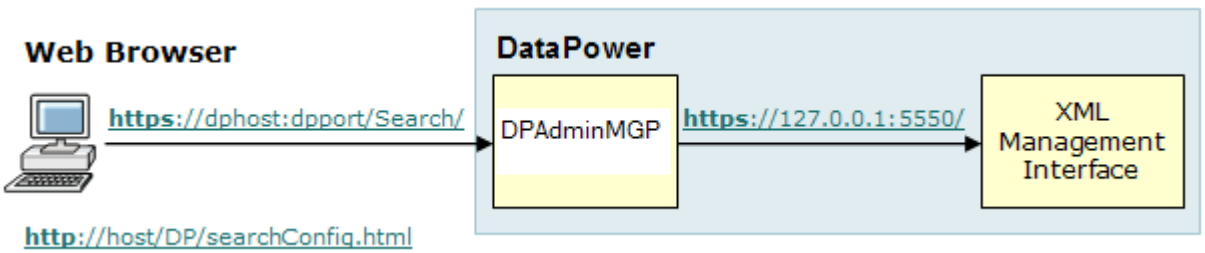
After completing this exercise, you should be able to:

- Use the DataPower cryptographic tools to generate cryptographic keys
- Use a cryptographic key and certificate object to create a cryptographic identification credential
- Use a validation credential object to validate certificates
- Create an SSL proxy profile to accept SSL connections from a client

Introduction

The DataPower appliance supports generating certificate-key pairs. These keys can be used to secure communication to and from the DataPower appliance.

In "Test the DPAdmin multi-protocol gateway service" on page 2-24, you provided a user name and password to the *DPAdmin* multi-protocol gateway service to search for configuration information about the appliance. This information is passed through the web browser by using HTTP basic authentication, which sends it unencrypted to the appliance. You should never send this information unsecured across a network. A solution to this problem is to use SSL. Using SSL, the entire link between the client and server is secured. In this exercise, you create an SSL server crypto profile object that secures the communication between the web browser and DataPower appliance.



Successful completion of the exercise includes the following tasks:

- Create the cryptographic objects: Crypto key, crypto certificate, crypto identification credential, and crypto profile
- Associate a server crypto profile with the *DPAdmin* multi-protocol gateway service
- Verify the generated SSL proxy profile
- Test the SSL connection

The crypto profile that is specified depends on whether the DataPower appliance acts as a client or server during SSL communication.

Requirements

To complete this exercise, the following resources are required:

- Access to the *DataPower* appliance
- Access to the *<lab_files>* directory

Information

In this exercise, you use the DataPower crypto tools to create key and certificate objects that are used during SSL communication.

At the start of this exercise, it is a good idea to check that the time on the student image matches with the time on the appliance. Significant time differences of more than a few minutes can cause problems with the use of SSL. These problems can occur if the client certificates are not yet valid (that is, if the validity date on the certificate is before the current date and time on the appliance).

If you incorrectly specify the key information, the key and certificate can be “redone.” Use file management to delete the key, certificate, and CSR from the `cert:` and `temp:` directories. You also must delete the related key object and certificate object from OBJECTS.

Exercise instructions

Preface

- The references in exercise instructions use the following values:
  - *<lab_files>*: `/usr/labfiles/dp`
  - *<image_ip>*: IP address of the image
  - *<dp_admin_login>*: DataPower secondary administrator user name
  - *<dp_admin_password>*: DataPower secondary administrator password
  - *<dp_internal_ip>*: IP address of the appliance’s management interfaces
  - *<dp_public_ip>*: IP address of the public services on the appliance

- <*dp_WebGUI_port*>: WebGUI port number of the DataPower appliance; the default port is 9090
- <*dp_xml_mgmt_port*>: The port number of the DataPower XML Management interface: 5550
- <student*nn*>: Student developer user account
- <student*nn*_password>: Password for the student account
- <student*nn*_domain>: Student application domain
- <*mpgw_dpadmin*>: The port for the DataPower administrative multi-protocol gateway (MPGW) service, port 10*nn*1

4.1. Generate a certificate-key pair on the DataPower appliance

In this section, you use the DataPower appliance to create a certificate-key pair. The certificate-key pair can be used during SSL connection. The generated certificate that contains your public key can be presented to clients.

__ 1. Log on to the DataPower Blueprint Console.

__ a. Open a web browser, and enter the URL:
`https://<dp_internal_ip>:<dp_WebGUI_port>/dp/login.xml`

__ b. Enter your user name (`student`*`nn`*), password, and select your domain (`student`*`nn`*`_domain`).

__ c. Click **Login**.

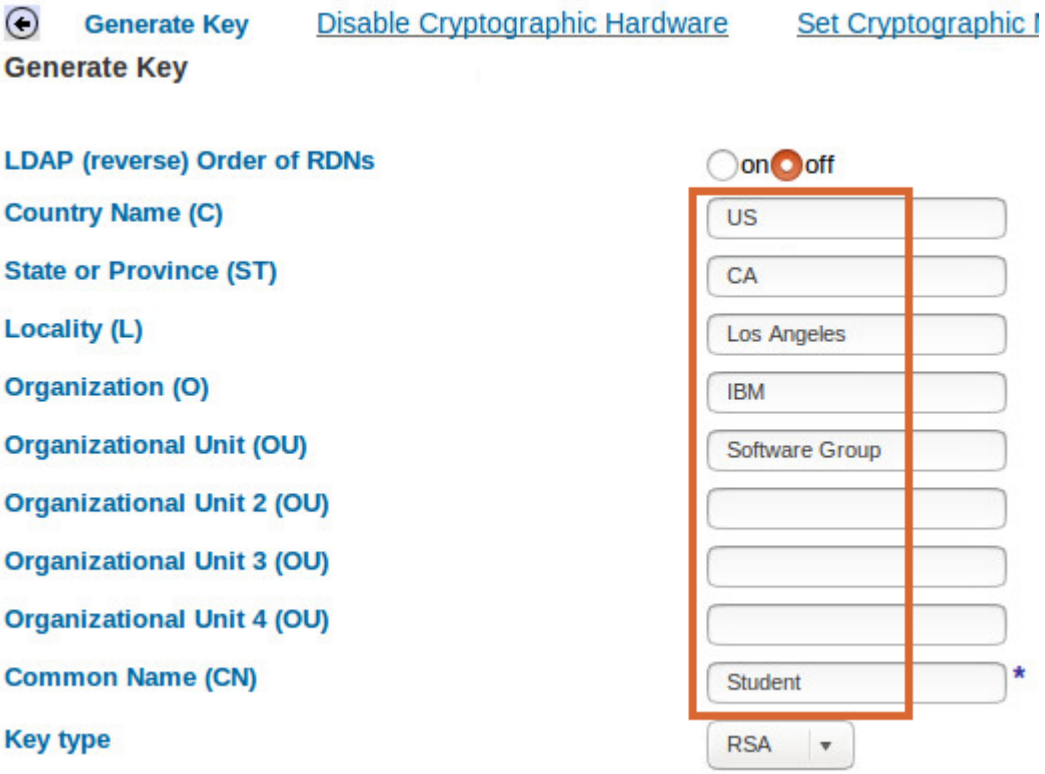
__ 2. Generate a certificate-key pair on the DataPower appliance.

__ a. In the DataPower Blueprint Console Search, type `crypto t` in the **Search** field.

__ b. Select **Crypto Tools** from the resulting list.

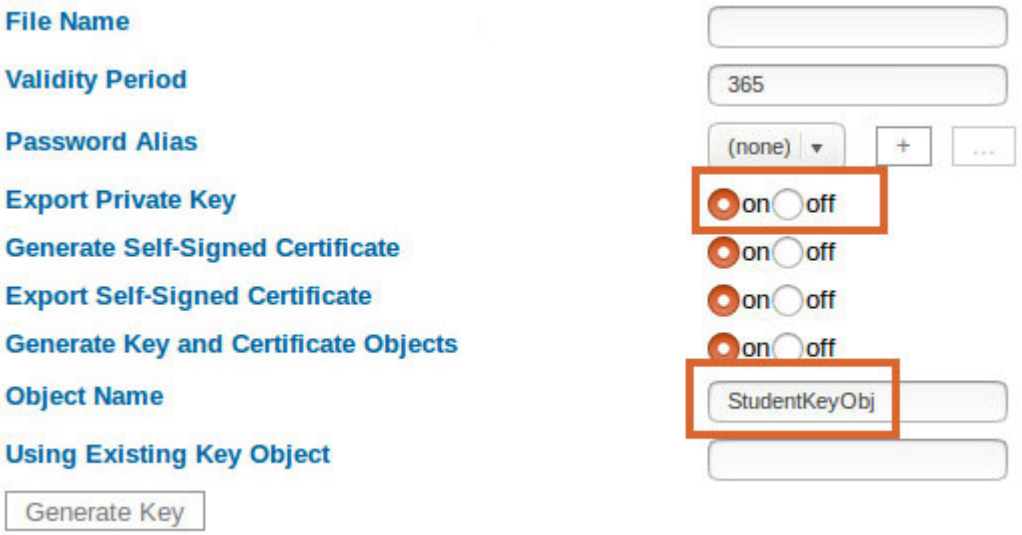
__ c. The “Generate Key” page generates a certificate-key pair by using the information that is typed on this page. The fields from **Country Name** down to **Common Name** are part of the distinguished name. Enter the following information for the distinguished name:

- **Country Name (C)**: `US`
- **State or Province (ST)**: `CA`
- **Locality (L)**: `Los Angeles`
- **Organization (O)**: `IBM`
- **Organizational Unit (OU)**: `Software Group`
- **Common Name (CN)**: `Student`



__ d. The remaining fields are for certificate-key pair information. Enter the following information and leave the remaining fields at their default values:

- **Export Private Key**: **on**
- **Object Name**: `StudentKeyObj`



__ e. Click **Generate Key**.

__ f. Click **Confirm** to proceed with generating the private key and self-signed certificate.

__ g. Verify that you see the following message in the dialog box that is displayed:

```
Action completed successfully.
```

This action generates a private key and self-signed certificate and places them in the DataPower appliance `temporary:///` directory. Two objects, each with the name `StudentKeyObj`, are created for both the private key and self-signed certificate.

In addition to generating the private key and self-signed certificate, a certificate signing request (CSR) is also generated. A CSR is a request message that is sent to a certificate authority (CA) to create a digital certificate. A CSR consists of identification information (for example, common name) and your public key. The CSR request is signed with your private key, but the actual private key is not included in the request. The CA issues a signed digital certificate that replaces your self-signed certificate.

Note

If you do not select **on** for **Export Private Key** or **Export Self-Signed Certificate**, then you are unable to download the keys. The keys are generated and placed in the `cert:///` directory only.

__ 3. Verify the generation of the private key and certificate objects that is called `StudentKeyObj`.

__ a. Click the **Search** link at the top of the Blueprint Console.

__ b. Type `Crypto` in the Search field. Then, select **Crypto Key**

__ c. Verify that you see the **StudentKeyObj** referencing the generated private key file.

__ d. In the Object list, click **Crypto Certificate**.

__ e. Verify that you see the **StudentKeyObj** referencing the generated self-signed certificate file.

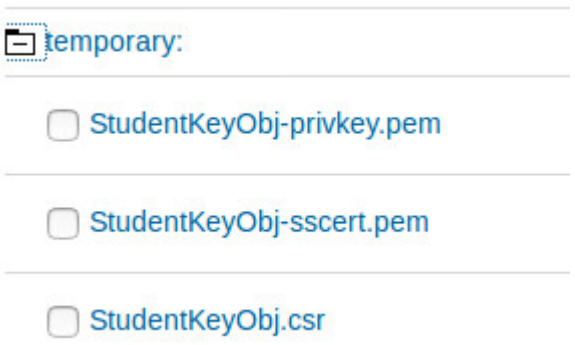
Information

The two objects that the DataPower appliance generates, and that have the same base name, are separate objects: one being the certificate and the second being the private key. Both objects are in the `cert:` directory.

__ 4. View the private key and self-signed certificate that are exported to the `temporary:` directory.

__ a. In the Search field, type `file`. Then, select **File Management**.

__ b. Expand the `temporary:` directory. Notice the exported private key, the self-signed certificate, and the CSR:



The `StudentKeyObj-privkey.pem` is the private key file.

The `StudentKeyObj-sscert.pem` is the self-signed certificate file.

The `StudentKeyObj.csr` is the certificate signing request (CSR) file.

4.2. Create crypto objects

__ 1. Create an identification credential object.

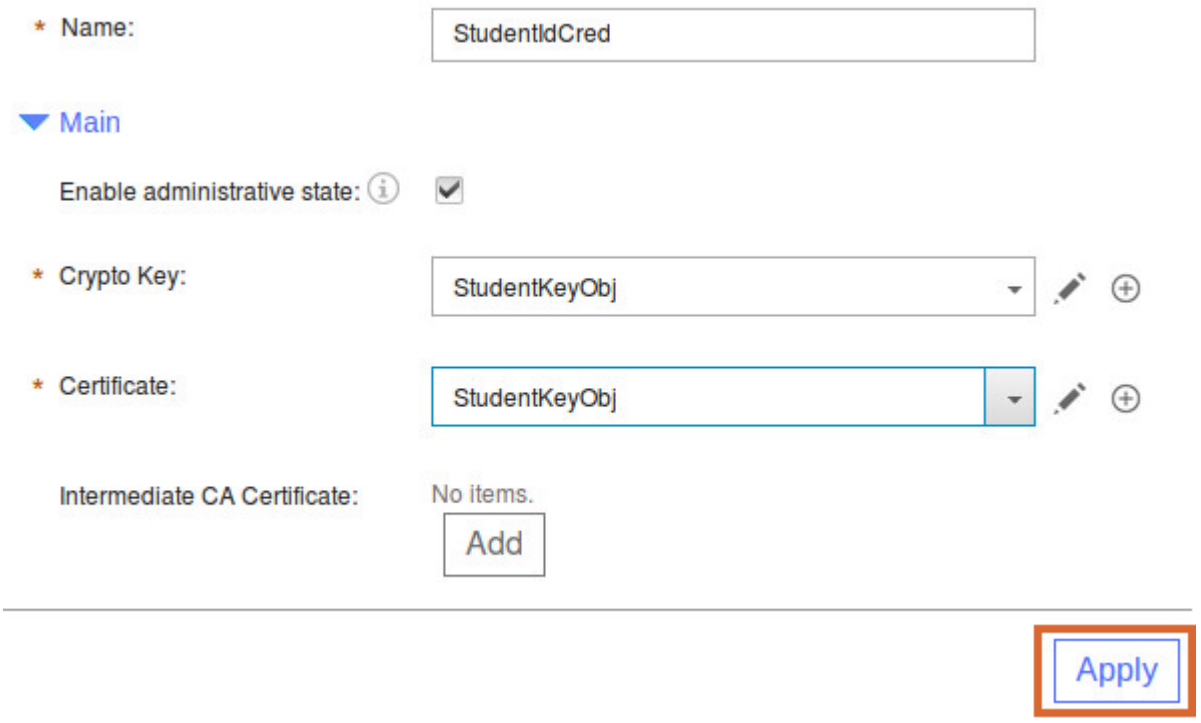
An identification credential object is used to reference a certificate-key pair during an SSL connection. You create an identification credential that references the certificate-key objects that are created in the previous steps. The identification credential object is used to identify yourself during an SSL connection, and to participate in the SSL handshake.

__ a. In the Main menu **Search** field, type `identification`. Then, click **Crypto Identification Credentials**.

__ b. On the Configure Crypto Identification Credentials page, click **New**.

__ c. On the next page, type the following information:

- **Name**: `StudentIdCred`
- **Crypto Key**: `StudentKeyObj`
- **Certificate**: `StudentKeyObj`



__ d. Click **Apply**.

This action creates an identification credential object. When a third-party CA signs the certificate, you can specify CA certificates in the **Intermediate CA Certificate** field.

__ 2. Create a validation credential object.

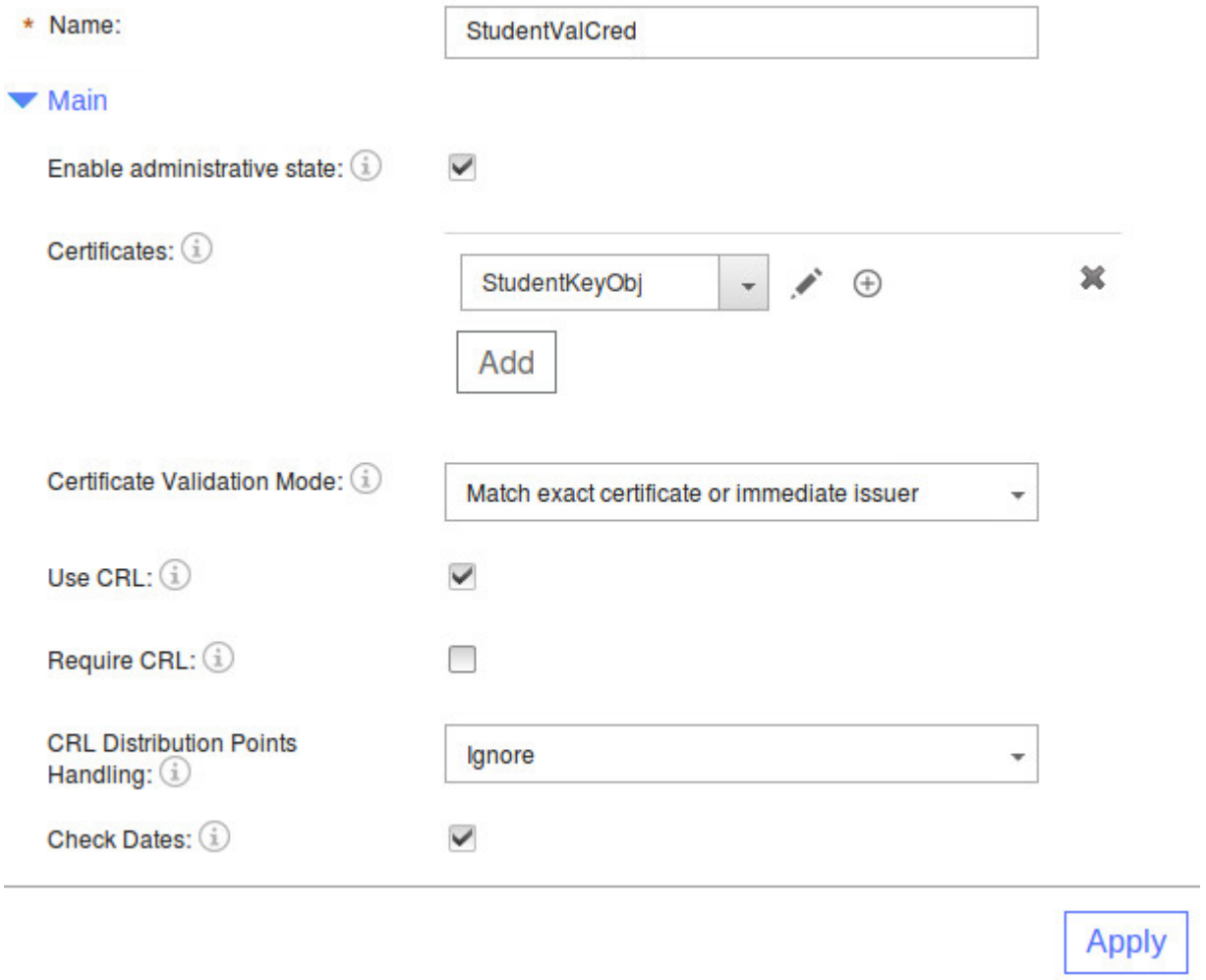
A validation credential object is used to validate the authenticity of certificates and digital signatures that is presented to a service.

__ a. In the left column in the Blueprint Console, click the **Crypto Validation Credentials** link.

__ b. On the Configure Crypto Validation Credentials page, click **New**.

__ c. Enter a credentials name of `StudentValCred`.

__ d. Alongside Certificates, click **Add**. Then, select the **StudentKeyObj** certificate from the Certificates drop-down list.



__ e. Click **Apply**. This validation credential validates the **StudentKeyObj** certificate if it is presented.

__ 3. Create a server crypto profile to use in an SSL communication.

A crypto profile is used to identify certificate-key pairs in an SSL connection. It can also validate presented certificates by using a validation credential object.

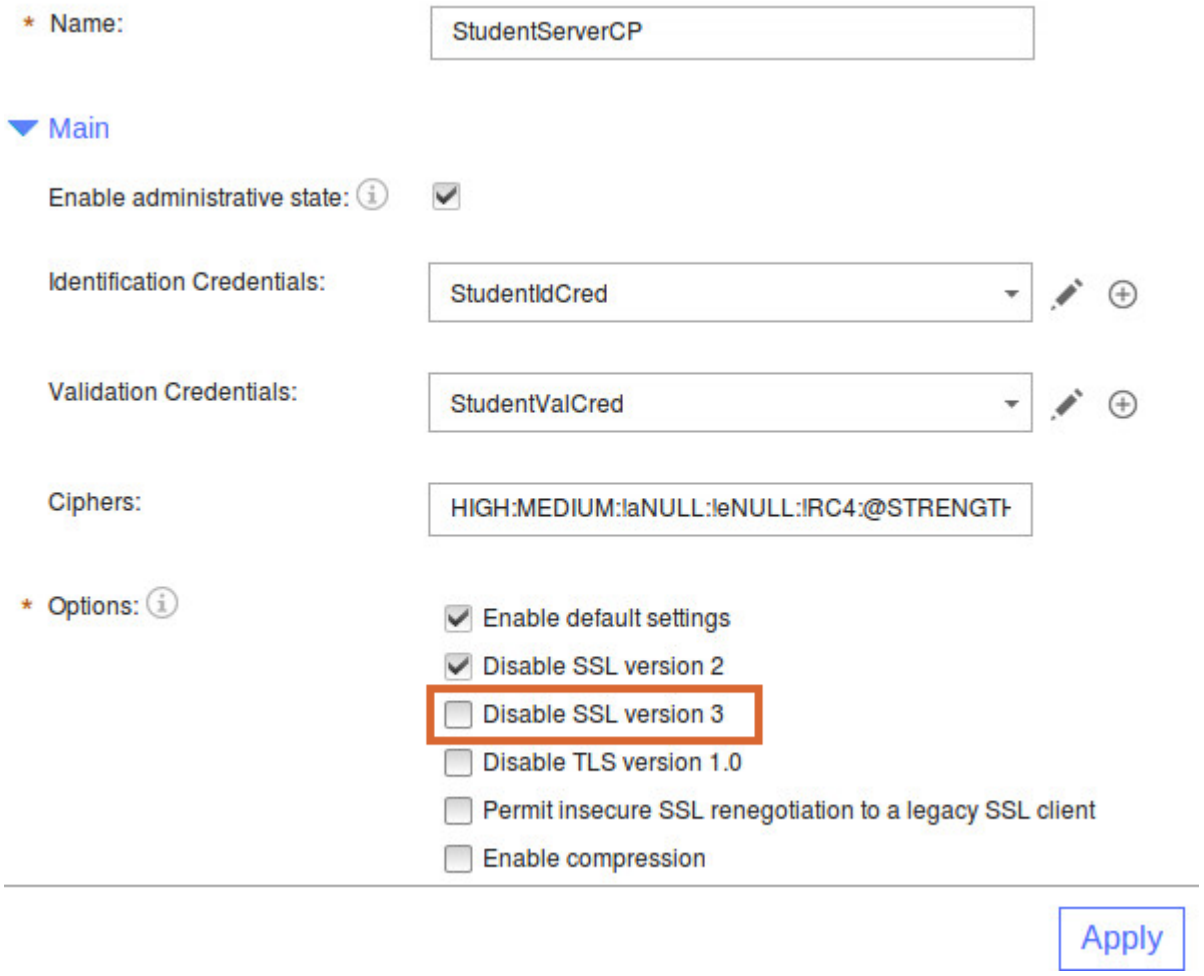
__ a. In the left column in the Blueprint Console, click the **Crypto Profile** link.

__ b. On the Configure Crypto Profile page, click **New** to create a crypto profile.

__ c. Type the following information (leave the remaining fields at their default values):

- **Name**: `StudentServerCP`
- **Identification Credentials**: `StudentIdCred`
- **Validation Credentials**: `StudentValCred`

- **Disable SSL version 3**: If selected, clear this check box



__ d. Click **Apply**.

This crypto profile specifies an identification credential object with a certificate-key pair. If the DataPower appliance requests a client certificate during SSL authentication, then you also need to specify a validation credential object that validates the client certificate.

Note

In this exercise, you use only the **StudentServerCP** crypto profile. You do not need to create a client crypto profile.

4.3. Configure server-side SSL

In this section, you modify the original crypto profile to use the new identification credential that you created in the previous section.

__ 1. Change the crypto profile DPAdminSSLProfile to use the new identification credential. This profile is the one used in the DPAdmin multi-protocol gateway.

__ a. From the Crypto Profile catalog list, select **DPAdminSSLProfile**.

__ b. On the Configure Crypto Profile page, change the **Identification Credentials** field to: `StudentIdCred`

This credential is the one that uses the StudentKeyObj key and certificate. If you want, you can click the **Edit** (...) icon to examine the key and certificate objects that are specified in the identification credential.

__ c. Leave the Validation Credentials field blank.

__ d. Click **Apply**.

__ 2. Examine the DPAdmin multi-protocol gateway configuration page.

__ a. In the Main menu **Search** field type `multi`, then click **Multi-Protocol Gateway**.

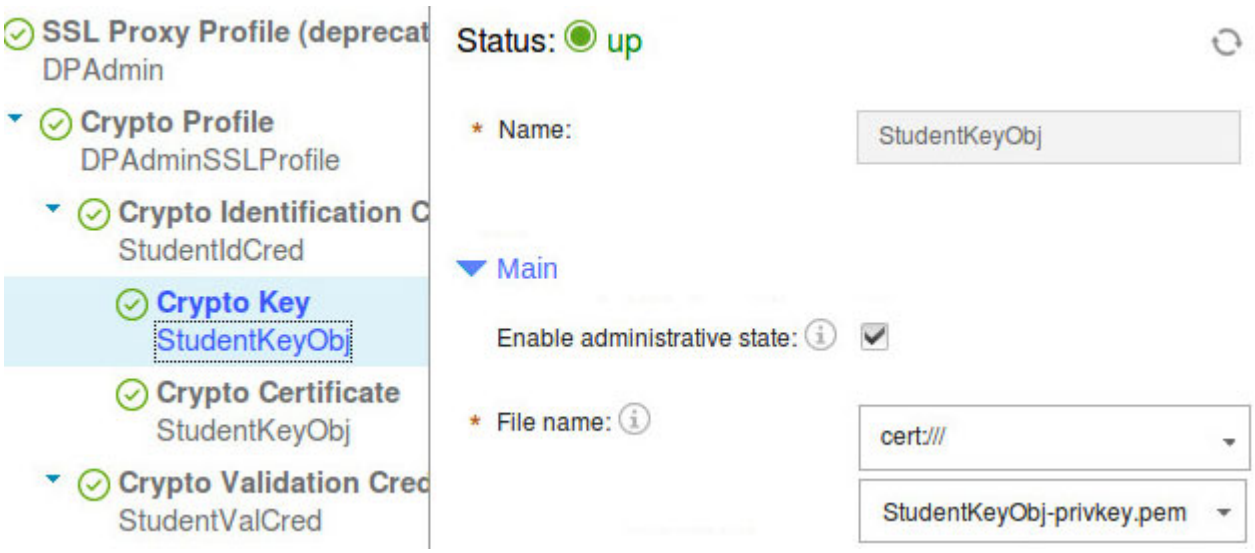
__ b. From the **Multi-Protocol Gateway** list, select **DPAdmin**. This action opens the Configure Multi-Protocol Gateway page.

__ c. Verify that the **Multi-Protocol Gateway Policy** field is populated with **DPAdminAppPolicy**.

__ d. From the **User Agent settings** section, find the **DPAdmin** proxy profile.

__ e. Click the **Edit** icon to examine the profile.

__ f. Click **Crypto Key** in the left pane.



The page shows the files that are referenced. Note the references to your `StudentKeyObj-privkey` and `StudentKeyObj-sscert` files. These files are the ones that were created in the previous section.

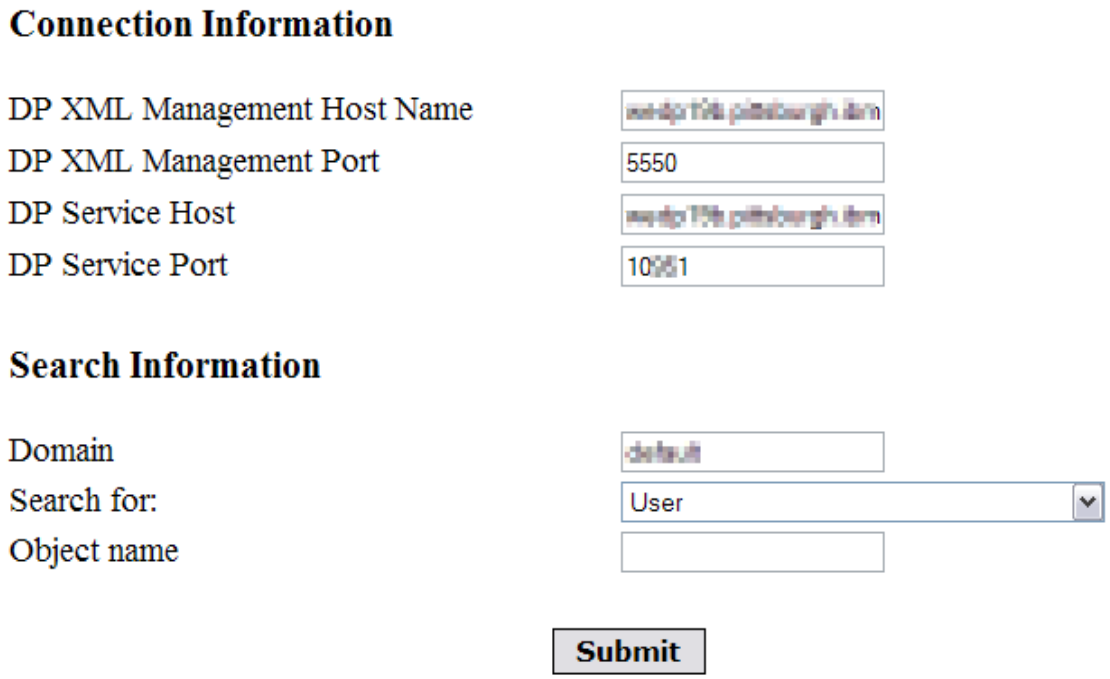
__ g. Click **Cancel** to close this page.

__ 3. Test the server-side SSL for the DPAdmin service.

__ a. Open a web browser and enter the URL: `http://<image_ip>/dp/searchConfig.html`

__ b. In the web page that is displayed, enter the following information:

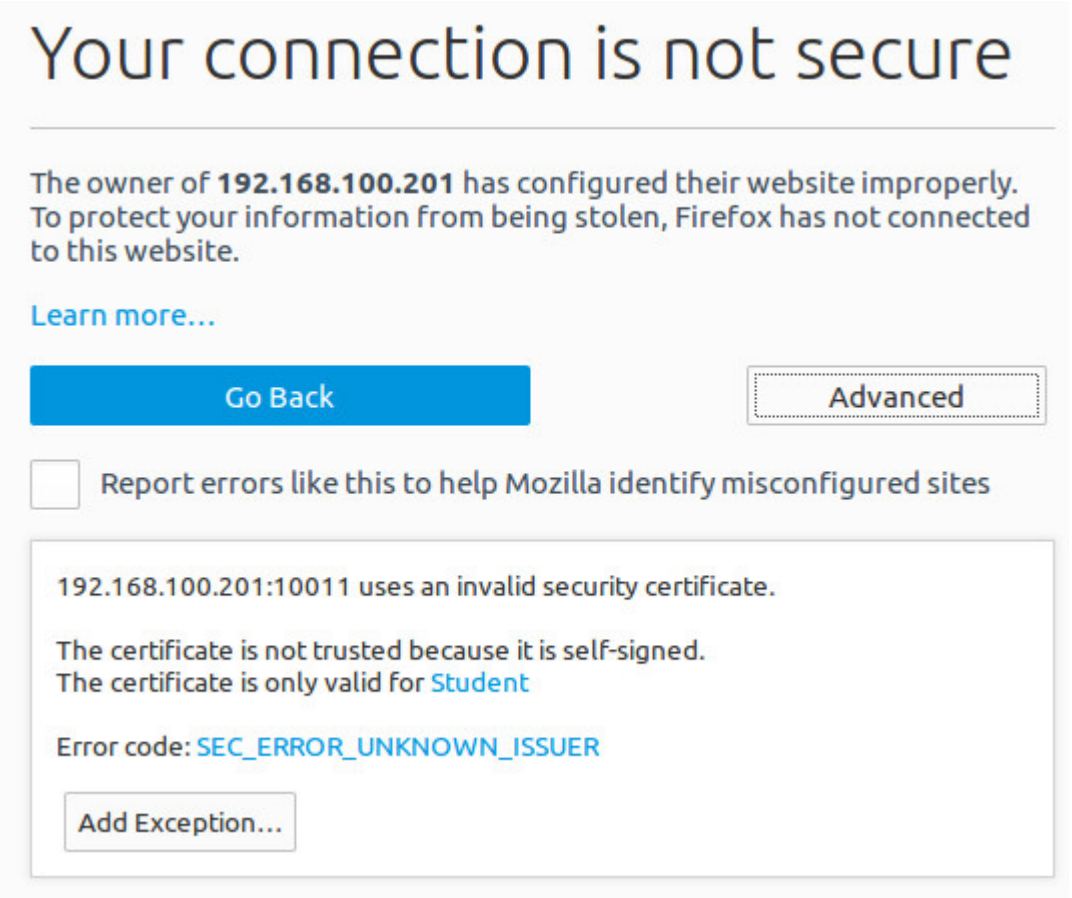
- **DP XML Management IP Address**: *<dp_internal_ip>*
- **DP XML Management Port**: *<dp_xml_mgmt_port>*
- **DP Service Host**: *<dp_internal_ip>*
- **DP Service Port**: *<mpgw_dpadmin>*
- **Domain**: `default`
- **Search for**: `User`



__ c. Click **Submit**.

__ d. If you are running the Firefox browser, the page displays that your connection is not secure. Click **Advanced**.

__ e. A dialog box is displayed that indicates you are being presented with a certificate that is only valid for “Student.” This name is the one you supplied when you created the key and certificate. Recall that on a previous occasion this certificate was valid for “Alice”.



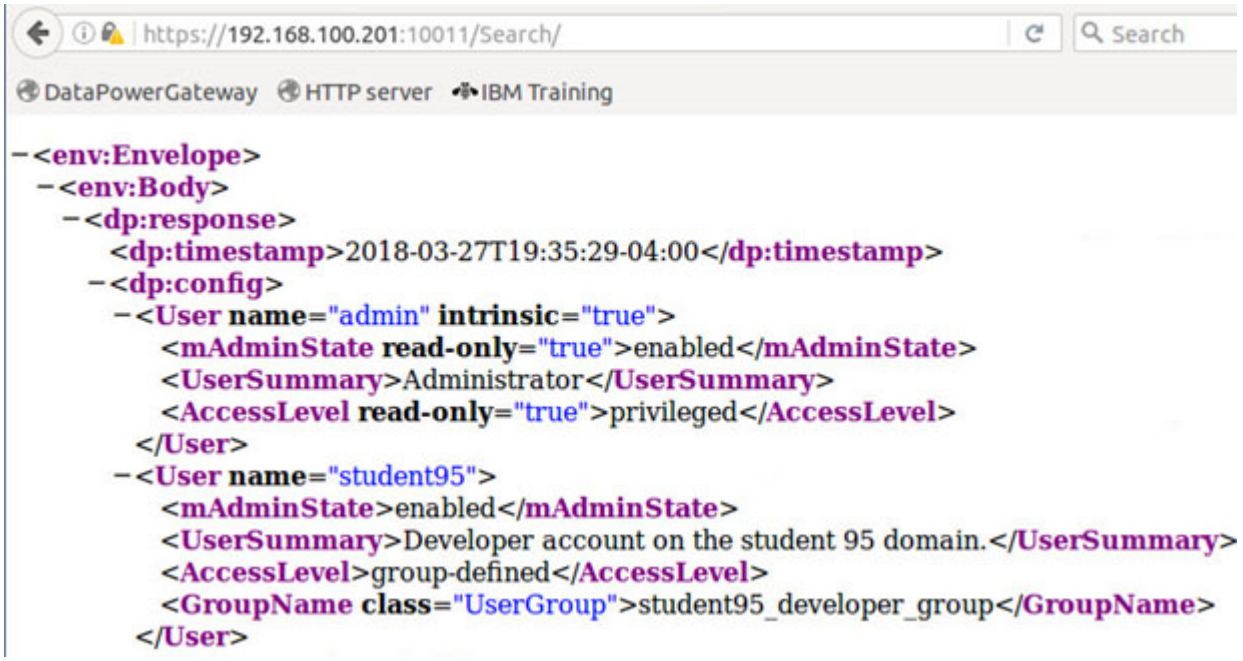
Click **Add Exception**.

__ f. Depending on the browser, you can examine the presented certificate. Notice the certificate values that reflect what you entered at creation time.

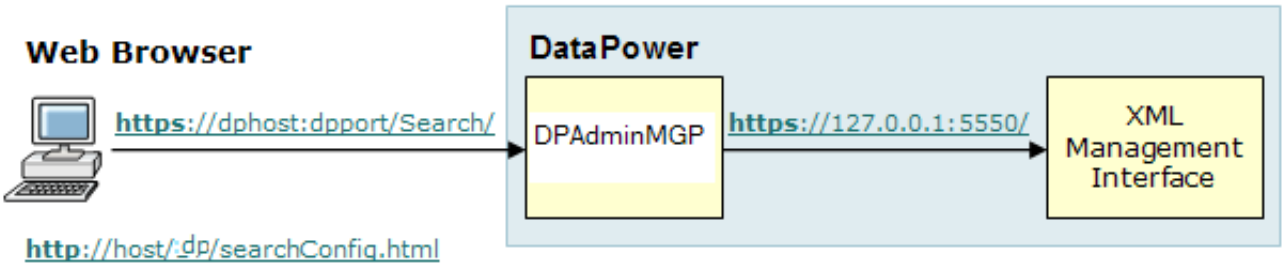
__ g. Click **Confirm Security Exception** to accept the certificate.

__ h. If requested, enter the user name *<dp_admin_login>* and the password *<dp_admin_password>*.

__ i. The correct response is the result of the query.



The following diagram explains the message flow:



__ j. Switch back to the browser tab with DataPower.

__ k. Click **Save changes**.

__ l. Log out of DataPower.

__ m. Completely close all instances of the browser.

Attention

The DPAdmin multi-protocol gateway service that is provided in this class is for demonstration purposes only. In production, you do not share the XML management interface to external clients without the appropriate security.

End of exercise

Exercise review and wrap-up

In this exercise, you generated a key and certificate object on the DataPower appliance. The key and certificate objects are used to create an identification credential object, which the crypto profile references. A crypto profile is used during SSL communication. Finally, the crypto profile is associated with the DPAdmin multi-protocol gateway service. During testing, you validated that the communication between the web browser, and DataPower appliance used SSL.

Exercise 5. Logging to an external system

Estimated time

01:00

Overview

This exercise shows you how to capture log messages and move them off the DataPower appliance. The DataPower appliance has limited memory capacity, and the on-box system logs can quickly become full. As a logging good practice, log messages that are generated on the appliance should be moved off the appliance. Most enterprises already have a logging system such as syslog, and the DataPower appliance supports many mechanisms for integrating with these systems.

Objectives

After completing this exercise, you should be able to:

- Use the Generate Log Event action to test the log target configuration
- Create a log target that subscribes to specific log categories
- Create a log target that sends log messages to an external logging system

Requirements

To complete this exercise, you need:

- Access to the *DataPower* appliance
- *cURL*, to send requests to the DataPower appliance
- Access to the `<lab_files>` directory for some needed exercise files
- *rsyslog* that is configured on the student image `<image_ip>`

The first few steps in the exercise you become familiar with the system log, system log view, log levels, and how to generate log events. It is fairly straightforward.

Troubleshooting

For the external logging, you import a LogTransformMPG service. Be sure to change the HTTP front-side handler *port number* on the service, and customize the log action destination. The same message is passed, by using the same cURL command (with a different port).

The “external logging system” is the DP_Loggers Java application, which was developed in-house to IBM to test external logging. It is a Java application that acts like an HTTP server that receives

log messages. Be aware of the port number on which it listens; the log action must match it. If the logger does not start, check that the configured port is not already in use on that image (`netstat -a`). If the students want to minimize the detail in the logger window, they can edit the `.sh` file and delete the `verbose` parameter.

To complete the syslog-tcp section, you need to be on a system that provides a rsyslog facility. The class image contains rsyslog by default. If you have problems with getting the log messages to Linux, verify that the `rsyslog.conf` file is edited to support the labs.

Exercise instructions

Preface

- Remember to use the domain and port addresses that you are assigned in the exercise setup. *Do not* use the default domain unless directed.
- The references in the exercise instructions are:
  - <*lab_files*>: **/usr/labfiles/dp**
  - <image_ip>: IP address of the image
  - <*dp_internal_ip*>: IP address of the appliance's management interfaces
  - <*dp_public_ip*>: IP address of the public services on the appliance
  - <*dp_WebGUI_port*>: WebGUI port number of the DataPower appliance; the default port is 9090
  - <*logger_app_port*>: Port on which external logger is listening, port 1112
  - <student*nn*>: Student developer user account
  - <student*nn*_domain>: Student application domain
  - <*mpgw_log_port*>: Port on which the LogTransformMPG service listens, port 10*nn*5

5.1. Examine and test the system log

The default system log is a critical tool to monitor or troubleshoot the appliance. You open the system log to manipulate the various filters. You generate a log event to observe the result.

__ 1. Log on to the `student`*nn*`_domain` of the DataPower WebGUI by using a web browser and your `student`*nn* account, `https://`*<dp_internal_ip>*`:`*<dp_WebGUI_port>*`/dp/login.xml`

__ a. Enter your user name (`student`*nn*), password, and select your domain (`student`*nn*`_domain`).

__ 2. Examine the system log target.

__ a. From the Main menu, click **Administration**. Then, select **Miscellaneous > Manage Log Targets** from the navigation bar.

__ b. The page refreshes and displays the catalog (list) of logs. A log is displayed; click **default-log**.

__ c. The Manage Log Target page is displayed. Although this entry is for the system log, notice that it is the same as any log target. However, in this log target, the fields are disabled, so you cannot change any of the characteristics.

__ d. Close the dialog box.

__ 3. Examine the system log entries.

__ a. In the **Search** field of the Blueprint Console, type `logs`. Then, click **System Logs**.

This method is one way to view the system log. You see entries from your earlier activities. One issue with using this path to open the system log is that you must leave it to complete other configuration tasks. Another way to open the system log is covered shortly.

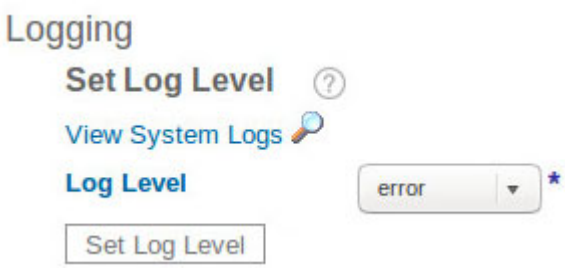
__ 4. Examine the Troubleshooting pane.

__ a. From the Main menu, click **Administration**. Then, click **Debug > Troubleshooting** from the navigation bar.

__ b. The second row on this page is the **Logging** section. The **Set Log Level** value controls the minimum severity of messages that are captured in the log. Set the Log Level to error.

__ c. From the **Log Level** list, select **error**.

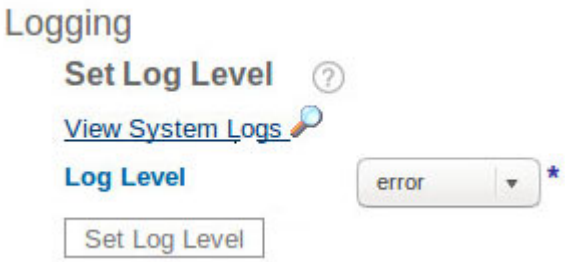
__ d. Click **Set Log Level**.



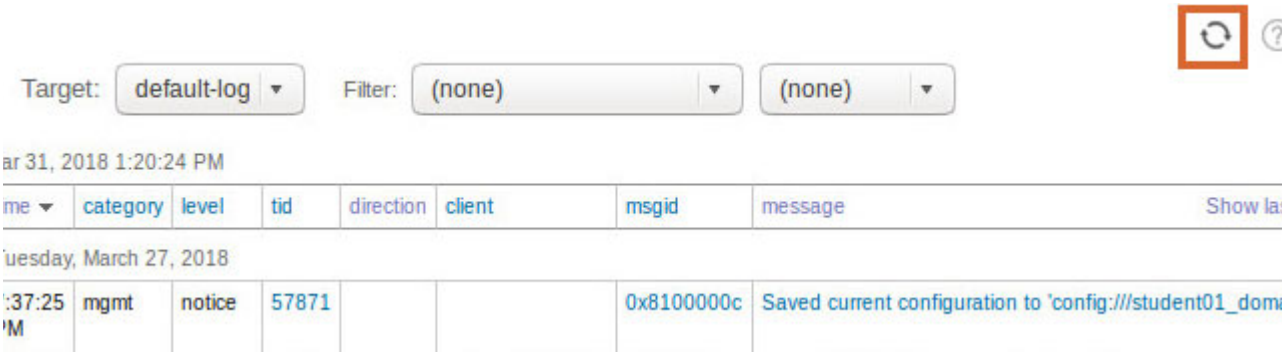
__ e. Confirm the change and close that window.

__ 5. Open the system logs.

__ a. Click the **View System Logs** link.



__ b. Another window opens, containing the system log. Using this path to the system log keeps this window open while changes are made back in the main Web Management page. Whenever you need to see the log again, switch back to the system log window and click the **Refresh Log** icon.



Information

If you click a **tid** (transaction ID), another log window opens that contains entries only for that specific transaction.

If you select **msgid**, you get another log window that contains only entries with that message number.

Selecting the actual *message* opens another window with a general description of the error.

Although you set the log level to **error**, the system log sets a log level for **mgmt** category messages to **notice**.

Also, note the **Filters** at the top of the page. One list filters on a single log category (log type), and the other filters on minimum severity.

If you look at the system log (in the default domain), you see another column that identifies from which domain the message came, and you also get a filter by domain.

__ c. Try filtering your log by different criteria to see the effect. If you do not have any “interesting” entries, run a few of the cURL test messages from previous labs to generate more messages.

__ d. Leave the log window open.

__ 6. Generate an event by using the **Generate Log Event** part of the Troubleshooting page.

__ a. Make sure that you are on the Troubleshooting page.

__ b. Under **Generate Log Event**, enter:

- Log Type (log category): `aaa`
- Log Level: `notice`
- Log Message such as `Test MSG - aaa @ notice`
- Event code: <*Click **Select Code** and choose any event code*>

__ c. Click **Generate Log Event**.

__ d. Confirm that the message was sent and close that window.

__ 7. Check the system logs for the log entries.

__ a. Switch back to the log window, and click **Refresh Log**.

__ b. You do not see your test message. Why not?
The reason is that the appliance Log Level is **error**, and the log level of the message is at a lower level.

__ c. Generate a test message again, but change the “Generate Log Event” Log Level to **error**.

__ d. Back on your log window, click **Refresh Log**.

__ e. Your message is shown.

__ f. Keep the log window open.

5.2. Create your own log category and log target

You can create your own “application” log category that allows you to specify a unique category in an `xsl:message` element, and send them to a log target defined for that category only. One **caution**: new categories must be unique across the domain, and across the entire appliance.

With a “custom” log category, you can restrict a log target for that specific category only.

__ 1. Create a log category.

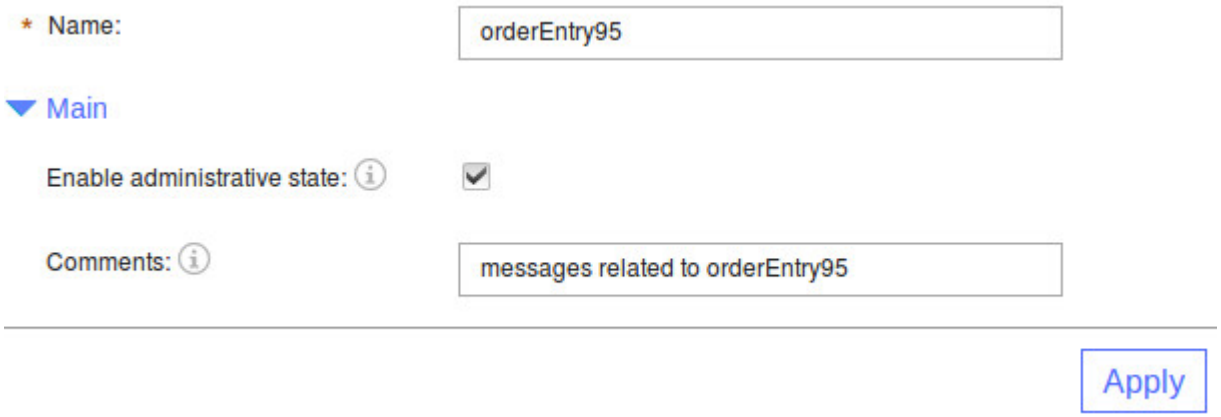
__ a. Make sure that you are in your domain `student`*`nn`*`_domain` by using the `student`*`nn`* user.

__ b. From the Main menu, click **Administration**. Then, click **Miscellaneous > Configure Log Categories**.

__ c. The catalog of predefined log categories is shown. Click **New**.

__ d. Enter the following information and click **Apply**. The sample entry for student 95 is shown here:

- **Name**: `orderEntry`*`nn`*
- **Comments**: `messages related to orderEntry`*`nn`*



__ 2. Generate a log event by using the new category.

__ a. From the Main menu, click **Administration**. Then, select **Debug > Troubleshooting** from the navigation bar.

__ b. Under **Generate Log Event**, enter:

- **Log Category**: `orderEntry`*`nn`*
- **Log Level**: `error`
- **Log Message**: `TEST MSG - orderEntrynn @ error`

__ c. Click **Generate Log Event**.

__ d. Switch back to the log window, and refresh it. You see your custom category message.

__ 3. Create a log target for the custom log category `orderEntry`*`nn`*.

__ a. To create a log target specifically for this category, from the Main menu, click **Administration**. Then, select **Miscellaneous > Manage Log Targets**.

__ b. Click **New**.

__ c. Use the **Target Type** list to select different log target types. The entry fields change depending on the type.

__ d. On the **Main** tab, enter:

- **Name**: `orderEntry`*`nn`*`LocalLog`
- **TargetType**: `File`
- **Log Format**: `XML`
- **File Name**: `logtemp:///orderEntry`*`nn`*`.log`
- **Number of Rotations**: `1`
- Leave the other fields at their default values.

__ e. Examine the **Event Filters** tab. You can specify event codes to include or suppress. No entries are needed here for now.

__ f. Examine the **Object Filters** tab. You can specify objects to include in the log.

__ g. Examine the **IP Address Filters** tab. You can specify that log messages from only specific IP addresses are written to this log target.

__ h. Examine the **Event Triggers** tab. You can specify CLI commands that are run when a message with a specific message ID or event code is received. This function is tested later.

__ i. Click the **Event Subscriptions** tab and click **Add**.

__ j. Select the **Event Category** of `orderEntry`*`nn`* and **Minimum Event Priority** of `notice`.

__ k. Click **Apply** on the Log Target page.

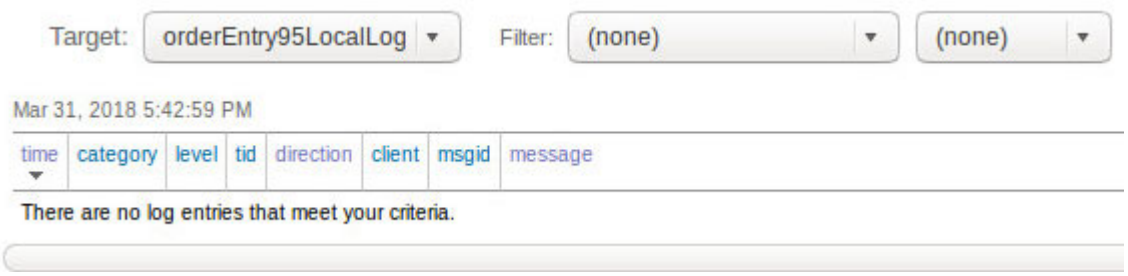
__ l. The new log is shown in the catalog (go back to the **Main menu > Administration > Miscellaneous > Manage Log Targets** to see it).

Name	Status	Op-State	Administrative state	Target Type	Log Format
default-log	saved	up	enabled	file	text
orderEntry95LocalLog	new	up	enabled	file	xml

__ 4. Open the system log in the opened window to verify the addition of the new log target.

__ a. Switch back to your log window, and refresh it.

__ b. Select the **Target** list. You now see your new log as a choice. For now, it is empty.



__ 5. Generate a log event by using the new category.

__ a. From the Main menu, click **Administration**. Then, select **Debug > Troubleshooting** from the navigation bar.

__ b. Under **Generate Log Event**, enter:

- **Log Category**: `orderEntry`*`nn`*
- **Log Level**: `error`
- **Log Message**: `TEST MSG - orderEntry`*`nn`*` @ error`

__ c. Click **Generate Log Event**.

__ d. Confirm the action and close the window.

__ 6. Find the log message in the system logs.

__ a. Switch back to your log window. Set the target as **default-log**, and refresh.

__ b. You see the new message.

__ c. Now change the Target to: `orderEntry`*`nn`*`LocalLog`

__ d. The log refreshes and only the `OrderEntry`*`nn`* category log message is shown in this custom log target.

Note

Remember, for temporary filtering of messages, you can also use the filters on the system log page.

5.3. Use an event trigger

In this section, you add an event trigger to the orderEntry*nn* log. When a specific message is detected (orderError), several CLI commands are run. The CLI commands make a directory in the local:///, and copy the current system log to it. This situation represents a case where the customer wants to save a copy of the log when a specific message is detected.

__ 1. Open the orderEntry*nn*LocalLog log target configuration again. You access the option from the **Main** menu **> Administration** > **Miscellaneous > Manage Log Targets**.

__ 2. Click the **Event Triggers** tab.

__ 3. Click **Add** to create a specific event trigger.

__ 4. For **Message ID**, enter `0x88776655`**.**

__ 5. Set the Regular Expression to **orderError**.

__ 6. The **CLI Command** field contains two commands that are separated by a semicolon:

mkdir local:///capturedLog; copy logtemp:///default-log local:///capturedLog/retained-default-log

This command string creates a subdirectory in the `local: directory`, and copies the current system log to that directory with a different name.

__ 7. Click **Apply** to save the event trigger and the edited log target.

__ 8. Send a test message that triggers the event.

__ a. Go to the Troubleshooting page.

__ b. Under **Generate Log Event**, enter:

- **Log Category**: `orderEntry`*`nn`*
- **Log Level**: `error`
- **Log Message**: `orderError - name is missing`
- **Event Code**: `0x88776655`

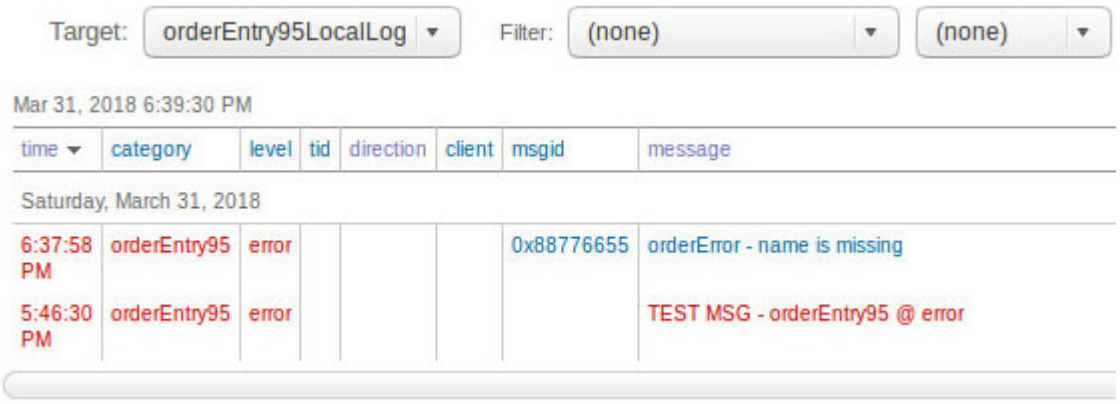
__ c. Click **Generate Log Event**.

__ d. Confirm the action and close the window.

__ 9. Find the log message in the system logs.

__ a. Switch back to your log window. Set the target as **orderEntry*nn*LocalLog**, and refresh.

__ b. You see the new message.



__ 10. Verify that the CLI commands ran.

__ a. Switch to the File Management page in the Blueprint Console. From the Main menu, click **Administration**. Then, select **Main > File Management**.

__ b. Expand the **local:** directory. You should see the **capturedLog** subdirectory that the first CLI command in the event trigger created.

__ c. Expand the **capturedLog** directory. You should see the file **retained-default-log**. The second CLI command in the event trigger caused the copy activity.

__ d. When you edit the retained-default-log, the generated error message is displayed at the bottom of the log file.

5.4. Create a syslog-tcp log target

A common logging facility in the Linux is “syslog-tcp”. If you are running on such a system for the lab exercises, such as the supplied class lab image, you have syslog-tcp available to you. In this section, you define a log target that uses syslog-tcp, and send a test message to it from the Troubleshooting page in the Web graphical management console.

__ 1. Create a log target that uses syslog-tcp.

__ a. To create another log target, from the main menu, click **Administration**, then select **Miscellaneous > Manage Log Targets**.

__ b. At the top of the catalog list, click **New**.

__ c. Use the **Target Type** list to select different log target types. The entry fields change depending on the type.

__ d. On the **Main** tab, enter:

- General Configuration
  - **Name**: `mySyslog-tcpTarget`
  - **TargetType**: `syslog-tcp`
  - **syslog Facility**: `local5`
  - **Rate Limit**: `10`
- Source Configuration
  - **Local IP Address**: `0.0.0.0`
  - **Local Identifier**: `Syslog-student`*`nn`*
- Destination Configuration
  - **Remote Host**: `<image_ip>`
  - **Remote Port**: `514`
  - Leave the other fields at their default values.

__ e. Click the **Event Filters** tab. You can specify event codes to include or suppress. For this exercise, you select a power failure event to include.

__ f. Click **Select Code** in the event subscription filter.

__ g. In the dialog box that is displayed, scroll to the “environmental” category. On the entry for “Power supply failure”, click **select**.

__ h. The dialog box closes. The event code “0x02220001” is shown in the Event Subscription Filter entry field. Click **Add**.

__ i. The power failure event code is shown in the **Subscription** list.

__ j. Click the **Event Subscriptions** tab and click **Add**.

__ k. Select the **Event Category** of **all**, a **Minimum Event Priority** of **notice**, and click **Apply**.

__ l. Click **Apply** on the Configure Log Target page.

__ m. The new log is shown in the catalog (click **Log Target** from Configure Log Target on the page to see it).

__ n. Click **Save changes**.

__ 2. In the previous section, you opened the system log to verify the addition of the new log target as a choice. Because the syslog-tcp log target is off the appliance, this new log target does not show. You can examine the system log to confirm this situation if you choose.

__ 3. Generate a log event that writes to the syslog-tcp log target.

__ a. From the Main menu, click **Troubleshooting**.

__ b. Under **Generate Log Event**, enter:

- **Log Category**: `all`
- **Log Level**: `critical`
- **Log Message**: `TEST MSG - power supply failure @ critical`
- **Event Code**: `0x02220001`

__ c. Click **Generate Log Event**.

__ d. Confirm the action and close the window.

__ 4. Find the log message in the system logs.

__ a. Switch back to your log window. Set the target as **default-log**, and refresh.

__ b. You see the new message.

__ 5. Find the log message in the syslog-tcp log target.

__ a. Open a terminal. Change to the directory where the syslog-tcp log writes these messages:

```
cd /var/log
```

__ b. The name of the file that receives the log messages is `syslog`. Rather than edit the whole text file, you can view the last few entries by using the `tail` command:

```
tail syslog
```

__ c. The last line should be like:

```
Apr 3 12:35:29 Syslog-student01 [0x02220001][all][critic] TEST MSG -
power failure @ critical
```

You see the local identifier for the file, along with the other typical DataPower logging fields.

5.5. Import and modify the LogTransformMPG service

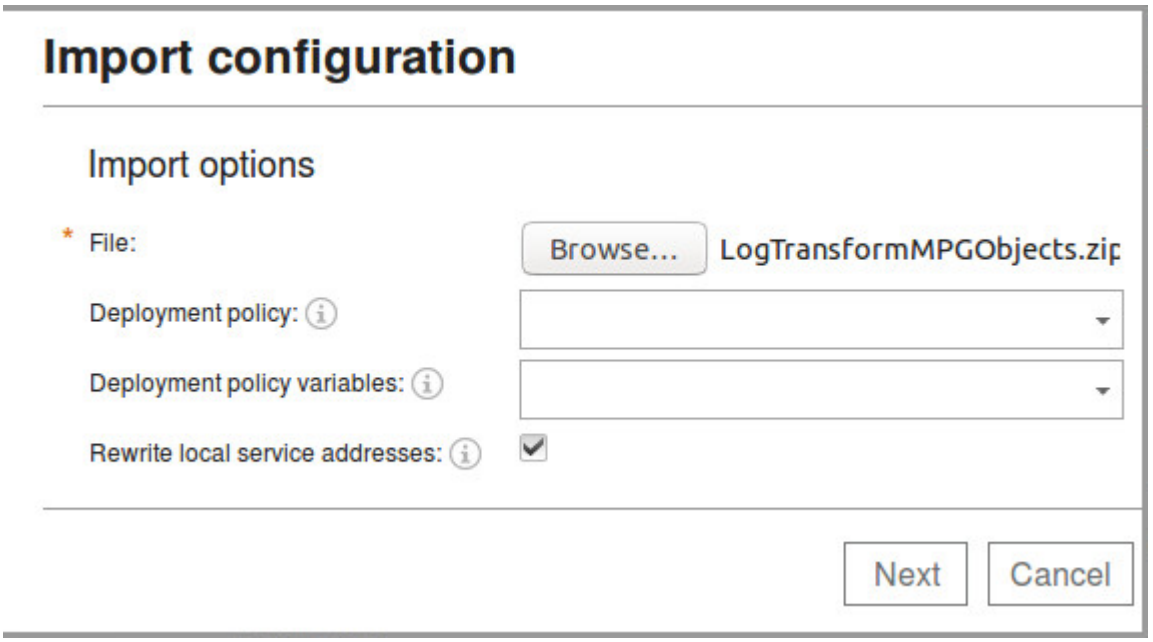
In the following section, you use a log action in a service policy to generate a log message that contains message data. First, you need to import a multi-protocol gateway service that contains the Log Action that is used for testing. After importing, you modify the service definition to your student values.

__ 1. Import the Logging multi-protocol gateway service.

__ a. Make sure that you are in your domain `student`*`nn`*`-domain` by using the `student`*`nn`* user.

__ b. From the Main menu, click **Import Configuration**.

__ c. Use **Browse** to specify `<`*`lab_files`*`>\logging\LogTransformMPGObjects.zip`. Click **Open**.



__ d. Click **Next**.

__ e. On the following page, the import wizard examines the `.zip` file, lists the objects and files that are found, and suggests what to import (selective import). In this case, follow the suggestions and click **Import**.

__ f. A page is shown that lists the results of the import. Click **Close**.

__ 2. Examine the imported LogTransformMPG service.

__ a. From the Main menu, click **Services**, then select **Multi-Protocol Gateway**.

__ b. The multi-protocol gateway catalog list is displayed. You see the LogTransformMPG service. The service can be shown as “up” or “down,” because all students are using the same port to import the same service. One service at a time uses a port. You fix that next.

__ c. Click **LogTransformMPG** to open the service.

__ d. Under **Front side settings**, click the **LogTransformHTTP_FSH** HTTP front-side handler. Then, click the **Edit** icon to open the front-side handler.

__ e. Enter the port number `<mpgw_log_port>` and click **Apply**.

Note

The <*mpgw_log_port*> is 10nn5.

__ 3. Update the Log action in the processing policy to point to your external logger.

__ a. Click the **Edit** button to modify the multi-protocol gateway policy **LogTransformLoopbackPolicy**.

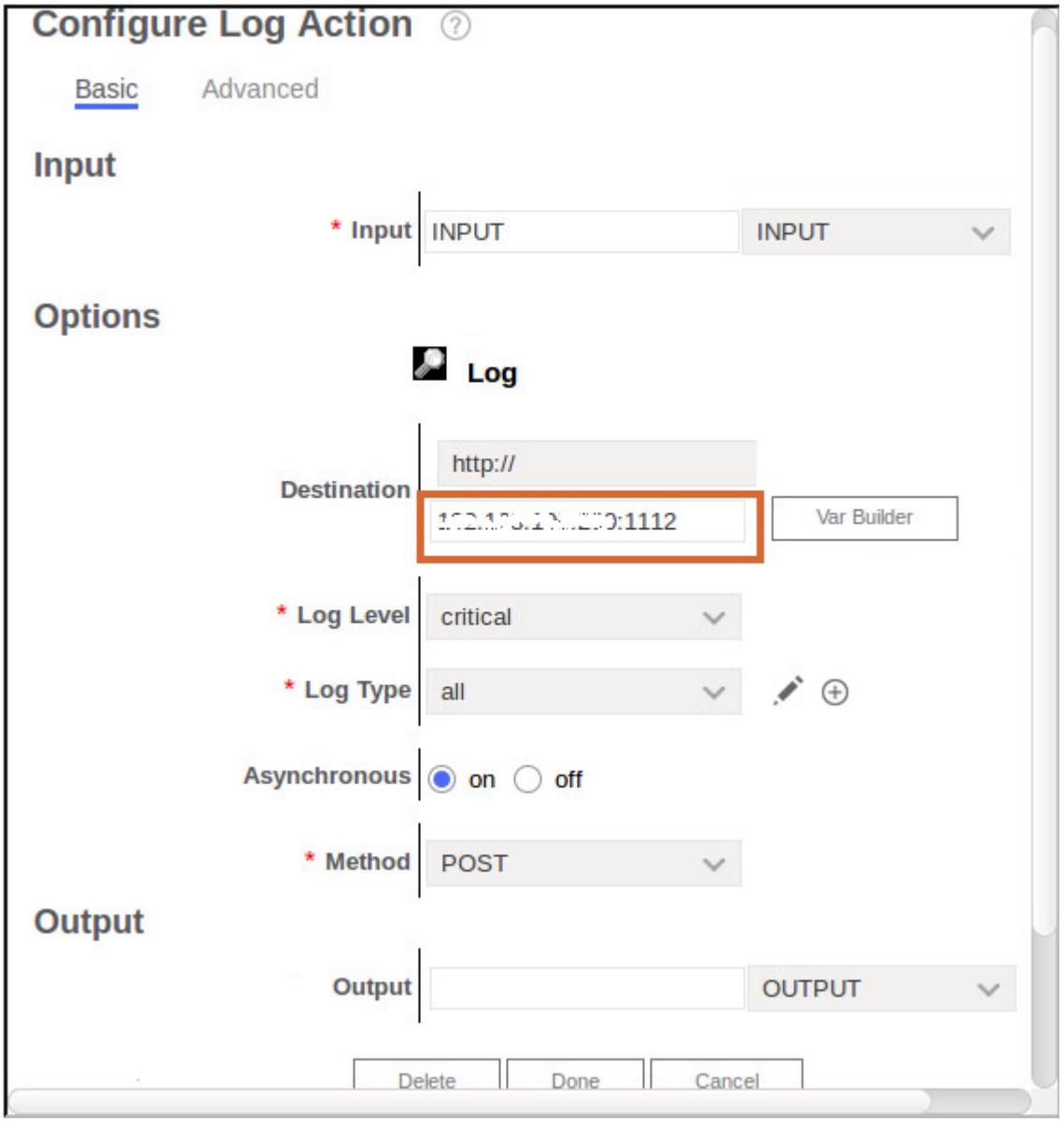
__ b. The policy editor opens. Double-click the **Log Action** icon.

__ c. Note the **Log Level** and **Log Type** that are specified; they are shown in the log entry.

__ d. The **Asynchronous** choice is set to **on**. This setting means that the log request is initiated, and then the next action is run without waiting for the log request to complete.

__ e. The **Input** identifies what gets logged. In this case, you want to log the input message from the client.

__ f. The **Destination** points to where you want to send the log. You want to log to the external logging server that listens on a specific port. Enter:
`http://<image_ip>:<logger_app_port>`



__ g. Click **Done**.

__ h. Click **Apply Policy** and close the window.

__ i. Click **Apply** to save the service definition.

5.6. Test the external logging

The “external logging system” is a Java-based server that is stored in your lab files directory. You start the logger, and then use cURL to send a message to the service. Then, you examine the logger to find your input message recorded.

__ 1. Start the external logging tool.

__ a. In a terminal window, change to `<lab_files>\DP_Logger`.

__ b. Using gedit or another text editor, open `DP_FileLogger.sh`. This shell script sets the parameters and starts the logger. Specifically, note the **port** parameter. This port is the port that the logger listens on. Make sure that it matches the destination port `<logger_app_port>` that you set in the **Log Action**.

__ c. Close the text file.

__ d. Start the logger by running `DP_FileLogger.sh`:

```
# ./DP_FileLogger.sh
```

Several threads for listening are started. Look for the **Current Log File Name**, which is the file that contains any log messages. The file name gets listed several times, but it is the same file for all threads. The logger is ready when you see:

```
Current log file name = logFile_201X-XX-XX_XX-XX-XX
DP_FileLogger init complete.  Listening for requests...
```

__ e. Keep this logger window open so the logger continues to listen.

__ 2. Send a message to the Logging multi-protocol gateway service.

__ a. Open another terminal window, and change to `<lab_files>\logging`.

__ b. The input message is transformed; `AddressReq.xml` exists in this directory. Enter the following cURL command:

```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_public_ip>:<mpgw_log_port>
```

__ c. In the output, you see the result of the `AddressReq.xml` input that is transformed into an HTML-based file. Nothing looks different from the client perspective.

__ d. In the `<lab_files>/DP_Logger` directory, open the logger output file `logFile_201X4-XX-XX_XX-XX-XX`. This name was the “current log file name” that is mentioned in the DP_FileLogger startup message. You see a log entry of type **all** and level **critical**. This type and level are what is set in the Log action. You also see the XML input message.

__ e. When you are finished with the external logger, press Ctrl+C to exit the logger.

__ f. Close the logging terminal window.

5.7. Send the Log action to a syslog-tcp destination

In this section, you modify the Log action to point to a syslog-tcp destination, send a message to the service, and examine the log file.

__ 1. Update the Log action in the processing policy to point to the syslog-tcp destination.

__ a. Click the **Edit** (...) icon to modify the multi-protocol gateway policy **LogTransformLoopbackPolicy**.

__ b. The policy editor opens. Double-click the **Log Action** icon.

__ c. The **Input** field identifies what is logged. In this case, you want to log the input message from the client, which is contained in the **INPUT** context.

__ d. The **Destination** points to where you want to send the log. You want to log to the syslog facility on the Linux image, listening on port 514. Enter: `http://<image_ip>:514`

__ e. Set the **Log Level** to **critical**, and **Log Type** to **all**; they are shown in the log entry.

__ f. The **Asynchronous** choice is set to **on**. This setting means that the log request is initiated, and then the next action is run without waiting for the log request to complete.

__ g. Leave the **Method** as **POST**.

__ h. Click **Done**.

__ i. Click **Apply Policy** and close the policy editor.

__ j. Click **Apply** and **Save changes** to commit the service definition.

__ 2. Set the system logging level to debug.

__ a. From the Main menu, click **Troubleshooting**.

__ b. The second row on this page is the **Logging** section. The **Set Log Level** controls the minimum severity of messages that are captured in the log. Set the Log Level to **debug**.

__ c. Use the **Log Level** list to set the choice to **debug**.

__ d. Click **Set Log Level**.

__ e. Confirm the change and close that window.

__ 3. Send a message to the Logging multi-protocol gateway service.

__ a. Open another terminal window, and change to `<lab_files>\logging`.

__ b. Resend the cURL command from the previous section:

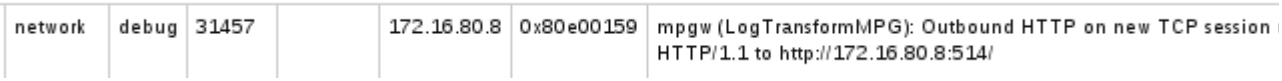
```
curl -H "Content-Type: text/xml" --data-binary @AddressReq.xml
http://<dp_public_ip>:<mpgw_log_port>
```

__ c. You get the requested address returned to you with HTML tags, as before.

__ 4. Examine the system log for the outbound message.

__ a. Refresh the system log.

__ b. You find an entry that the logging message is sent to the syslog facility:



network	debug	31457		172.16.80.8	0x80e00159	mpgw (LogTransformMPG): Outbound HTTP on new TCP session HTTP/1.1 to http://172.16.80.8:514/

Notice that this log entry is at the **debug** level. This fact is why the log level had to be changed from **error**.

__ 5. Examine the syslog-tcp log file messages.

__ a. Open a terminal window. Change to the directory where the syslog-tcp log writes these messages:

```
cd /var/log
```

__ b. Use the tail command to see the last few entries in the file:

```
tail syslog
```

__ c. A sample response is provided here:

```
Sep 18 18:21:18 172.16.78.11 Host: 172.16.80.75:514#015
Sep 18 18:21:18 172.16.78.11 Content-Length: 956#015
Sep 18 18:21:18 172.16.78.11 #015
Sep 18 18:21:18 172.16.78.11 ?xml version="1.0" encoding="UTF-8"?>
Sep 18 18:21:18 172.16.78.11 SOAP-ENV: Envelope
xmlns:dplog="http://www.datapower.com/schemas/log"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Head
er><dplog:location-tag>LogTransformLoopbackPolicy_rule_0_log_0</dplog:loc
ation-tag></SOAP-ENV:Header><SOAP-ENV:Body><log-entry
serial="1411078132327000" domain="student53_domain"><date>Thu Sep 18
2014</date><time
utc="1411078132327">18:08:52</time><type>all</type><class>mpgw</class><ob
ject>LogTransformMPG</object><level
num="2">critic</level><transaction>5541345</transaction><global-transacti
on-id>5541345</global-transaction-id><client>172.16.80.75</client><messag
e><address:getAddressInfo xmlns:address="http://dpedu.ibm.com">
Sep 18 18:21:18 172.16.78.11 #011<address: name>
Sep 18 18:21:18 172.16.78.11 #011#011<address: title>Mr.</address:title>
Sep 18 18:21:18 172.16.78.11 #011#011<address:
firstName>John</address:firstName>
Sep 18 18:21:18 172.16.78.11 #011#011<address:
lastName>Doe</address:lastName>
Sep 18 18:21:18 172.16.78.11 #011</address: name>
```

__ d. This format is a POST format, which is what you specified in the Log action. The message is the XML request that is sent to the service.

__ e. Close the terminal window.

End of exercise

Exercise review and wrap-up

In this exercise, you learned how to easily generate test events to verify that the log targets are specified correctly. You also became familiar with how to manipulate the system log view. For more specific logging, you created a custom category and separate log target. You used events to trigger CLI commands. To move logging data off the file-space limited appliance, you pointed a log target to an external logging system.

Appendix A. Master exercise variable table

Port and variable table values

The IBM Remote Lab Platform (IRLP) that runs the lab environment assigns you a student number **nn**, which you substitute as needed in the port numbers that you define for your services on the appliance. For examples, students assigned student number 01 replace nn with the student number. In this example, student 01 using port 6**nn**1, would literally use port 6**01**1 as their port number, with 01 replacing nn in the port assignment.

If you are doing this course as a self-paced virtual classroom (SPVC), then use student number 01.

This table contains the variables for the DataPower administration course WE761.

Table A-1. Variable and port assignments

Object	**Value (default)**
Student information	
<*nn*>	
<student*nn*>	student95 (development)
<student*nn*_domain>	student95_domain (development)
<student*nn*_password>	<student*nn*> student95 (development) (Must also select student95_domain)
<student*nn*_updated_password>	
<image_ip>	192.168.100.200 (development)
<student*nn*_admin_group>	
<student*nn*_admin_password>	<student*nn*_admin>
<student*nn*_admin>	
<student*nn*_developer_group>	

Non-DataPower login	
<linux_user>:	localuser
<linux_user_password>	passw0rd
<linux_root_user>	root (Ubuntu use sudo instead)
<linux_root_password>	passw0rd
DataPower information	
<dp_admin_login>	sysadmin
<dp_admin_password>	sysadminpassw0rd (development)
<dp_snmp_port>	161
<dp_internal_ip>	192.168.100.201 (development)
<dp_public_ip>	192.168.100.202 (development)
<dp_ssh_port>	22
<dp_WebGUI_port>	9090
<dp_xml_mgmt_port>	5550
Server information	
<local_address>	127.0.0.1
<logger_app_port>	1112
Application ports: Admin	
<lab_files>	(/home/localuser/labfiles/dp)

<mib_snmp_port>	10*nn*7
<mpgw_basic_port>	10*nn*6
<mpgw_cfgmgmt>	10*nn*2
<mpgw_dpadmin>	10*nn*1
<mpgw_log_port>	10*nn*5
<mpgw_loopback_port>	10*nn*4

Appendix B. Exercise solutions

This appendix describes:

- Dependencies between the exercises and other tools.
- How to load the sample solution configurations for the various exercises. The solutions are exported from the appliance into a `.zip` file. You can import a sample solution into your domain.

B.1.Dependencies

Certain exercises depend on previous exercises, and on other resources, such as the need for the back-end application server to support the service calls.

Exercises 5 does not have any solution files.

Table 2. Dependencies

Exercise	Depends on exercise	Uses cURL	Uses HTTP server
1. Upgrade Image Firmware		No	Yes
2. Using the CLI and the XML management interface to manage DataPower appliances Solution: `<lab_files>/Solutions/admin_EX2b_XML.zip`		No	Yes
3. Using the troubleshooting tools to debug errors Solution: `<lab_files>/Solutions/admin_Ex3_Troubleshooting.zip`	2	Yes	No
4. Securing connections with SSL Solution: `<lab_files>/Solutions/admin_Ex4_Crypto.zip`	2	No	No
5. Logging to an external log system Solution: `<lab_files>/Solutions/admin_Ex5_Logging.zip`	2	Yes	No

* Exercises 2 and 3 require the creation of user accounts, user group, and application domains. For more information, see the `readme.txt` file in the respective solution folders.

** You also need to upload the key files separately. The importing process does not import private or public key files.

B.2.Setting up the user accounts, user group, and domain

Note

All the exercises after exercise 2, depend on the user accounts, user group, and domains that are set up in exercise 2. Regardless of which later exercise you are intending starting with, you must first perform this part.

You must decide which student number you are going to use in the range 01 - 30.

__ 1. Click the File Manager and browse to the `<lab_files>/Solutions/CLI` directory.

__ a. Open the `CLI_Solution.xml` with an editor.

__ b. Search and replace all instances of student*nn* with your student number in the CLI_Solution.xml file.

__ c. Save the changes.

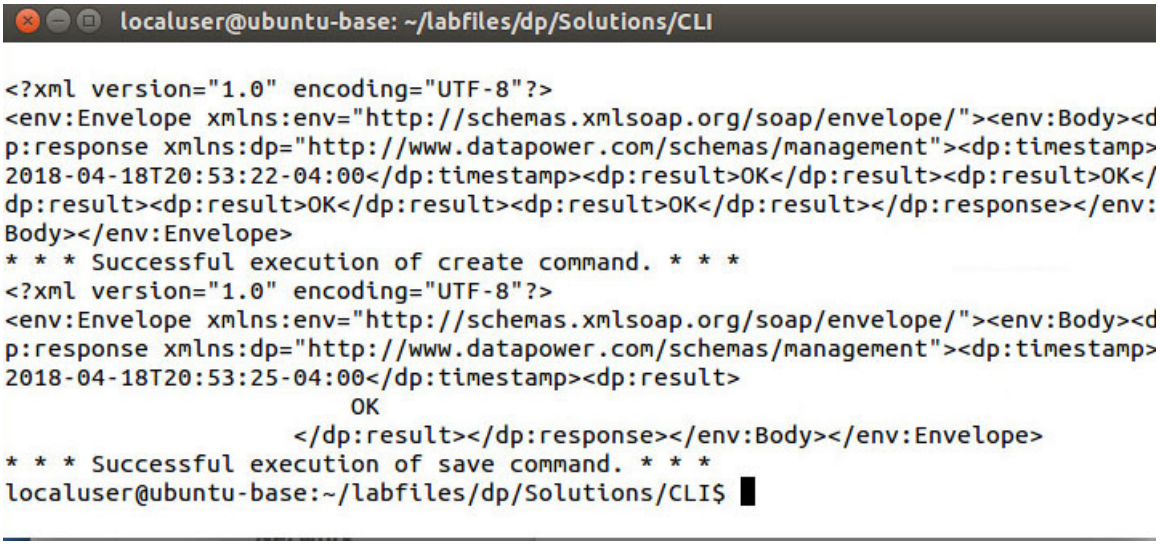
__ 2. Open a terminal and change to the `<lab_files>/Solutions/CLI` directory.

__ a. `cd ~/labfiles/dp/Solutions/CLI`

__ 3. Run the `addDomainsWithSave.sh` script in the `<lab_files>/Solutions/CLI` directory.

__ b. `./addDomainsWithSave.sh` *`<dp_internal_ip>`* `5550` *`<dp_admin_login>`* *`<dp_admin_password>`*

__ c. You should see some XML output that displays information that the create and save command ran successfully.



__ 4. Test that you can sign on to the studentnn_domain with the users that are created.

__ a. Sign on to the IDG Console https://*<dp_internal_ip>*:*9090*/dp/login.xml

__ b. Change the domain to student*nn*_domain.

__ c. Test that you can sign on with the user name <*studentnn*> and password <*studentnn*>.

__ d. Test that you can sign on with the user name <*studentnn_admin*> and password <*studentnn_admin*>.

__ e. If prompted to change your passwords, write down the passwords in the appendix table for later reference.

Note

If you are unable to sign in to the student*nn*_domain, sign on to the IDG Console by using the **default** domain with the user `<dp_admin_login>` and `<dp_admin_password>`.

From the Main menu, type `user`, then select **User Account**.

Click the studentnn in the list of users, and change the password.

Do the same for the studentnn_admin user, if required.

Write down the passwords in the appendix table for later reference.

__ 5. Sign on to the IDG Console with the user name <*studentnn*> and password <*studentnn*> and student*nn*_domain.

__ 6. Import the exercise 2 solution file `<lab_files>/Solutions/admin_EX2b_XML.zip` as described in the next part.

B.3.Importing solutions

Note

The solution files use port numbers that might already be in use. You need to change the port numbers of the imported service.

__ 1. Determine the `.zip` file to import.

__ a. The `.zip` files are named `ExNN_Name`, where `NN` represents the two-digit exercise number.

__ b. To import a solution to begin a new exercise, you need to import the solution for the previous exercise. The path to the solutions folder is listed in the table. For example, if you are ready to start Exercise 4, you would import `<lab_files>/Solutions/admin_EX2b_XML.zip` because Exercise 4 depends on Exercise 2.

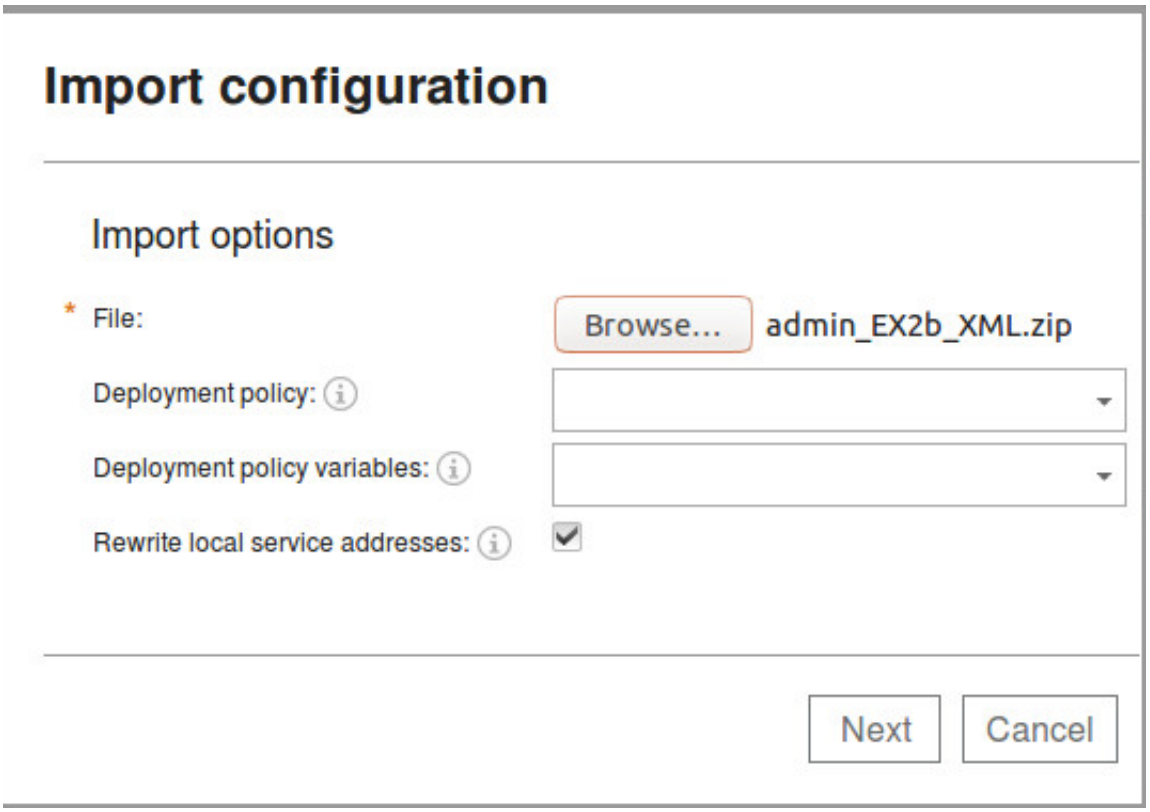
__ 2. Import the compressed (`.zip`) file that contains the solution into your application domain.

__ a. From the Blueprint Console open menu, select **Import Configuration**.

__ b. Check to see that Import Configuration has the following selections:

- From: **ZIP Bundle**
- Where: **File**

__ c. Click **Browse** and go to your respective solution `.zip` file.



__ d. Click **Next**.

__ e. In the next page, ensure that the files that are going to be imported are selected. Scroll down and click **Import**.

__ f. Make sure that the import is successful. Click **Close**.

__ 3. Upload the root DataPower certificate to the appliance.

__ a. In the DataPower Blueprint open menu, type `file`. Then, click the **File Management** link.

__ b. For the `cert:` directory, click the **Actions** link and click **Upload Files**.

__ c. Click **Browse**, go to the *<lab_files>*`/Solutions/CLI` directory, and select `Alice-sscert.pem`

__ d. Click **Upload**.

__ e. Click **Continue**.

__ f. For the `cert:` directory, click the **Actions** link and click **Upload Files**.

__ g. Click **Browse**, go to the *<lab_files>*`/Solutions/CLI` directory, and select `Alice-privkey.pem`

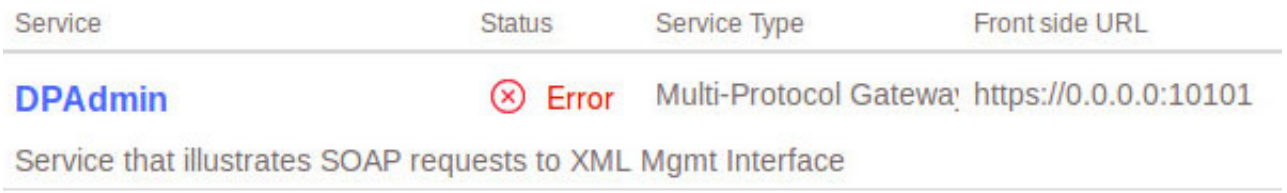
__ h. Click **Upload**.

__ i. Click **Continue**.

__ 4. Update the DPAdmin service port *<mpgw_dpadmin>*.

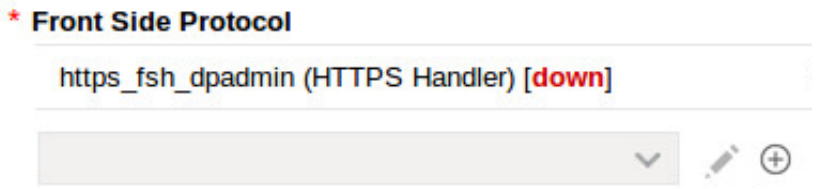
__ a. Click **Services** from the Blueprint Console.

__ b. Click the link for the **DPAdmin** service.



Notice that the status for the MPG is error. You fix this error shortly. Also, notice that the Front Side URL should be changed to 10nn1, where nn is your student number.

__ c. Click the value in the Front Side Protocol to enable editing.



__ d. Change the Port value to 10nn1, where nn is your student number.

__ e. Click **Apply**.

__ 5. Modify the Crypto Key to use the Alice private key that you uploaded.

__ a. Click **Crypto Key** in the Front Side Protocol dialog box.

__ b. Select **Enable administrative state**.

__ c. Click **Apply**.

__ d. Select administrative state.

__ e. Click **Apply**.

__ f. The status of the Crypto key changes to up.

__ 6. Modify the Crypto Certificate to use the Alice certificate that you uploaded.

__ a. Click **Crypto Certificate** in the Front Side Protocol dialog box.

__ b. Clear **Enable administrative state**.

__ c. Click **Apply**.

__ d. Select administrative state.

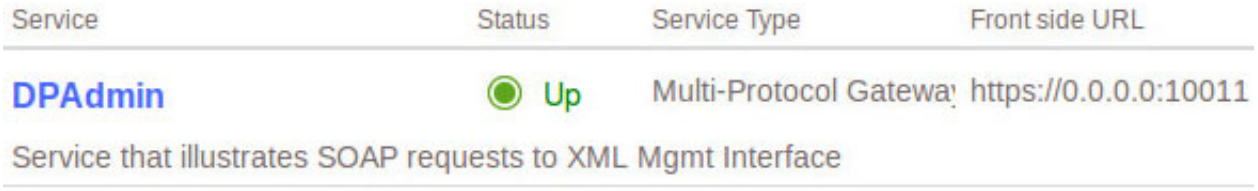
__ e. Click **Apply**.

__ f. The status of the Crypto certificate changes to up.

__ g. The status of the Front Side Protocol changes to up.

__ h. Refresh the browser.

__ 7. The DPAdmin service displays that it is up with the new *<mpgw_dpadmin>* port number.



Note

You might need to refresh the browser to view the status change.

__ 8. You have completed loading the dependencies for exercise 2, and can continue with the later exercise, or you can load the solutions for later exercises.

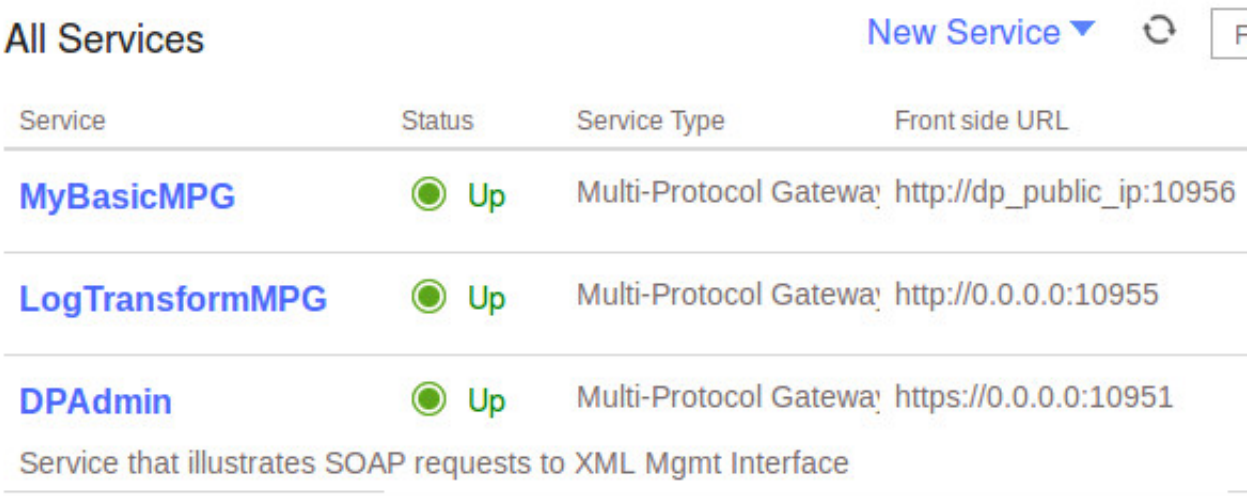
Information

Remember to change the port number for the Front Side URL in each multi-protocol gateway that you import. The port number should include your student number which might differ from the port number in the file you imported.

After you import the exercise that includes the MyBasicMPG, sign on to the default domain with the student*nn*_admin user and add a host alias for *dp_public_ip* to match the IP address on your system. You do this from the **Network > Interface > Host Alias** settings.

Before importing the solution for the Securing connections with SSL exercise, first create the StudentKeyObj as described in "Generate a certificate-key pair on the DataPower appliance" on page 4-4. Select the option to overwrite configurations that exist during the import of the solution file.

__ 9. The figure shows the running services and front side URLs that include the port numbers for student 95.



IBM Training