

# Gluware® Application Note

Introduction to Gluware Config Modeling

Getting Started Guide for Network Engineers and Operators

June 30, 2020

# Contents

Overview .....	1
Introduction to Gluware.....	2
Gluware Config Modeling.....	2
Onboarding Your Configuration Policy .....	5
Gluware Users .....	6
Pre-Requirements for Gluware Config Modeling.....	7
Steps To Use Config Modeling for Network Provisioning .....	8
Detailed Breakdown of Each Step Using Config Modeling.....	9
Automating Global Network Features .....	19
Automating DNS.....	19
Assembly.....	20
Feature.....	21
CLI Command Group .....	22
CLI Command List.....	22
Concept Item.....	25
Conclusion .....	26
Additional Resources .....	27

# Overview

This Gluware *Application Note* is designed to ramp up your understanding of the Gluware **Config Modeling** approach and building blocks. Enterprise networking professionals often face a lot of challenges. The business needs and demands on the network continue to grow with the move toward digitization. Network teams are often siloed and have limited human resources and legacy tools to manage multi-vendor networks. Studies show a high percentage of network changes are still performed manually via the command-line interface (CLI). These manual changes often result in misconfigurations and out of policy networks.

Network engineers often think the only path to network automation is through developing new programming skills and scripting. Gluware offers a solution to bring purpose-built applications that leverage an intelligent network orchestration engine to automate the network without programming. Gluware also empowers network engineers to leverage their current vendor specific CLI skill set and wrap reliable, scalable automation around it. Within a short amount of time users of Gluware can be performing inventory, config drift, config audit, OS upgrades and config management of global networks. Network engineering and operations will be empowered to keep up with the growing business demands while improving security and reliability.

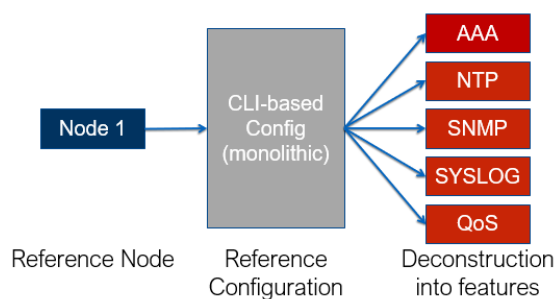
This *Application Note* was created using Gluware v3.7 and focuses on the basics of the Config Modeling application used for configuration management.

# Introduction to Gluware

Gluware is the leader in Intelligent Network Automation delivering an Intent-Based Networking (IBN) automation platform with the power to automate and orchestrate mission-critical networks at scale. With Gluware® Intelligent Network Automation, Network Operations can keep complex multi-vendor networks in policy while enabling new features that keep pace with changing business objectives. The Gluware automation and orchestration engine powers applications built to get you automating quickly. The application for policy-based configuration management is called Gluware Config Modeling.

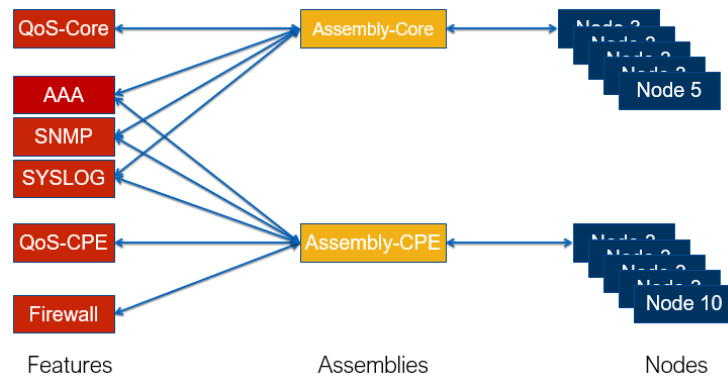
## Gluware Config Modeling

Gluware Config Modeling provides Intent-Based configuration management for Gluware Intelligent Network Automation. The Config Modeling application rapidly automates large-scale networks by leveraging pre-existing brownfield, multi-vendor configurations, bringing agility and cost savings to network operations. Config Modeling enables the rapid onboarding of customer provided CLI-based configured features. Then it turns features into network policy used to make only the changes needed declaratively for each network feature across the range of supported vendors and device types. Config Modeling is built on technology base packages (routing, switching, firewall, load-balancer, WAN optimizer and wireless LAN controller), vendor extensions and Config Modeling Kits (CMKs) that accelerate the on-boarding of your network's existing features without programming. The approach of Config Modeling is to deconstruct a monolithic CLI-based configuration into network features, which are each wrapped in data-modeling and treated as code instead of text within Gluware.



**Figure 1.** Example deconstructing a monolithic configuration into features

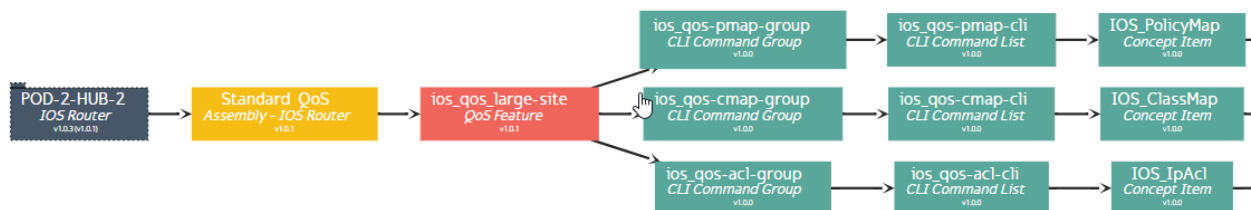
Those network features can then become the desired automation policy that Gluware will use to provision on other devices. Gluware can take those feature policies, group them using Assemblies, and assign them to Nodes. The Assembly can be thought of as a modular template made up of features to be automated on each set of Nodes. Config Modeling can automate just a few features on a set of Nodes, or it can automate the entire configuration.



**Figure 2.** Gluware groups Features using Assemblies, which are assigned to Nodes

The power of automation starts to be realized when one update to a SNMP feature can be provisioned to hundreds or thousands of Nodes. Since network features can be quite complex and powerful, the Gluware constructs were built to support them accordingly. In Gluware Config Modeling, a Feature is made up of:

- **Feature** - A high-level container including all the elements to define, read and write each feature you are automating.
- **CLI Command Group** - The CLI Command Group specifies all the CLI Command Lists.
- **CLI Command List** - The CLI Command List uses the native vendor CLI along with the ability to insert variables used by Gluware for specific parameters. Users can edit the CLI commands to match the desired policy.
- **Concept Item** - The Concept Item is the construct created to enable Gluware to read the currently configured state of the feature and capture any required variables. It is defined with vendor specific show commands and regex to extract the required information. Other advanced features, like Discovery Elements, can be defined in the Concept Item, which enable Gluware to read information from the node and assign it to a variable. Concept Items are created as part of the vendor Config Modeling Kits (CMKs) so there is a seeded library for each supported operating system. Users can create Concept Items using the Network Feature Design workflow. Gluware service delivery can also create custom Features and Concept Items for customers.



**Figure 3.** In this Node Instance Map from Gluware Config Modeling the QoS Feature is made up of Policy Maps, Class Maps and ACLs.

Within the Config Modeling app, the Gluware constructs are broken into four categories:

- **Nodes/Devices** - These are the actual network devices to be automated. Devices are usually onboarded in the **Device Manager** solution which synchronizes with the other solutions, including Config Modeling
- **Domains** – This construct defines items that span multiple devices, including Assemblies, IP address plans and more
- **Features** – A high-level container including all the elements to define, read and write each feature you are automating
- **Globals** - Lower-layer constructs that include specific CLI configurations to write and show commands with regex to read from network devices

## Onboarding Your Configuration Policy

To onboard your existing CLI based configuration there are three options:

- **Intelligent Model Discovery** - This Gluware workflow provides the ability to connect to a live device, read each of the configured features, then enable the creation of the required constructs in Config Modeling so that they can be used on any other device with the same operating system.
- **Network Feature Design** – Is a workflow option that steps a user through the process to define and create the configuration feature.
- **Example solution package** – One of the most common ways in which users get started with Config Modeling is to use example packages provided by Gluware that have commonly configured features (for each vendor operating system) and customize them to use your specific configuration.

## **Gluware Users**

Users of Gluware Config Modeling generally fall into two categories: Network Engineers/Architects and Network Operations.

Network Engineers/Architects typically use Gluware to onboard network feature policies and validate them in a lab or test devices before a broader rollout in production.

Network Operations typically use Gluware to deploy and maintain network features automated in the network.



# Pre-Requirements for Gluware Config Modeling

This *Application Note* will not cover the administrative tasks required when setting up a Gluware system. It assumes the following pre-requirements have been completed:

- Gluware software is installed and running
- Required packages including vendor Config Modeling Kits (CMKs) have been installed
- Licenses have been installed
- Organizations and users have been defined
- Network Devices/Nodes have been added in Device Manager
- Network features have been onboarded (addressed but not covered in this document)

For more information on these, see the *Gluware Installation Guide* and online Help in Gluware.

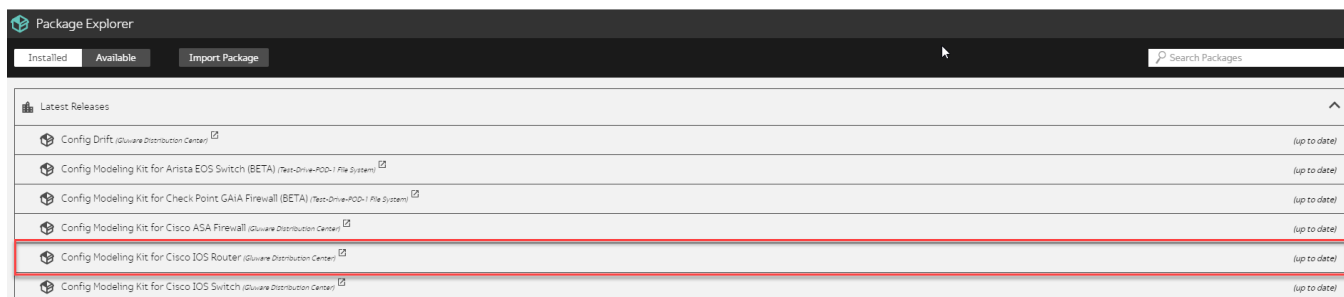
# Steps To Use Config Modeling for Network Provisioning

To use Gluware Config Modeling there are 10 steps (several are optional):

1. Ensure the pre-requirements (listed above) are complete
2. Select one or more Nodes to apply the same Assembly to
3. Associate the Assembly with each Node and validate the Features and Policies (instance data) associated with each one
4. (Optional, but recommended) Leverage the Node Instance Map to visualize the Assembly, Features and CLI constructs assigned
5. Validate the Assembly and each set of Features and Policies (instance data) associated with each one
6. Validate the constructs assigned to each Feature including the CLI Command Groups, CLI Command Lists and Concept Items.
7. (Optional, but recommended) Provision the Nodes using the Preview Feature
8. (Optional, but recommended) Review the detailed logs
9. (Optional, but recommended) Review the processed CLI that will be generated for each Node
10. Once satisfied with the validation and preview, execute the Action to start a Provision Features task and ensure all the Nodes result in the synchronized state

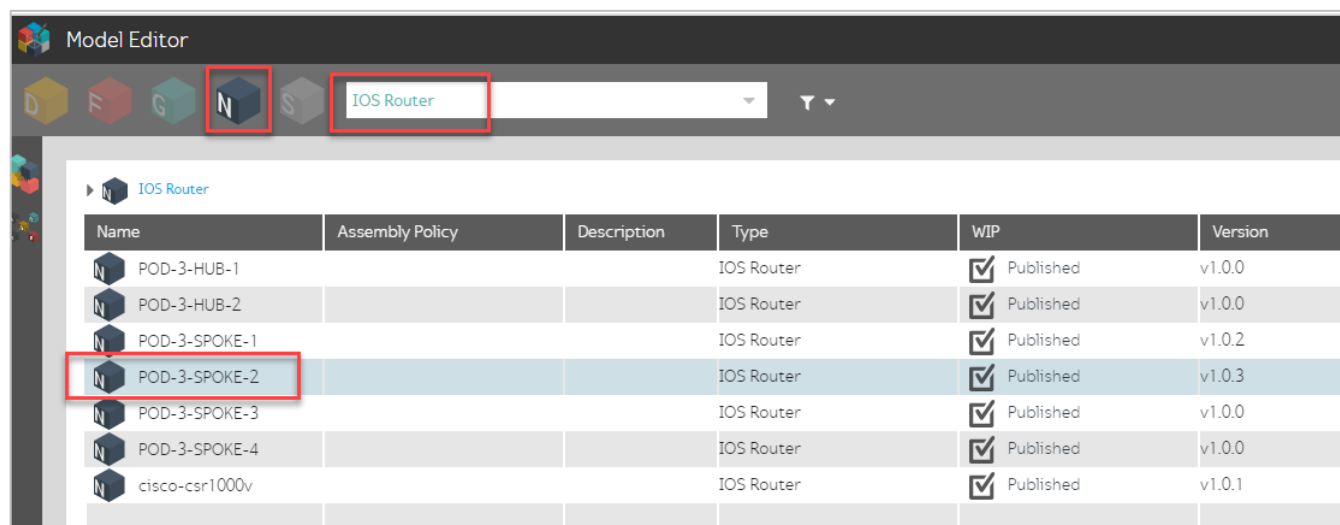
## Detailed Breakdown of Each Step Using Config Modeling

- 1) To support the required network vendor operating system, ensure the Gluware Config Modeling Kit (CMK) is installed and up to date. This example is using Cisco IOS for routers, so using the Gluware **Solutions Management** solution, validate the CMK is installed and up to date.



**Figure 4.** View of the Packages, including the Config Modeling Kits in Solutions Management

- 2) From the Gluware home page, navigate to the **Model Editor** solution. Click on the **N** in the blue block for the Node view, use the drop-down filter, select **IOS router**, then click on one of the Cisco IOS routers to begin configuration. In this example we will start with the device named POD-3-SPOKE-2.



**Figure 5.** Selecting the Node view and using the filter for IOS Router

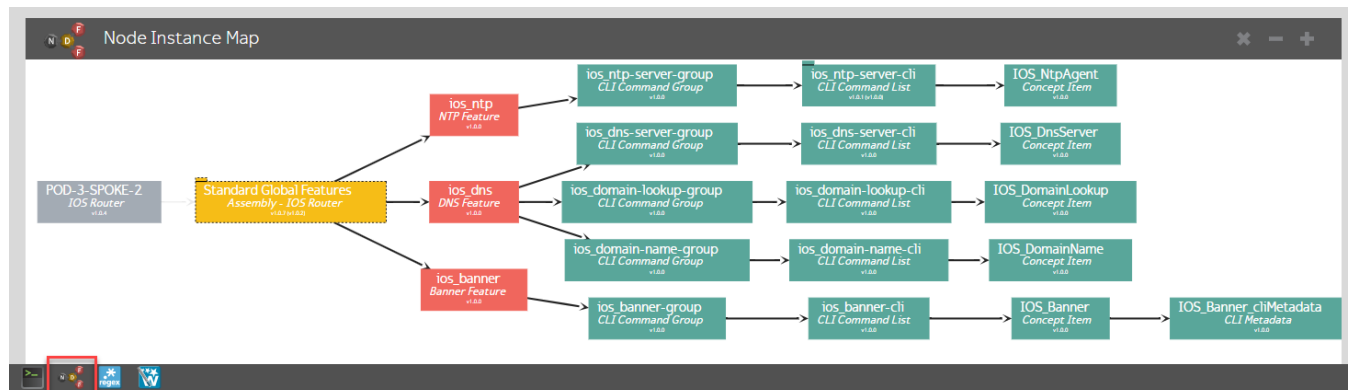
- 3) Assign an Assembly to the Node. A Gluware Assembly is basically a modular template that defines the Features you will be automating on the Node,

along with the feature policies (desired configurations). By clicking on the network Node, you should see the right-hand panel with the Node Instance details. Click the **Associations** tab and in the Assembly Policy dropdown, select the Assembly. In this example, we are using an Assembly called **Standard Global Features**. Click **Save** on the bottom of the form.

The screenshot shows a web application window titled "Instance". At the top, there are tabs for "\*Configuration" and "Attached Storage". Below this, a header bar displays "POD-3-SPOKE-2" with edit and info icons. A text field "Add Instance Description" is present. The main form area contains several sections: "Site Location (optional)" with a dropdown menu showing "No Instances Selected"; "Control Management Type" with a dropdown menu showing "managed"; a tabbed interface with "Connection", "Associations" (highlighted with a red box), "Persona", and "Hardware Specs" tabs; an "Assembly Policy" dropdown menu showing "Standard Global Features" (highlighted with a red box); a "Cloud Provider" dropdown menu showing "Local (not in a cloud)"; and a link "Add Node Specific Settings +". At the bottom, there are icons for "JSON" and "v1.0.3", a timestamp "Updated by mhaugh on 6/25/20 12:08 pm", and a "Save" button (highlighted with a red box).

**Figure 6.** In the Node Instance, use the Associations tab and select an Assembly Policy to assign to the Node.

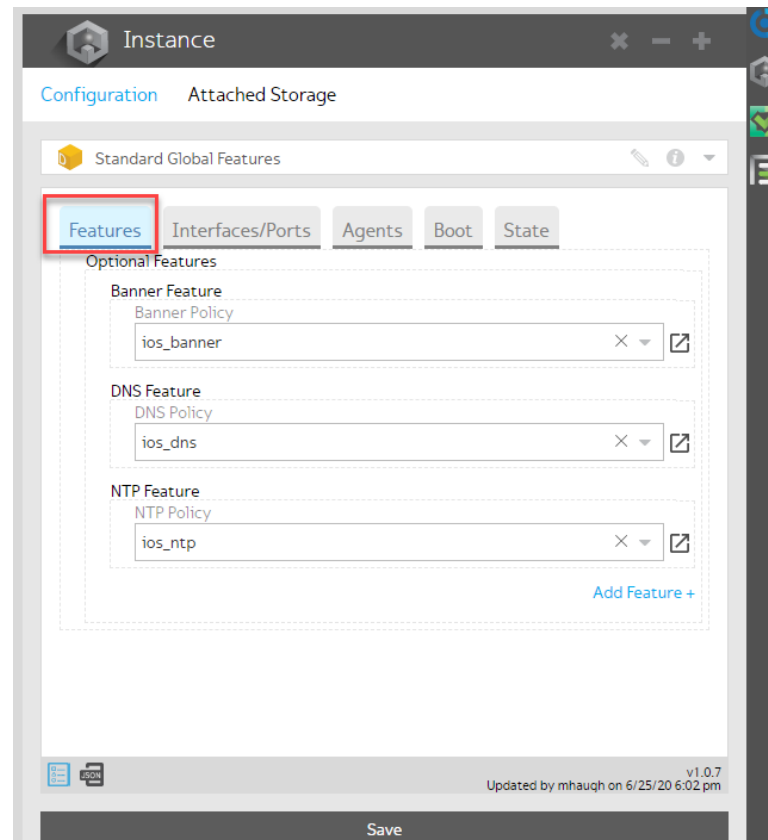
- 4) The **Node Instance Map** is a helpful way to visualize the Gluware constructs and access the instance data in each one. Ensure you are still in the Node view and have clicked on a Node. Use the Node Instance View icon on the bottom tool bar to bring up the view. Note that you can zoom in/out and move the view around, so it looks best.



**Figure 7.** Use the Node Instance Map view to see the Assembly and Features assigned to the Node

- 5) Using the **Node Instance Map**, click on the **Assembly**, which is the yellow box in the above example. This opens the instance details for the **Standard Global Features** assembly in the right-side panel.

The Features tab can be used to add or remove features as well as assign a policy to each feature (more will be covered on each feature). If you make changes, be sure to save. For this *Application Note* we do not need to make changes, but during testing and development of features you may want to work one at a time to isolate each feature.



**Figure 8.** In the Assembly Instance, use the Features tab to select the Features and Policies which will be assigned to all Nodes using this Assembly

- 6) Using the **Node Instance Map**, you can click on each Feature and navigate the tree of constructs including the CLI Command Group, CLI Command List and Concept Item. Clicking on each one will bring up the Instance data and options in each one. More on each of the features will be provided in the next section. Remember to click **Save** when making any changes.

The screenshot displays the Gluware Config Modeling interface. At the top left, a table lists CLI Command Lists. Below it, the Node Instance Map shows a hierarchical tree of constructs. A red box highlights the 'ios\_ntp-server-cli' node, with a red arrow pointing to its configuration panel on the right. The configuration panel shows the CLI commands for the 'ios\_ntp-server-cli' instance.

Name	Description	WIP	Version	Created By	
ios_logging-buffered-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_nac-global-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_nac-interface-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_netflow-exporter-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_netflow-interface-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_netflow-monitor-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_netflow-record-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh
ios_ntp-server-cli		<input checked="" type="checkbox"/>	Published	v1.0.0	mhaugh

Node Instance Map

```
graph LR
    subgraph Features
        F1[IOS NTP Feature]
        F2[IOS DNS Feature]
        F3[IOS Banner]
    end
    subgraph Groups
        G1[ios_ntp-server-group CLI Command Group]
        G2[ios_dns-server-group CLI Command Group]
        G3[ios_domain-lookup-group CLI Command Group]
        G4[ios_domain-name-group CLI Command Group]
    end
    subgraph Lists
        L1[ios_ntp-server-cli CLI Command List]
        L2[ios_dns-server-cli CLI Command List]
        L3[ios_domain-lookup-cli CLI Command List]
        L4[ios_domain-name-cli CLI Command List]
    end
    subgraph Concepts
        C1[IOS_NtpAgent Concept]
        C2[IOS_DnsS Concept]
        C3[IOS_Domain Concept]
    end
    F1 --> G1
    F1 --> L1
    F1 --> C1
    F2 --> G2
    F2 --> L2
    F2 --> C2
    F3 --> G3
    F3 --> L3
    F3 --> C3
    G1 --> L1
    G2 --> L2
    G3 --> L3
    G4 --> L4
    L1 --> C1
    L2 --> C2
    L3 --> C3
    L4 --> C3
```

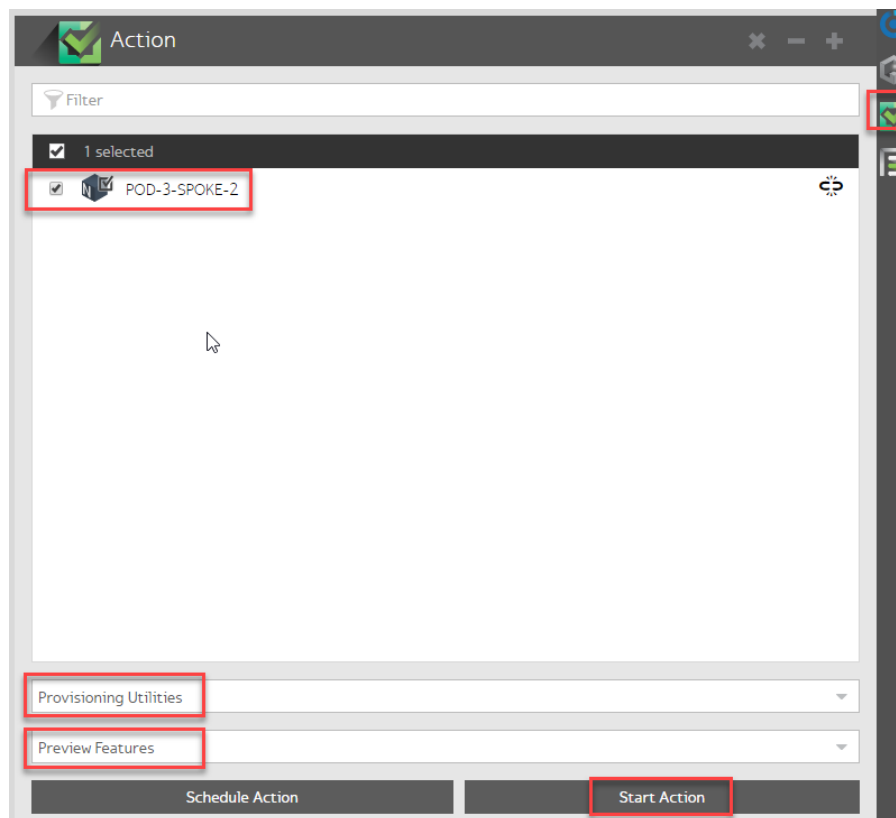
Instance Configuration for ios\_ntp-server-cli

```
1 clock timezone PST -7 0
2 clock summer-time PDT recurring
3
4 static ntp server
5
6 ntp server 10.12.3.11
7 ntp server 10.12.4.11
8
9 ntp server filled from a form
10
11 ntp server $context.ntp.server[*].ip
12
13 dynamic ntp server filled from the default-gateway on the running-config
14
15 ntp server $discover.NtpAgent.global.route.gateway1
16
17 ntp source $context.portMap.MGMT1.name
```

Save

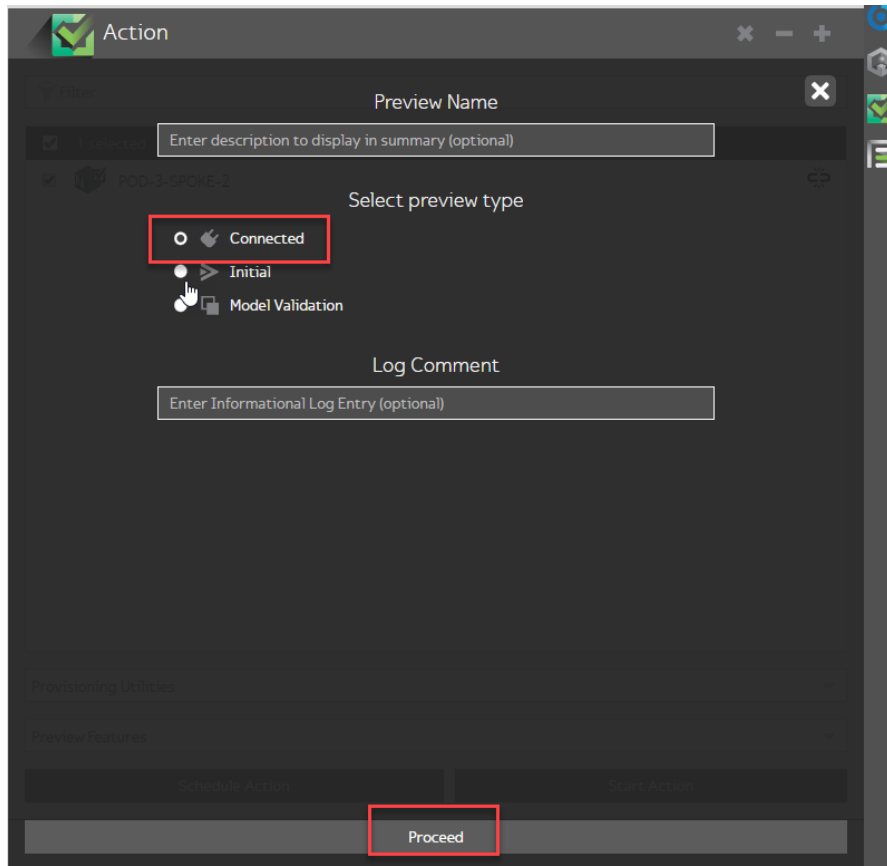
**Figure 9.** Use the Node Instance Map to view and validate the CLI constructs

7) Assuming you have validated your features, the next step is provisioning. Using the Node view, select one or more Nodes to provision. On the right slide menu bar, select the **Actions** view. Ensure that the desired Nodes are selected. In this example it is just the single POD-3-SPOKE-2 router. Using the dropdowns at the bottom, choose **Provisioning Utilities** and **Preview Features**, which is always recommended before writing changes to the devices. This generates the CLI commands needed by the Gluware orchestration engine depending on what is already configured on each Node. Click **Start Action** to begin the preview, select the **Connected** preview to enable Gluware to connect to the Node and compare its current configuration to the specified policy for each feature.



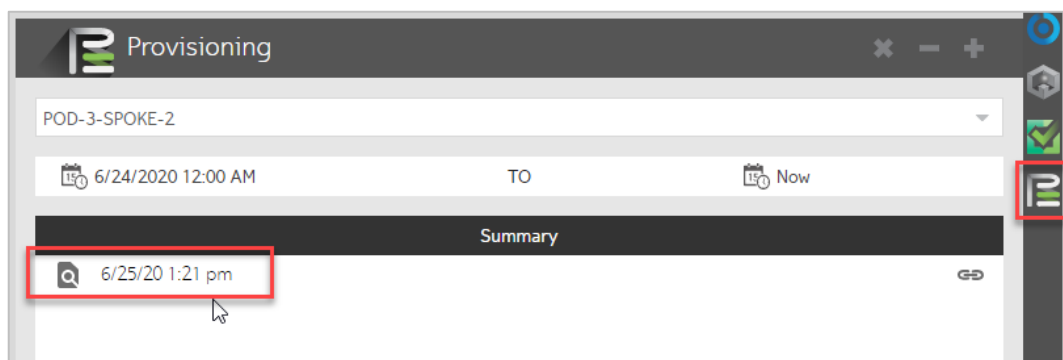
**Figure 10.** In the Actions view, select the Node(s) and Preview Features as the Provisioning Utility and Start Action





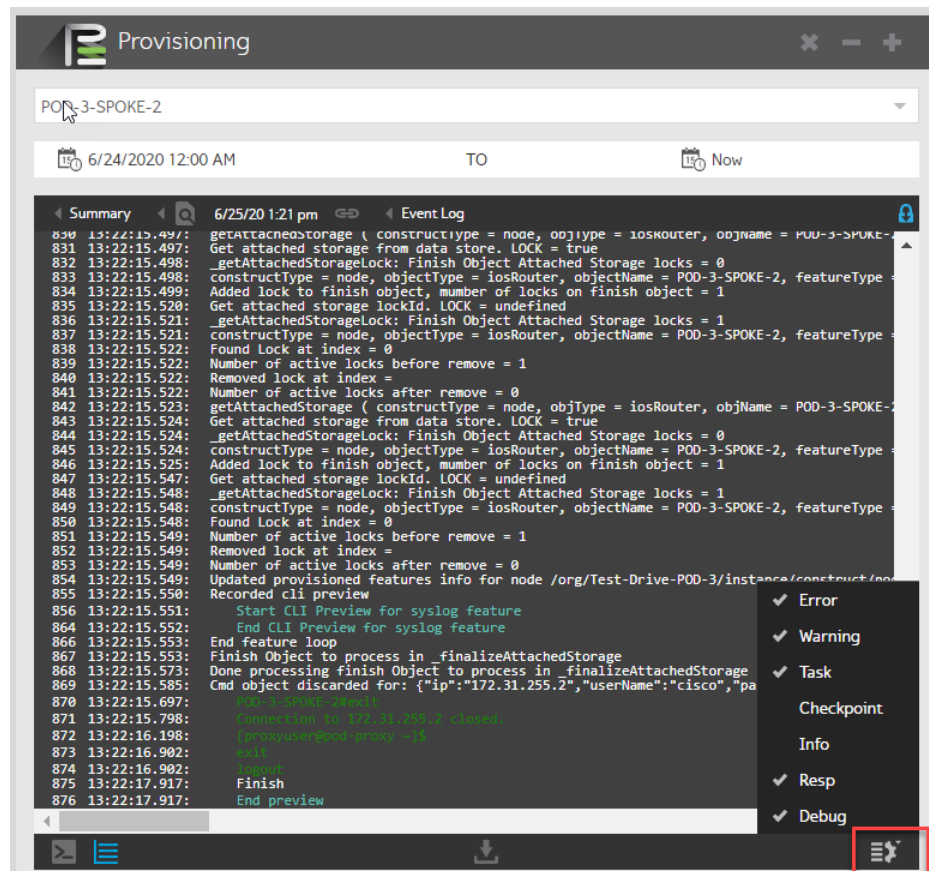
**Figure 11.** Select the Connected option for the Preview so Gluware can connect to the Node(s) and compare the current config state to the desired state for each feature

- 8) Select the **Provisioning** view using the icon on the right-side bar and double-click on the summary log to see the details.

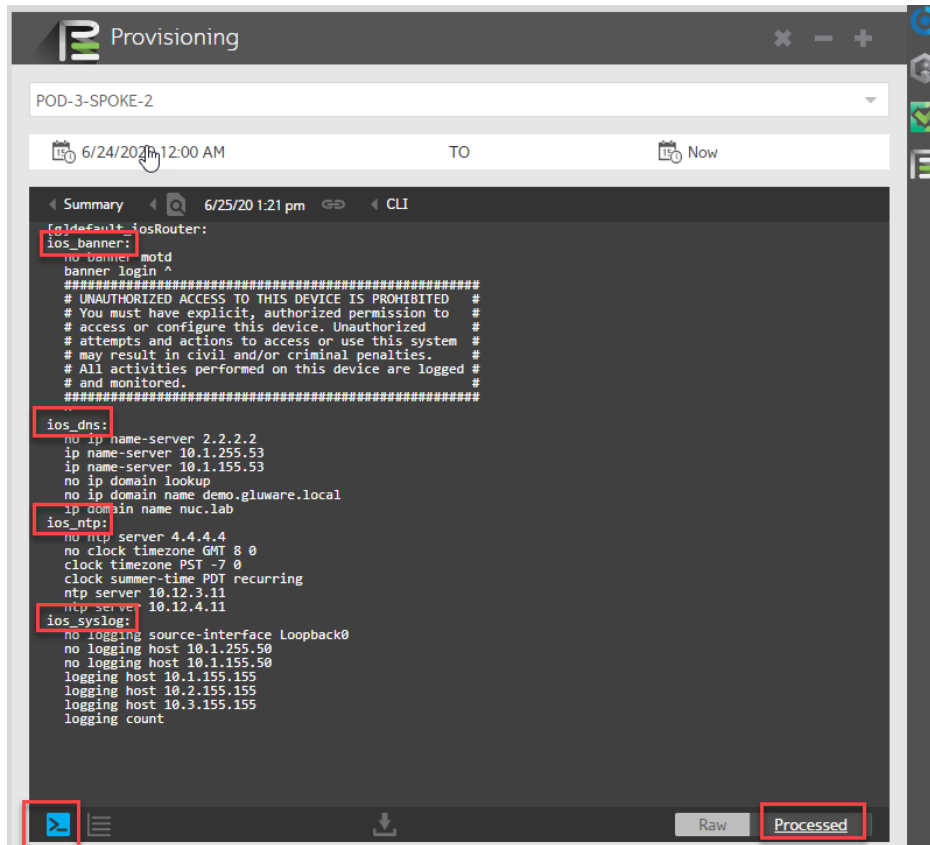


**Figure 12.** Use the Provisioning view to see the log and double-click to open it

- 9) Scroll down the detailed log to see all the interactions Gluware is having with the Node. The logging level is customizable via the cog on the lower right. For the CLI view, click on the >\_ icon and then select the **Processed** view to see exactly what CLI commands are needed for Gluware to configure the Node in the desired state for each feature.

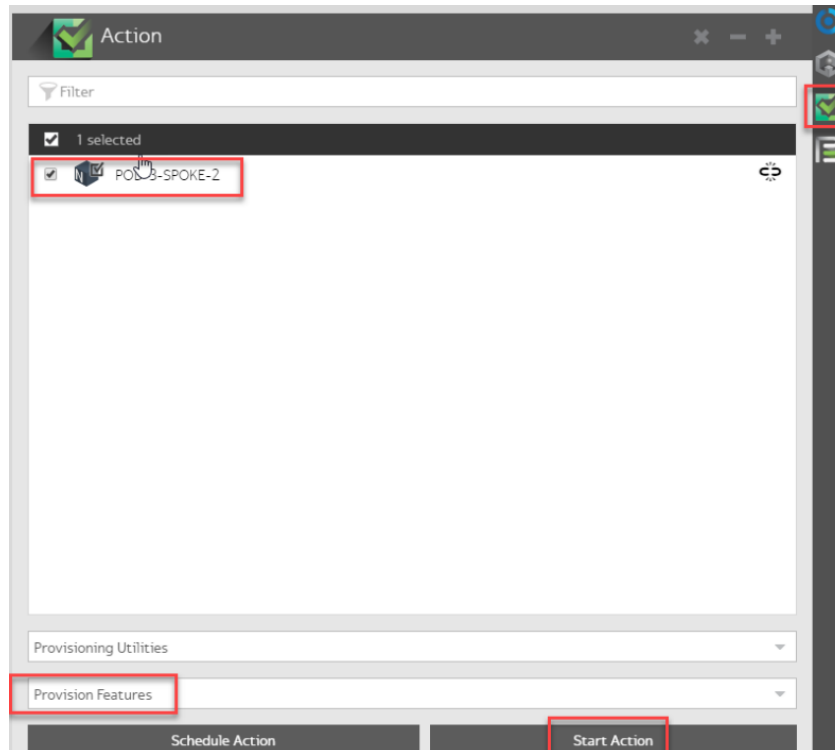


**Figure 13.** Browse the detailed log and adjust the logging level as needed



**Figure 14.** Use the CLI view and click Processed to see the CLI that has been rendered based on what is needed to provision the Node

- 10) Once satisfied with the preview and ready to provision, return to the **Actions** view, ensure you have the Node selected, and change from **Preview Features** to **Provision Features**. Click **Start Action**. Again, you can use the Provisioning view to see all the details.



**Figure 15.** In the Actions view, select the Node and the Provision Features option to implement the changes on the Node(s)

In the Node Grid there is a summary status of the progression of each feature until it reaches synchronization. All Nodes should result in the **Provisioned** state when complete.

The screenshot shows the 'Model Editor' window with the 'Node View' table. The table has columns: Hostname, Referenced Instances, Feature Config Status, Feature Config Summary, Feature Provisioning Stat..., Features Provisioning, Features In Error, and Feature Provisioning Sum. The row for 'POD-3-SPOKE-2' is highlighted with a red box.

Hostname	Referenced Instances	Feature Config Status	Feature Config Summary	Feature Provisioning Stat...	Features Provisioning	Features In Error	Feature Provisioning Sum
POD-3-SPOKE-1	(0) (1)	✓ (1)	✓ Published	⚙ (1)			⚙ Not provisioned
POD-3-HUB-1	(0) (1)	✓ (1)	✓ Published	⚙ (1)			⚙ Provisioned
POD-3-SPOKE-2	(1) (5)	✖ (1) ✓ (4)	✓ Published	⚙ (3) ⚙ (1) ... ios_banner			⚙ Provisioning
POD-3-SPOKE-4	(0) (1)	✓ (1)	✓ Published	⚙ (1)			⚙ Not provisioned
POD-3-SPOKE-3	(0) (1)	✓ (1)	✓ Published	⚙ (1)			⚙ Not provisioned

**Figure 16.** Scroll to the right in the Node View to see a summary status of the provisioning

# Automating Global Network Features

*Globals* is a generic term for common features configured on network devices that handle standard services. They are configured at the global level of a configuration, not under a protocol or interface. This example will focus on a commonly configured global parameter - domain-name service (DNS).

## Automating DNS

The Domain Name System (DNS) is a distributed database in which you can map hostnames to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated hostname. For this example, we will configure three parameters related to DNS.

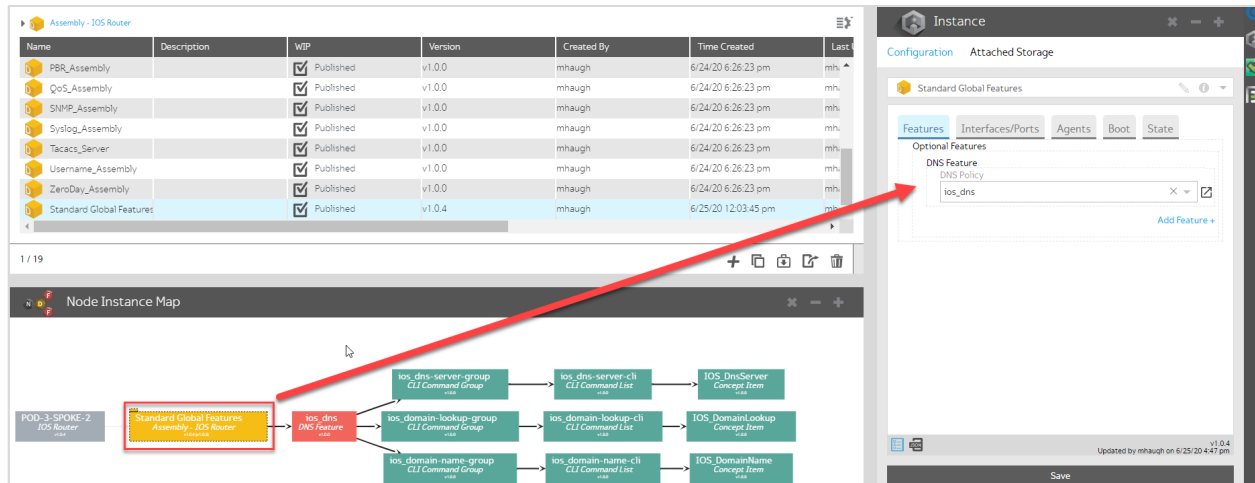
**DNS Server** – A name server is used to keep track of information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IP addresses. There are public DNS services for registered (public) IP addresses, like Google Public DNS (IPv4) available at 8.8.8.8 and 8.8.4.4. Organizations also often have private name services for domain names and IP addresses in the private IP space. This example is using IP address in the private IP range.

**Domain Lookup** – By default, any single word entered on an IOS device that is not recognized as a valid command is treated as a hostname with which you want to use telnet. The device will try to translate that word to an IP address in a process that can last about a minute. To eliminate that issue, the command **no ip domain-lookup** can be used to turn that off.

**Domain Name** - Defines a default domain name that the Cisco IOS software will use to complete unqualified hostnames.

## Assembly

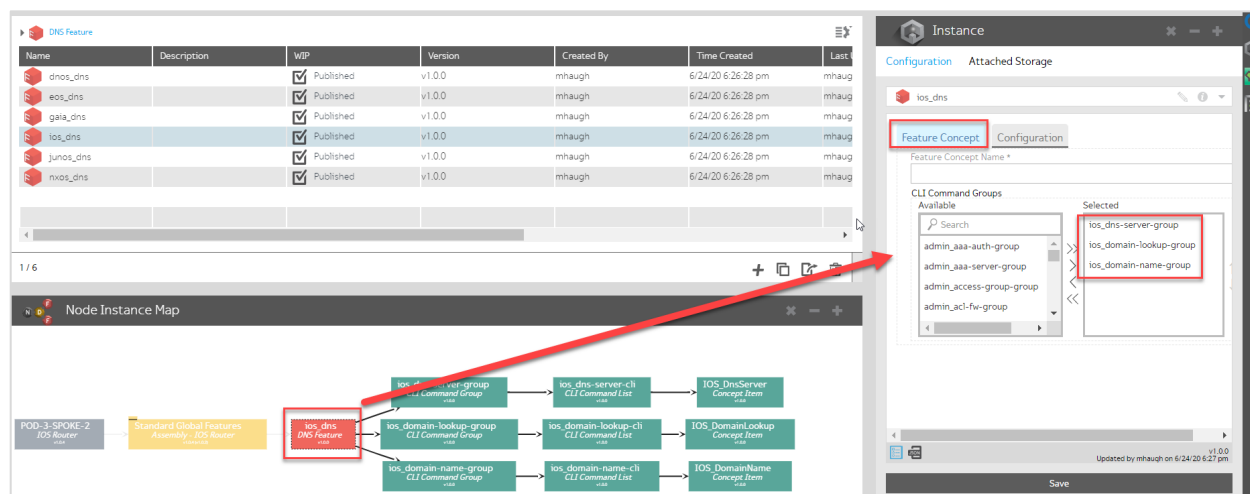
To simplify this example the Assembly will only include a single feature. Features and their policies can be added and removed as part of the modular template to specify exactly what Gluware is automating on all the Nodes that use that Assembly.



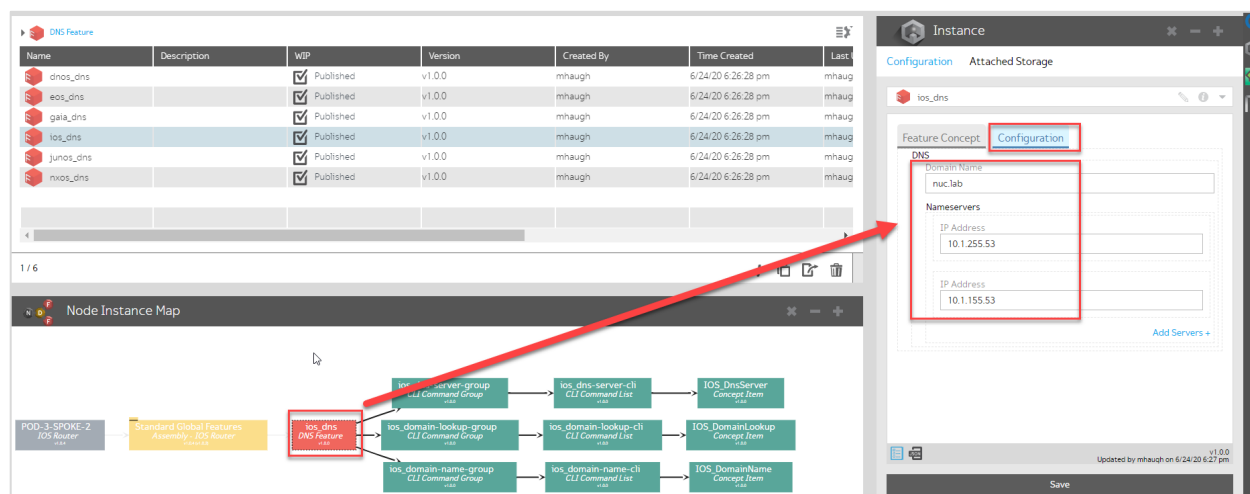
**Figure 17.** Use the Node Instance Map to view the Assembly Instance

## Feature

The Gluware Feature is used to associate all the CLI Command Groups required. The Feature also has a Configurations tab which is used to capture user input as variables. In this case, the DNS Feature is made up of three CLI Command Groups for DNS server, domain lookup, and domain name. The Configuration tab is also used in this example to capture user input for variables including the domain name and the IP addresses of the name servers.



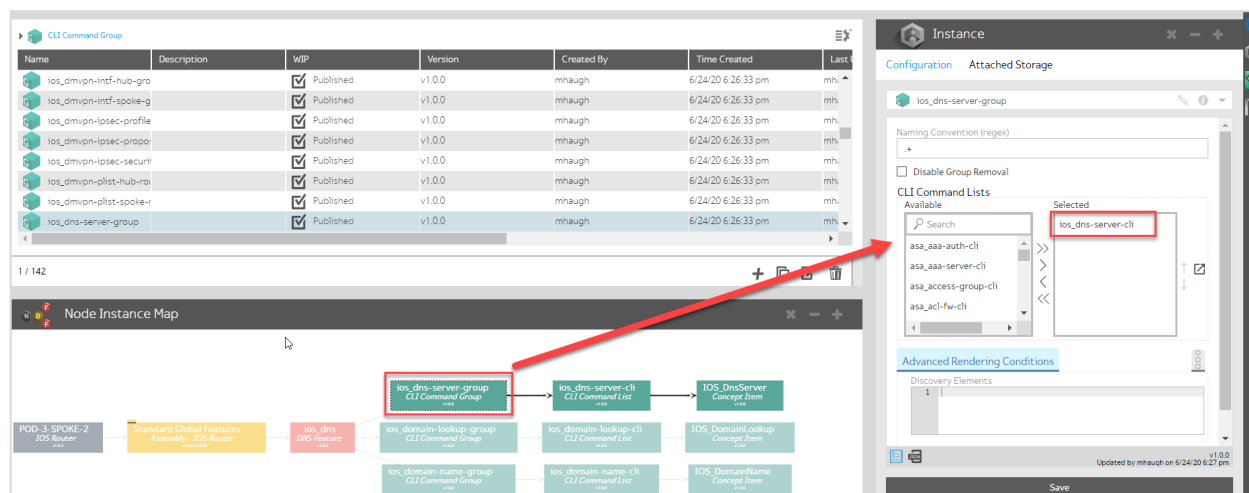
**Figure 18.** Use the Node Instance Map to view the Feature Instance



**Figure 19.** Use the Node Instance Map to view the Configuration parameters in the Feature Instance

## CLI Command Group

The CLI Command Group is used to specify all the CLI Command Lists. Again, in this case DNS is a simple feature consisting of a single CLI Command List. Advanced Rendering Conditions are not required for DNS. Since each of the CLI Command Groups each have a single CLI Command List, only one of the three is shown as an example below.

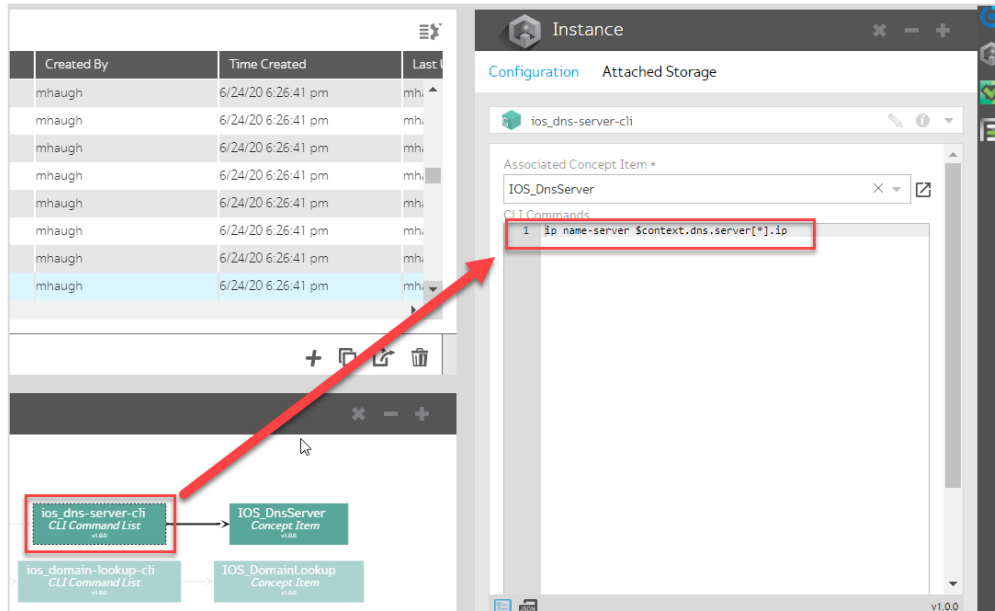


**Figure 20.** Use the Node Instance Map to view the CLI Command Group

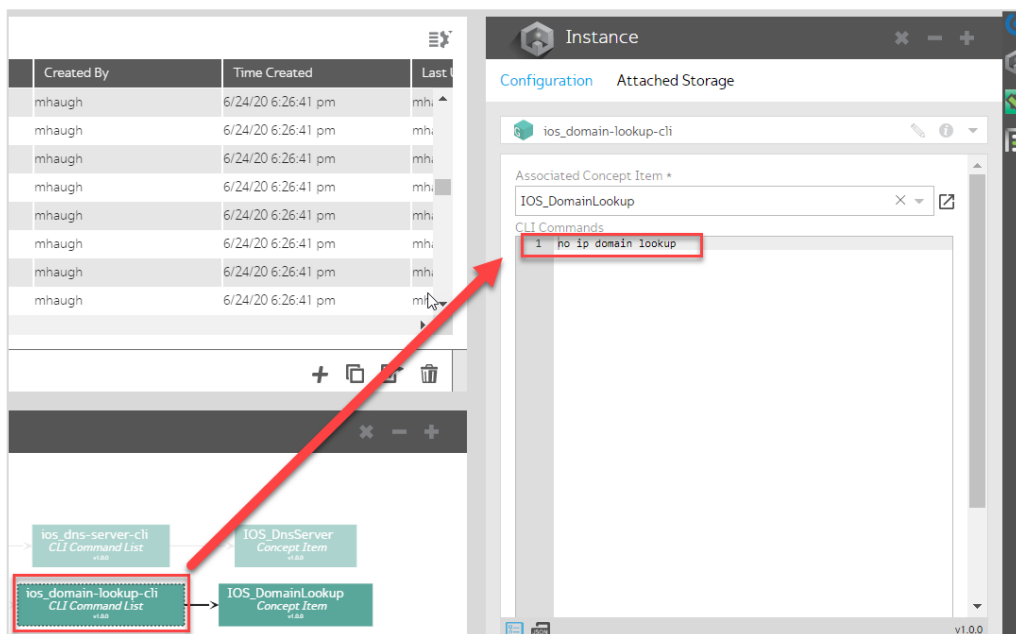
## CLI Command List

The CLI Command List gets a little more interesting with the DNS features since two of the three are using Gluware variables. Any time you see a **\$context** it is referring to a variable that is stored locally in the Feature data-model shown in the DNS Feature Configurations tab. For name servers, the variable specifies an array so the CLI line will get rendered for each of the server IP addresses provided in the list. In this string, the bolded part indicates an array: `ip name-server $context.dns.server[*].ip`. The domain lookup Feature is using static CLI. The domain name Feature is using the variable, also pulled in from the DNS Feature Configurations tab using the string `ip domain name $context.dns.domainName`.

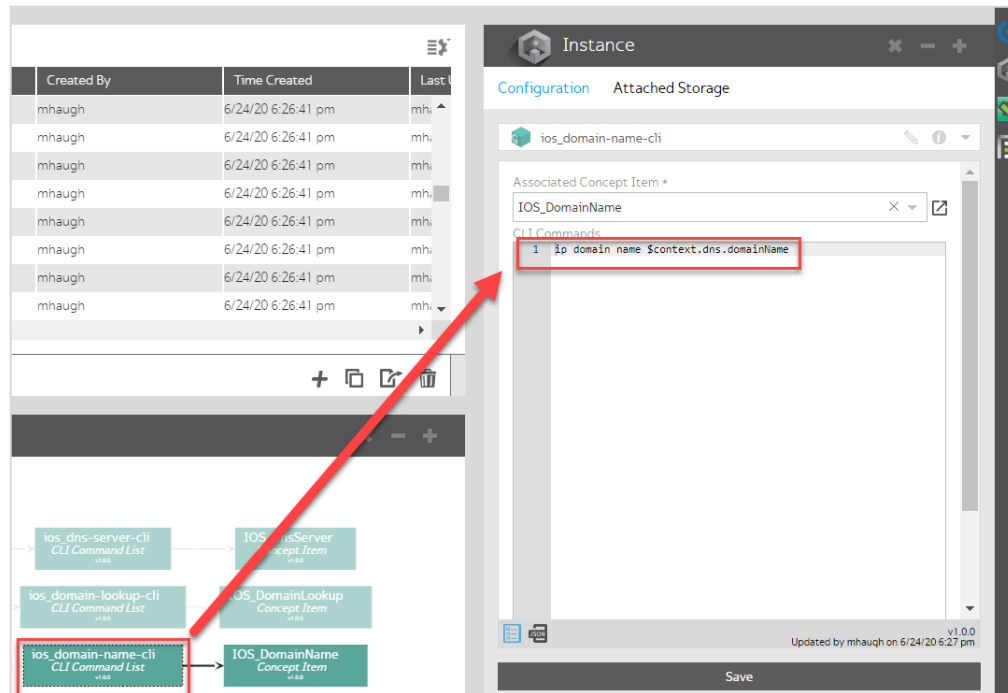




**Figure 21.** Use the Node Instance Map to view the CLI Command List for the DNS Name Server



**Figure 22.** Use the Node Instance Map to view the CLI Command List for the Domain Lookup feature



**Figure 23.** Use the Node Instance Map to view the CLI Command List for the Domain Name feature

## Concept Item

The Concept Item is the construct created to enable Gluware to read the currently configured state of the feature and capture any required variables. It is defined with vendor specific show commands and regex to extract the required information. Other advanced features, like Discovery Elements, can be defined in the Concept Item, but are not required for the DNS feature. The DNS Concept Items are part of the Cisco IOS CMK provided by Gluware. Since these are simple examples, below is one example of the Concept Items for DNS.

The screenshot displays the Gluware interface. On the left, the 'Node Instance Map' shows a list of Concept Items. The 'IOS\_DnsServer Concept Item' is highlighted with a red box, and a red arrow points from it to the configuration panel on the right.

Created By	Time Created	Last Modified
mhaugh	6/24/20 6:17:03 pm	mhaugh
mhaugh	6/24/20 6:17:03 pm	mhaugh
mhaugh	6/24/20 6:17:03 pm	mhaugh
mhaugh	6/24/20 6:17:03 pm	mhaugh
mhaugh	6/24/20 6:17:03 pm	mhaugh
mhaugh	6/24/20 6:17:04 pm	mhaugh
mhaugh	6/24/20 6:17:04 pm	mhaugh
mhaugh	6/24/20 6:17:04 pm	mhaugh

The configuration panel for 'IOS\_DnsServer' includes the following fields:

- Concept Item Name: DnsServer
- Operating Systems: Cisco Systems (Operating System: IOS/IOS XE)
- Show Commands: sh run | sec ^ip name-server
- Operating Mode: Basic
- Key CLI Line: Key CLI Line
- Flat Structure CLI?: ☒

The bottom of the panel shows a 'Save' button and a status bar indicating 'Updated by mhaugh on 6/24/20 6:17 pm'.

**Figure 24.** Use the Node Instance Map to view the Concept Item for DNS Server

To preview and provision the DNS feature on the network Nodes, follow steps 7 – 10 in the **Steps To Use Config Modeling for Network Provisioning** section above.

## Conclusion

This *Application Note* was intended to provide a base knowledge for users of Gluware Config Modeling to understand the concepts and constructs used. The DNS global feature example is a straightforward example that uses static CLI along with features that leverage variables. As the configuration gets more complex, the power of using Gluware will become even more apparent.

Spending some time to model network features using Gluware can help to standardize configurations across your network devices, eliminate manual errors and configuration bloat.

If you do not have a Gluware system available for testing and implementation you have three options:

- 1) For individuals serious about getting Gluware in production we offer a (limited time) free 30-day pilot to production trial. Learn more here: <https://gluware.com/business-continuity-offer/>
- 2) For individuals that want to “kick the tires” and try it themselves but don’t have resources for a trial, request a Test Drive where we spin up the environment for you. Learn more here: <https://gluware.com/test-drive-request/>
- 3) For those that want to meet with the Gluware team, ask questions and get a more tailored demo, request a demo here: <https://gluware.com/request-demo/>

## Additional Resources

Solution briefs, research papers, eBooks and more:

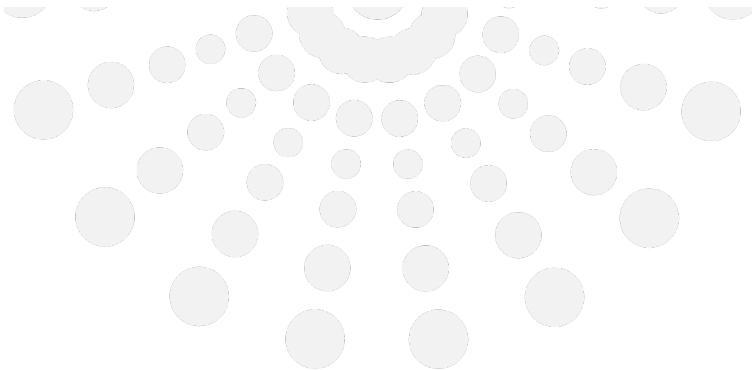
<https://gluware.com/resources/collateral/>

Videos including demonstrations and tutorials:

<https://gluware.com/resources/videos/>

Webinars: <https://gluware.com/resources/webinars/>

Blogs: <https://gluware.com/resources/blog/>



2020 L Street, Suite 130  
Sacramento, CA 95811

[www.gluware.com](http://www.gluware.com)

© 2020 Gluware, Inc. All rights reserved.