

GENI

Global Environment for Network Innovations

Milestone LEARN S2.c Establish ORCA framework

For eventual integration with LEARN network

Document ID: GENI-S2C-LEARN-April23

April 23, 2010

Prepared by:

Shade EL-Hadik¹, D. Gurkan¹, Ilia Baldine², Aydan Yumerefendi²

1: University of Houston: College of Technology

2: RENCI, UNC-Chapel Hill

Under Project Nr. 1733

“Programmable Measurements over Texas-based Research Network: LEARN”

TABLE OF CONTENTS

1	Document Scope.....	3
1.1	Executive Summary	3
1.2	Related Documents	3
1.2.1	GENI Documents	3
1.3	Document Revision History	3
2	Introduction	4
3	Current status of the ORCA instance in LEARN.....	5
4	Issues with the current ORCA handlers	6
5	Summary and Conclusions	7
6	Bibliography	7
7	Appendix: command mapping from Cisco 6509 to Cisco 3750.....	8

1 Document Scope

This section describes this document's purpose, its context within the overall GENI project, the set of related documents, and this document's revision history.

1.1 Executive Summary

This technical note presents the results obtained in work package "Milestone S2.c: Specifications for the Measurement Handler Software" of Project Nr. 1733, "Programmable Measurements over Texas-based Research Network: LEARN". We are trying to use a Handler software to later-on integrate ORCA framework with the LEARN network in order to start conducting meaningful optical network research experiments within the substrates, network protocols, and specifications of LEARN. This milestone addresses an attempt to create an instance of ORCA in the lab using a Cisco 3750 switch which will be the main connection network elements on LEARN later.

1.2 Related Documents

The following documents are related to this document, and provide background information, requirements, etc., that are important for this document.

1.2.1 GENI Documents

Document ID	Document Title and Issue Date

1.3 Document Revision History

The following table provides the revision history for this document, summarizing the date at which it was revised, who revised it, and a brief summary of the changes. This list is maintained in chronological order so the earliest version comes first in the list

Revis	Date	Revised By	Summary of Changes
1.0	April 2, 2010	Shade EL-Hadik	Initial draft
1.1	April 5, 2010	Deniz Gurkan	Revision
1.2	April 18, 2010	Shade EL-Hadik	Revision
1.3	April 19, 2010	Deniz Gurkan	Revision

2 Introduction

Open Resource Control Architecture (ORCA) is an extensible architecture for on-demand networked computing infrastructure. It may be viewed as a service-oriented resource control plane for an Internet operating system. Its purpose is to manage the hosting of diverse computing environments (guests) on a common pool of networked hardware resources such as virtualized clusters, storage, and network elements. This introduction has a collection of information from GENI-ORCA wiki pages as listed in the Bibliography section.

ORCA is a resource control plane organized around resource leasing as a foundational abstraction. The architecture of ORCA does not impose any particular structure on the shared resources, which means that it is possible to instantiate any experimental configuration for purposes of testing and deployment. The ORCA framework is one possible way to realize the goals of NSF's GENI project . ORCA is compatible with much of the experimental GENI architecture. Some concepts and entities in GENI map directly on to corresponding concepts and entities within ORCA.

Both GENI and ORCA are built on top of a substrate of physical resources. The GENI Management Core (GMC) corresponds to the site authority and broker actors in the ORCA system. In both cases, user services run on hosted resources mediated by a management layer.

The ORCA family consists of a number of projects. This page provides a comprehensive list of them. Some of the projects consist of one or more subprojects.

Name	Description
Automat	Umbrella project for the Autonomic Computing Testbed.
Boot	Provides tools to processes ORCA container configuration files.
Container	Defines the API for managing application containers.
Cluster on Demand	Provides a site authority implementation.
Drivers	Offers a collection of resource drivers to be used by site authorities.
Log	The Log family of projects deals with collecting and analyzing events.
Handlers	A collection of configuration handlers for network elements.
Manage	Provides the basic management framework and offers a number of tools to manage machines, storage servers, etc.
Node Agent	The Node Agent is a multi-purpose web service.
Node Agent	The Node Agent Host is a hosting standalone server for the Node Agent service.

Host	
Policy	Provides a collection of policies for service managers, brokers, and site authorities.
Portal	Provides the ORCA web portal core and a collection of extensions.
Run	Provides a harness for running ORCA experiments.
Security	A library for performing authentication and authorization.
SHARP	Implementation of SHARP tickets and accountability protocols.
Shirako	Shirako leasing core.
Tools	Offers a collection of build and management tools.
Util	A general-purpose library of useful utilities.

The project we are utilizing for this milestone is the **Handler**. **In this project we are building a customized handler for the CISCO switch 3750.** Handlers will be covered in more detail in the next sections.

3 Current status of the ORCA instance in LEARN

We have installed all the pre-requisites necessary for creation of an ORCA Framework. We followed guidelines provided through the following GENI WIKI page “<https://geni-orca.renci.org/trac/wiki/Prerequisites>” We have a dedicated PC in our lab for this installation. The prerequisite packages include but not limited to

- Java version 1.5.x,
- Ant 1.7.0+ ,
- Maven 2.2.1+ ,
- Subversion client with *https* support,
- ssh client...etc

We have installed the official release of ORCA and established a new security certificate for that particular build, as instructed by the ORCA wiki page at <https://geni-orca.renci.org/trac/wiki/buildInstructions>. As per directions on another orca wiki page “<https://geni-orca.renci.org/trac/wiki/instructions>”, we have deployed an ORCA instance locally after installing Tomcat (java application server) and populated MSQl database with the corresponding ORCA schema. We ran and tested this build locally and we managed to run ORCA web application under the following URL “<http://localhost:8080/orca>”.

Finally, we installed a node agent on another machine. Therefore, and as per instructions found on this wiki page <https://geni-orca.renci.org/trac/wiki/DeployNodeAgent>, we installed a Node Agent Host, and then the real Node Agents (NA) with their network drivers. NA machine is configured to run driver actions on behalf of the first ORCA instance. However, the NA will not be used to

initiate switch control as instructed by the ORCA team during the GEC7. Therefore, we will be developing a handler based on the Cisco 6509 version for the Cisco 3750 switches.

4 Issues with the current ORCA handlers

We have been testing handlers with NA components until March 2010, and in the process, ORCA team informed us that the driver we were trying to utilize in order to establish VLAN is not ready to work in the approved manner. Furthermore, we have learnt that the approach to run drivers from a Node Agent service happened to be problematic for researchers to bring into play. The ORCA team informed us that their future release of drivers will not use the node agent model. And since our CISCO 3750 provides SSH version of remote access, we now are developing a customized version of a java driver using SSH/Telnet API, which will run directly on the ORCA instance.

With the support of ORCA team, especially Ilia Baldine and Aydan Yumerefendi, we have customized a java driver using the following steps.

1. Obtaining an ORCA source code (from trunk) and then conducting the usual preparation and building instructions.
2. Making sure that the java class “Cisco6509Device” exists in the network/handlers project and more specifically under the following package “orca.handlers.network.router”

```
public class Cisco6509Device extends CiscoRouterDevice
{
    public Cisco6509Device(String deviceAddress,
String uid, String password, String adminPassword)
{
    super(deviceAddress, uid, password,
adminPassword);
    basepath =
"/orca/handlers/network/router/cisco/6509";
}
}
```

3. Defining a class to represent the 3750 switch device in the same way by changing the “basepath” string to point to the expected xml files for the 3750 switch.
 - For example: Change the expected output from “6509” to “3750” as in the following example. “<expect>6509</expect>” should be changed to “<expect>3750</expect>”
4. Using “test.xml” file, in the root of the project, to invoke the ant tasks. This ant task works as a *temporary handler* for the purpose of testing the

customized driver. Later on we will develop a customized handler that will take in consideration the proposed architecture for integrating LEARN.

5. Modifying the property file

“handlers/network/ant/tests.properties” where properties to be used by test.xml are stored. The new file will include the following properties:

```
emulation=false
test.mode=true
router.user=teamion
router.password=uhcotnet230dt2
router.adminpassword=uhcotnet230dt2
router=208.117.132.141
vlan.tag=100
router.ports=gigabitethernet 1/25
router.src.vlan.tag=123
router.dst.vlan.tag=456
router.map.port=gigabitethernet 1/26
```

You can invoke the test handler from the command line “ant -f test.xml -D test.mode=false cisco.6509.createVlan”

5 Summary and Conclusions

We are trying to use the Handler software to later-on integrate ORCA framework with the LEARN network in order to start conducting meaningful optical network research experiments within the substrates, network protocols, and specifications of LEARN. We have now a fully customized version of ORCA driver that we can use as a test handler in order to test the manipulation of a network resource, CISCO 370 switch as a step toward understanding the best architecture of integrating LEARN with ORCA.

6 Bibliography

<https://geni-orca.renci.org/orca-doc/current/>

<https://geni-orca.renci.org/trac/attachment/wiki/WikiStart/ORCA%20Book.pdf>

<https://geni-orca.renci.org/trac/wiki/buildInstructions>

<https://geni-orca.renci.org/trac/wiki/instructions>

<https://geni-orca.renci.org/trac/wiki/DeployNodeAgent>

<https://geni-orca.renci.org/trac/wiki/Simple%20Driver%20HOWTO>

<https://geni-orca.renci.org/trac/wiki/Driver%20Template%20HOWTO>

7 Appendix: command mapping from Cisco 6509 to Cisco 3750

ADDCLIENTPORTS

Cisco6509AddClientPorts.Xml: This script is used to Add Client ports whose port numbers are to be chosen from the dynamic port range i.e. from 49152 to 65535.

Cisco 6509	Cisco 3750
<p><u>config t</u></p> <p>A command "config t" was sent to the console.</p>	<p>conf t / config t / configure terminal</p>
<p>A response beginning with "6509" is to be expected from the console.</p>	<p>Switch(config)#</p> <p>Enter into the configuration mode</p>
<p><u>interface range port-range</u></p> <p>This command allows us to specify the interface range. I.e. using this we can execute a command on multiple ports at the same time. We can use this command on existing VLAN SVI's only. The Gbports specifies the port range.</p>	<p>interface range FastEthernet1/0/7-24 .</p> <p>Note: The 3750 switch has only two of Gigabit Ethernet ports and 24 Fast Ethernet ports. So we might be using mainly the Fast Ethernet ports</p>
<p>After a timeout of 4000 secs a response is to be expected from the console.</p>	<p><u>Action:</u> switch (config-if) #</p> <p>With the above command, after few seconds we enter into the interface configuration mode</p>
<p><u>switchport trunk encapsulation dot1q</u></p> <p>This command is used to set the trunk characteristics when the interface is in trunking mode and set the encapsulation</p>	<p>switchport trunk encapsulation dot1q</p> <p>Note: The above command causes a port configured as a switched interface to encapsulate in 802.1Q trunking format</p>

<p>format on the trunk port to IEEE 802.1Q. With this format, the switch supports simultaneous tagged and untagged traffic on a port.</p>	<p>regardless of its default trunking format in trunking mode</p> <p>Trunking Mode: A switch's database has information about a local VLAN, and VLAN information is not passed between switches. Trunking mode enables the VLAN information to be exchanged between switches.</p>
<p>6509</p> <p>A response beginning with "6509" is to be expected from the console.</p>	<p>Action: Switch(config-if)#</p> <p>Waits for the port to be configured as a switch interface</p>
<p><u>switchport trunk allowed vlan vlan-list</u></p> <p>This command is used to set the list of allowed VLANs that can receive and send traffic on this interface in tagged format when in trunking mode. The VLANTagName should also be specified.</p>	<p>Switch(config)# interface</p> <p>gigabitethernet1/0/2</p> <p>Switch(config-if)# switchport trunk allowed vlan add 1,2,5,6</p> <p>Note: This example shows how to add VLANs 1, 2, 5, and 6 to the allowed list that can receive and send traffic on the gigabit Ethernet ports</p> <p>PS: No reference to Vlan-tagging could be found.</p>
<p>After a timeout of 3000 secs a response beginning with "6509" is to be expected from the console.</p>	<p>Action: no action</p> <p>Takes few seconds to add the vlans to the allowed list.</p>

<p><u>switchport mode trunk</u></p> <p>This command causes the interface to go into trunking mode permanently and negotiates to convert the link into a trunk link even if the neighboring interface does not agree to the change. A trunk is a point-to-point link between two switches or between a switch and a router.</p>	<p>Switch(config)# interface gigabitethernet2/0/1</p> <p>Switch(config-if)# switchport mode trunk</p> <p>Note: It is observed that when the interface which is going into the trunking mode permanently is usually a gigabit Ethernet port</p>
<p>A response is to be expected from the console.</p>	<p>_Action: no action</p> <p>Waits for few seconds to go into the trunkmode</p>
<p><u>no shut</u></p> <p>This command is used to enable an interface i.e. in this case the interface on which the VLAN was created will be enabled (interface will not be shutdown).</p>	<p>Switch (Config-if)# no shutdown</p> <p>If you do not set a recovery mechanism, you must enter the shutdown and then the no shutdown commands to manually recover an interface from the error-disabled state.</p>
<p>A response is to be expected from the console.</p>	<p><u>Switch(config)#</u></p> <p>waits for few seconds and comes out of the interface mode.</p>
<p><u>End</u></p> <p>This command is issued to return to the global config mode or the privileged EXEC mode.</p>	<p>end/exit</p>

<p>A response is to be expected from the console.</p>	<p><u>Action:</u> “ Press Return to enter console” message is displayed. Waits few seconds to exit from the privileged mode to user mode.</p>
---	---

ADD PORTS TO CISCO6509 CONSOLE

Cisco6509Addports.Xml: The purpose of this script is to add ports to the Cisco6509 console. **<action name="AddPorts">**: This specifies the action or the purpose of the commands being used. It is clear that the following commands will be used to Addports.

</action>

<exchange> </exchange>: This tag specifies that commands have been exchanged between the console(switch) and device(workstation).

<p><u>config t</u> This command "config t" was sent to the console. I.e. A communication channel has been initiated by the user</p>	<p>config t/ configure terminal/ conf t</p>
---	---

<p>A response is to be expected. An acknowledgement sent out by the console (6509) that it has participated in the communication.</p>	<p><u>Action:</u> Switch(config)# Enter into the configuration mode</p>
---	--

SENDING A REQUEST TO ADD PORTS

Cisco6509AddPortsRequest.Xml: The purpose of this script is to send a request to add ports to the console.

This xml file calls 3 scripts namely :

<script>Cisco6509Log_on</script>

<script>Cisco6509AddPorts</script>

<script>Cisco6509Log_off</script>

Scripts are identical to functions and methods in programming which perform a specific task.

<script>Cisco6509Log_on</script>: This script is used to login in to the console.	Switch>enable Password: uhcotnet230 Switch#
<script>Cisco6509AddPorts</script>: This script is used to add ports to the console.	Please Refer to the related commands given under “ Add ports to console”
<script>Cisco6509Log_off</script>: This script is used to log off from the console.	Switch#exit Switch>

ADD TRUNKPORTS

Cisco6509AddTrunkPorts.Xml: This script contains commands to add trunk ports.

Trunk port: A trunk port is a port that has been configured to carry traffic for any or all of the VLAN's associated with a switch.

<p><u>config t</u></p> <p>A command "config t" was sent to the console.</p>	<p>conf t/ config t/configure terminal</p>
<p>A response is to be expected from the console.</p>	<p><u>Actions:</u></p> <p>Switch (Config)#</p> <p>Waits for entering into the configuration terminal</p>
<p><u>interface range port-range</u></p> <p>This command allows us to specify the interface range. i.e. using this we can execute a command on multiple ports at the same time.</p>	<p><u>Switch(config)#</u> interface range</p> <p>FastEthernet1/0/2-6 or gigabit Ethernet 1/0/3-7</p> <p><u>Please note:</u> the features of 6509 are more advanced and complicated than 3750 switches. As we can infer, there are many gigabit Ethernet ports being used.</p>
<p>After a timeout of 4000 secs a response is to be expected from the console.</p>	<p><u>Action:</u></p> <p>Switch(config-if)#</p> <p>Waits for few seconds for the previous command to execute.</p>

<p><u>switchport trunk encapsulation dot1q</u></p> <p>This command is used to set the trunk characteristics when the interface is in trunking mode and set the encapsulation format on the trunk port to IEEE 802.1Q. With this format, the switch supports simultaneous tagged and untagged traffic on a port.</p>	<p>Switch (config-if) # switchport trunk encapsulation dot1q</p> <p>Note: The above command causes a port configured as a switched interface to encapsulate in 802.1Q trunking format regardless of its default trunking format in trunking mode</p>
<p>A response is to be expected from the console.</p>	<p><u>SW3750Action:</u></p> <p>No action</p> <p>Switch(config-if)#</p> <p>Waits for few seconds and the prompt appears</p>
<p><u>switchport trunk allowed vlan <i>vlan-list</i></u></p> <p>This command is used to set the list of allowed VLANs that can receive and send traffic on this interface in tagged format when in trunking mode.</p>	<p>Switch(config)# interface gigabitethernet1/0/2</p> <p>Switch(config-if)# switchport trunk allowed vlan add 1,2,5,6 Note: This example shows how to add VLANs 1, 2, 5, and 6 to the allowed list that can receive and send traffic on the gigabit Ethernet ports</p> <p>PS: No reference to Vlan-tagging could be found.</p>
<p>After a timeout of 4000 secs a response is</p>	<p><u>Action:</u> no action</p>

<p>to be expected from the console.</p>	<p>Switch(config-if)#</p> <p>Waits for few seconds and the prompt appears</p>
<p><u>switchport mode trunk</u></p> <p>This command causes the interface to go into trunking mode permanently and negotiates to convert the link into a trunk link even if the neighboring interface does not agree to the change. A trunk is a point-to-point link between two switches or between a switch and a router.</p>	<p>Switch(config)# interface</p> <p>gigabitethernet2/0/1</p> <p>Switch(config-if)# switchport mode trunk</p> <p>Note: It is observed that when the interface which is going into the trunking mode permanently is usually a gigabit Ethernet port</p>
<p>A response beginning with "6509" is to be expected from the console.</p>	<p><u>Action:</u></p> <p>Switch(config-if)#</p> <p>Waits for few seconds and the prompt appears</p>
<p><u>no ip address</u></p> <p>This command is used to disable the IP processing on the interface range provided above.</p>	<p>Switch(Config-if): no ip address ip-address subnet-mask</p> <p>Note: Use the ip address interface configuration command on the switch stack or on a standalone switch to set an IP address for the Layer 2 switch or an IP address for each switch virtual interface (SVI) or routed port on the Layer 3 switch.</p>

	Use the no form of this command to remove an IP address or to disable IP processing.
A response beginning with "6509" is to be expected from the console.	Action: Waits for few seconds and the prompt appears
<u>logging event link-status</u> This command is used to enable system logging of interface state-change events on all the interfaces.	No relevant command found
A response beginning with "6509" is to be expected from the console.	Action: Waits for few seconds and the prompt appears
<u>no shut</u> This command is used to enable an interface i.e. in this case the interface on which the VLAN was created will be enabled (interface will not be shutdown).	Switch (Config-if)# no shutdown If you do not set a recovery mechanism, you must enter the shutdown and then the no shutdown commands to manually recover an interface from the error-disabled state.
A response beginning with "6509" is to be expected from the console.	Action: Waits for few seconds and the prompt appears Switch#
<u>End</u>	end/exit

<p>This command is issued to return to the global config mode or the privileged EXEC mode.</p>	
<p>A response beginning with "6509" is to be expected from the console.</p>	<p>Action: Waits for few seconds and the prompt appears</p> <p>Switch></p> <hr/>

Create VLAN

Cisco6509CreateVLAN.Xml: This script is used to create a VLAN.

<p><u>config t</u></p> <p>A command "config t" was sent to the console.</p>	<p>conf t/ config t/configure terminal</p>
<p>A response beginning with "6509" is to be expected from the console.</p>	<p>Actions: Waits for entering into the configuration terminal</p> <p>Switch (Config)#</p>
<p>Vlan_vlan-id</p> <p>This specifies that the user creates a VLAN by sending the command vlan along with the VLAN name.</p> <p>A timeout of 3000 secs a response beginning with "6509" is to be expected from the console.</p>	<p>Switch(config)# vlan vlan-id</p> <p><i>Vlan -id</i> :VLAN identification. The range is 1 to 4094.</p> <p>Action: Waits for few seconds and the prompt appears</p> <p>Switch(config-vlan)#</p>
<p>name vlan-name</p> <p>This command specifies that the user will</p>	<p>Switch (config-vlan)# name <i>vlan0013</i></p> <p>Vlan0013 is the name of the vlan. The</p>

<p>enter a name for the VLAN and send it to the console.</p> <p>After a timeout of 2000 secs a response beginning with "6509" is to be expected from the console.</p> <p><u>Exit</u></p> <p>This command specifies that a user sends an exit command to log off from the console.</p>	<p>default vlan name is vlan1.</p> <p>name <i>vlan-name</i>: names the VLAN with an ASCII string from 1 to 32 characters that must be unique within the administrative domain. The default is <i>VLANxxxx</i> where <i>xxxx</i> represents four numeric digits (including leading zeros) equal to the VLAN ID number.</p> <p>Action: Waits for few seconds and the prompt appears</p> <p>Switch(config-vlan)#</p> <p>exit/end</p>
<p>A response beginning with "6509" is to be expected from the console.</p> <p><u>interface vlan vlan-id</u></p> <p>This command specifies that the user enters the interface on which a VLAN should be created along with the VLAN id. An SVI (Switch Virtual Interface) is created with this command</p> <p>After a timeout of 3000 secs a response beginning with "6509" is to be expected from the console.</p>	<p>Action: Gets out of the configuration mode after few seconds</p> <p>Switch(config)#</p> <p>Switch(config)# interface vlan <i>vlan-id</i></p> <p>Use the interface vlan global configuration command on the switch stack or on a standalone switch to create or access a dynamic switch virtual interface (SVI) and to enter interface configuration mode. Use the no form of this command to delete an SVI.</p> <p>Action: Waits for few seconds and the prompt appears</p> <p>Switch(config-if)#</p>
<p><u>logging event link-status</u></p> <p>This command is used to enable system</p>	<p>No Relevant command Found</p>

<p>logging of interface state-change events on all the interfaces.</p> <p>A response beginning with "6509" is to be expected from the console.</p> <p><u>no shut</u></p> <p>This command is used to enable traffic an interface i.e. in this case the interface on which the VLAN was created will be able to receive traffic.</p> <p>A response beginning with "6509" is to be expected from the console.</p> <p><u>End</u></p> <p>This command is issued to return to the global config mode or the privileged EXEC mode.</p> <p>A response beginning with "6509" is to be expected from the console.</p>	<p>Waits for few seconds and the prompt appears as : Switch(config-if)#</p> <p>Switch (Config-if)# no shutdown</p> <p>If you do not set a recovery mechanism, you must enter the shutdown and then the no shutdown commands to manually recover an interface from the error-disabled state.</p> <p>Waits for few seconds and the prompt appears : Switch(config-if)#</p> <p>exit/end</p> <p>Waits for few seconds and the prompt appears :</p> <p>Switch></p>
---	---

Request for Creating a VLAN

Cisco6509CreateVLANRequest.Xml: The purpose of this script is to request for the creation of a VirtualLAN.

This xml file calls 5 scripts namely :

<script>Cisco6509Log_on</script>: This script is used to login in to the console.

<script>Cisco6509CreateVLAN</script>: This script is used to create a VirtualLAN.

<script>Cisco6509AddClientPorts</script>: This script is used to Add Client ports.

<script>Cisco6509AddTrunkPorts</script>: This script is used to Add Trunk ports.

<script>Cisco6509Log_off</script>: This script is used to log off from the console.

DELETE VLAN

Cisco6509DeleteVLAN.Xml: This script is used to delete a VLAN.

<p><u>config t</u></p> <p>A command "config t" was sent to the console.</p> <p>A response beginning with "6509" is to be expected from the console.</p> <p><u>no vlan vlan-id</u></p> <p>This specifies that the user deletes a VLAN by sending the command no vlan along with the VLAN id.</p> <p>After a timeout of 4000 secs a response beginning with "6509" is to be expected from the console</p> <p><u>End</u></p> <p>This command is issued to return to the global config mode or the privileged EXEC mode.</p> <p>A response beginning with "6509" is to be expected from the console.</p>	<p>conf t/ config t/configure terminal</p> <p>Waits for few seconds and the prompt appears</p> <p>Switch (config)#</p> <p><u>Switch(config)# no vlan vlan-id</u></p> <p>Use the vlan global configuration command on the switch stack or on a standalone switch to add a VLAN and to enter the config-vlan mode. Use the no form of this command to delete the VLAN. Configuration information for normal-range VLANs (VLAN IDs 1 to 1005) is always saved in the VLAN database. When VLAN Trunking Protocol (VTP) mode is transparent, you can create extended-range VLANs (VLAN IDs greater than 1005), and the VTP mode, domain name, and the VLAN configuration are saved in the switch running configuration file.</p> <p>Waits for few seconds and the prompt appears as : Switch(config)#</p> <p>exit/end</p> <p>Waits for few seconds and the prompt appears</p> <p>Switch></p>
---	---

Request for Deleting a VLAN

Cisco6509DeleteVLANRequest.Xml: The purpose of this script is to request for the removal of a Virtual LAN creation request.

This xml file calls 5 scripts namely:

<script>Cisco6509Log_on</script>: This script is used to login in to the console.

<script>Cisco6509RemoveClientPorts</script>: This script is used to Remove the Client ports.

Please refer to the command sets for “Remove Client Ports” as given below

<script>Cisco6509RemoveTrunkPorts</script>: This script is used to Remove Trunk ports.

Please refer to the command sets for “Remove trunkPorts” as given below

<script>Cisco6509DeleteVLAN</script>: This script is used to delete a VirtualLAN.

Please refer to the command sets for “Delete VLAN” as given above

<script>Cisco6509Log_off</script>: This script is used to log off from the console.

LOG OFF

Cisco6509Log_off.Xml: This script is used to log off from the console or disconnect.

Exit

This command specifies that a user sends an exit command to log off from the console.

Eof

This command specifies that the user is expecting an eof signal from the console that acknowledges the disconnection.

Switch# exit

Press RETURN to enter message is seen on the screen

LOG ON

Cisco6509Log_on.Xml: This script is used to login to the console (username and password required)

En

A username has been entered and sent to the console

Switch>en

A password is expected from the user to login and a time out of 2000 secs has been set for the user to enter the password.

Password : (User is expected to type the password)

Password has been sent by the user.

At the password prompt type
"uhcotnet230"

The user is expecting a response from the console. (At CLI, it means that the switch enters into Privileged mode)

The switch enters into the privileged mode

Switch#

MAPPING VLANS

Cisco6509MapVLANS.Xml: This script is used to map the VLANS.

config t

A command "config t" was sent to the console.

conf t/ config t/configure terminal

A response beginning with "6509" is to

<p>be expected from the console.(At the CLI it implies that the switch enters into the configuration mode)</p> <p>interface port</p> <p>This command is used to configure an interface with a specified port number.</p> <p>After a timeout of 3000 secs a response beginning with "6509" is to be expected from the console.(Implies that after few seconds, the terminal for configuration of the port specified is displayed at CLI)</p> <p><u>Switchport</u></p> <p>This command is used to modify the switching characteristics of the Layer 2-switched interface.</p> <p>A response beginning with "6509" is to be expected from the console.(It implies that the modification is registered)</p> <p>switchport vlan mapping enable</p> <p>This command is used to enable VLAN</p>	<p>Waits for entering into the configuration terminal</p> <p>Switch (Config)#</p> <p>Switch(config)#interface fastethernet 1/0</p> <p>Waits for few seconds and the prompt appears as :</p> <p>Switch(config-if)#</p> <p>Switch(config-if)#switchport</p> <p>Waits for few seconds and the prompt appears:</p> <p>Switch(config-if)#</p> <p>Note: Use “show”command to see if the changes are registered or not.</p>
---	---

<p>mapping per switch port.</p> <p>A response beginning with "6509" is to be expected from the console.(It implies that the modification is registered)</p> <p>switchport vlan mapping <i>vlan-id translated-id</i></p> <p>This command is used to map the traffic arriving on the VLAN original-vlan-id (Source VLAN) to the VLAN translated-vlan-id (Destination VLAN) and the traffic that is internally tagged with the VLAN translated-vlan-id with the VLAN original-vlan-id before leaving the port.</p> <p>A response beginning with "6509" is to be expected from the console.</p> <p><u>no shut</u></p> <p>This command is used to enable an interface i.e. in this case the interface on which the VLAN was created will be enabled.</p> <p>A response beginning with "6509" is to</p>	<p>No relevant command found</p> <p>Waits for few seconds and the prompt appears:</p> <p>Switch(config-if)#</p> <p>Note: Use “show running-config” command to see if the changes are registered or not.</p> <p>switchport vlan mapping <i>vlan-id translated-id</i></p> <p>Use the show vlan mapping : Verify the configuration</p> <p>Waits for few seconds and the prompt appears as:</p> <p>Switch(config-if)#</p> <p>Switch(config-if)#no shutdown</p>
---	--

<p>be expected from the console.</p> <p><u>End</u></p> <p>This command is issued to return to the global config mode or the privileged EXEC mode.</p>	<p>Waits for few seconds and the prompt appears as:</p> <p>Switch(config-if)#</p> <p>exit/end</p>
<p><u>Request for Mapping VLANs</u></p> <p><u>Cisco6509MapVLANsRequest.Xml:</u> This script is used to send a request for the mapping of VLANs.</p> <p>This XML file mainly calls 3 scripts:</p> <p><u><script>Cisco6509Log_on</script>:</u> This script is used to login in to the console.</p> <p><u><script>Cisco6509MapVLANs</script>:</u> This script is used to map the VLANs.</p> <p><u><script>Cisco6509Log_off</script>:</u> This script is used to logoff from the console.</p>	
<p><u>Remove Client Ports</u></p> <p><u>Cisco6509RemoveClientPorts.Xml:</u> This script is used to remove the Client ports that have already been added.</p>	
<p><u>config t</u></p> <p>A command "config t" was sent to the console.</p>	<p>conf t/ config t/configure terminal</p>
<p>A response beginning with "6509" is to be expected from the console.</p>	<p>Waits for entering into the configuration terminal</p> <p><u>Switch (Config)#</u></p>
<p><u>interface range port-range</u></p>	<p>interface range FastEthernet1/0/7-24 .</p>

This command allows us to specify the interface range. i.e. using this we can execute a command on multiple ports at the same time.

After a timeout of 3000 secs a response beginning with "6509" is to be expected from the console.(At CLI, the switch enters into the interface configuration mode)

switchport trunk allowed vlan remove *vlan-list*

This command is used to manually remove VLAN's from a trunk link (i.e. VLAN's attached to a trunk port). The trunk link contains all the VLAN's that exist on a switch.

Note: The 3750 switch has only two of Gigabit Ethernet ports and 24 Fast Ethernet ports. So we might be using mainly the Fast Ethernet ports

Waits for few seconds and the prompt appears as **:Switch(config-if)#**

switchport trunk allowed vlan remove *vlan-list*

Define the VLANs that are *not* allowed to transmit and receive on the port. The *vlan-list* parameter is a range of VLAN IDs separated by a hyphen or specific VLAN IDs separated by commas.

Example: This example shows how to define the allowed VLANs list for trunk port Fa0/1 to allow VLANs 1 to 100, VLAN 250, and VLANs 500 to 1005, and how to verify the allowed VLAN list for the trunk:

```
Switch(config)# interface fa0/1
```

```
Switch(config-if)# switchport mode trunk
```

```
Switch(config-if)# switchport trunk allowed vlan remove 101-499
```

```
Switch(config-if)# switchport trunk
```

<p>After a timeout of 4000 secs a response beginning with "6509" is to be expected from the console.(At CLI, the switch waits for the getting to know about the modifications and the prompt returns)</p> <p><u>End</u> This command is issued to return to the global config mode or the privileged EXEC mode.</p> <p>A response beginning with "6509" is to be expected from the console.(At CLI, it comes out of the privileged mode)</p>	<pre> allowed vlan add 250 Switch(config-if)# end Switch# show interface fa0/1 switchport allowed-vlan "1-100,250,500-1005" → Waits for few seconds and the prompt appears as: Switch# exit/end Waits for few seconds and the prompt appears as : Switch> </pre>
---	---

Remove Ports from Console

Cisco6509RemovePorts.Xml: The purpose of this script is to remove ports from the Cisco6509 console.

<action name="RemovePorts"> This specifies the action or the purpose of the

commands being used. It is clear that the following commands will be used to Remove ports.</action>

<exchange> </exchange>: This tag specifies that commands have been exchanged between the console(switch) and device(workstation).

config t The command conf t was sent to the console A response beginning with "6509" is to be expected from the console.	conf t/ config t/configure terminal Waits for entering into the configuration terminal : <u>Switch (Config)#</u>
---	---

Request for Removing Ports

Cisco6509RemovePortsRequest.Xml: This script is used to request for the deletion of the added ports.

This XML file mainly calls 5 scripts:

<script>Cisco6509Log_on</script>: This script is used to login in to the console. Please refer to the commands in the “log on” section above

<script>Cisco6509RemovePorts</script>: This script is used to Remove the ports. Please refer to the commands in the “RemovePorts” section above

<script>Cisco6509RemoveClientPorts</script>: This script is used to Remove Client ports. Please refer to the commands in the “Remove Client Ports” section above

<script>Cisco6509RemoveTrunkPorts</script>: This script is used to Remove Trunk ports.. Please refer to the commands in the “Remove Trunk Ports” section below

<script>Cisco6509Log_off</script>: This script is used to log off from the console. Please refer to the commands in the “log off” section above

Remove Trunk ports

Cisco6509RemoveTrunkPorts.Xml: This script is used to Remove the Trunk ports that have already been added.

config t

A command "conf t" was sent to the console

A response beginning with "6509" is to be expected from the console (At CLI, the switch enters into the configuration mode)

interface range port range

This command allows us to specify the interface range. I.e. using this we can execute a command on multiple ports at the same time.

After a timeout of 2000 secs a response beginning with "6509" is to be expected from the console. (Switch enters into the interface configuration mode)

switchport trunk allowed vlan remove vlan-list

This command is used to manually remove VLAN's from a trunk link (i.e. VLAN's attached to a trunk port). The trunk link contains all the VLAN's that

conf t/ config t/configure terminal

Waits for entering into the configuration terminal : Switch (Config)#

interface range FastEthernet1/0/7-24 or gigabitethernet 1/0/1-2

Note: The 3750 switch has only two of Gigabit Ethernet ports and 24 Fast Ethernet ports. So we might be using mainly the Fast Ethernet ports

Waits for few seconds and the prompt appears as : switch(config-if)#

Switch(config-if)# **switchport mode trunk**

Switch(config-if)# **switchport trunk allowed vlan remove vlan-list**

<p>exist on a switch.</p> <p>After a timeout of 4000 secs a response beginning with "6509" is to be expected from the console.(the response is in the form of switch prompt)</p> <p><u>End</u></p> <p>This command is issued to return to the global config mode or the privileged EXEC mode.</p> <p>A response is to be expected from the console.</p>	<p>Waits for few seconds and the prompt appears : <u>Switch(config-if)#</u></p> <p>Exit/end</p> <p>Waits for few seconds and prompt appears as: <u>switch></u></p>
---	---

Unmapping The VLANS

Cisco6509UnMapVLANs.Xml: This script is used to unmap the VLANs.

<p><u>config t</u></p> <p>A command "conf t" was sent to the console</p> <p>A response beginning with "6509" is to be expected from the console (At CLI, the switch enters into the configuration mode)</p> <p><i>interface range port range</i></p>	<p>conf t/ config t/configure terminal</p> <p>Waits for entering into the configuration terminal : <u>Switch (Config)#</u></p> <p>interface range FastEthernet1/0/7-24 or</p>
--	--

This command allows us to specify the interface range. I.e. using this we can execute a command on multiple ports at the same time.

After a timeout of 2000 secs a response beginning with "6509" is to be expected from the console. (Switch enters into the interface configuration mode)

no switchport trunk allowed vlan mapping vlan-id translated-id

This command is used to remove the mappings between a pair of VLANS i.e. between Source VLAN and the Destination(translated) VLAN.

After a timeout of 4000 secs a response beginning with "6509" is to be expected from the console.(the response is in the form of switch prompt)

End

This command is issued to return to the global config mode or the privileged EXEC mode.

A response is to be expected from the console.

gigabitethernet 1/0/1-2

Note: The 3750 switch has only two of Gigabit Ethernet ports and 24 Fast Ethernet ports. So we might be using mainly the Fast Ethernet ports

Waits for few seconds and the prompt appears as : switch(config-if)#

Switch(config-if)# **no switchport vlan mapping *vlan-id translated-id***

Waits for few seconds and the prompt appears : **Switch(config-if)#**

Exit/end

Waits for few seconds and prompt appears as: **switch>**

Request for Un-mapping of VLANS

Cisco6509UnmapVLANRequest.Xml: This script is used to send a request for the mapping of VLANS.

This XML file mainly calls 3 scripts:

<script>Cisco6509Log_on</script>: This script is used to login in to the console. Please refer to the commands in the “log on” section above

<script>Cisco6509UnMapVLANs</script>: This script is used to unmap the VLANs. Please refer to the commands in the “Unmapping VLANs” section above

<script>Cisco6509Log_off</script>: This script is used to logoff from the console. Please refer to the commands in the “log off” section above