

User Manual

Overview

The program that constructs networks, extrapolates configurations from a user defined template. In this template you will need to define which interfaces are connected to each device and specify the routing protocols, advertisements and a variety of other components. Each template will specify a particular sub-topology that then gets replicated across sets of devices. The tools also allow you to specify which kits, racks and rooms you want to include in replication and facilitates the deployment of multiple networks utilising VRF.

Designing Networks with Templates and Models

Templates are made up of user defined settings that are abstracted from the particular devices, interface names and connections. When wanting to deploy a new network there is a 4 step process. However as you gain familiarity with the system, many templates and settings will be able to be reused each time. As a brief overview there are 3 template files that are made for each network deployment. The first is the physical template which describes each device, the interface names and ‘direction’. This template can be often resued if not changing the physical topology of the network. The config template is where you define the network settings, protocols and services of the network within a sub-topology. The replication template specifies what devices you want to push configs two in each room, rack and kit, and also if you want ot deploy multiple networks through VRFs, this template can be easily modified to deploy to more or less kits as required.

Errors and Invalid configs

It can be easy to accidentally configure conflicting or incorrect configurations, these may include configuring the same interface twice or referencing settings or protocols that haven’t been defined. In these cases the appropiate setting will be highlighted red and when hovered over a description of the issue will be shown.

Step 0

Before starting you will have the complete network diagram that you want to deploy. In order to use this system you will need to look at the interfaces and routing protocols being deployed and identify a sub-topology that is restricted to a single kit. The sub-topology will need to include all the unique configurations that may be deployed to the devices.

If there are configurations that exist less frequently then every device in each kit, rack and room. Do note that some configuration (except ip addresses and routing advertisements) can be excluded based on kit, rack and room.

Using this sub-topology you will need to identify ‘interface-directions’ These are used to abstract the interface names. A direction will be a number that groups interfaces together based on their context. That is they contain the same logical configuration. When labelling interfaces, connected interfaces cannot have the same number (having the same number is like saying it connects to itself).

NOTE: Good practice is to have connected interfaces be adjacent numbers (1 connects to 2 (1 is the opposite direction of two)). **NOTE:** The Reasoning of why connected interfaces shouldn’t be labelled the same can be found in the System Documentation (5.1).

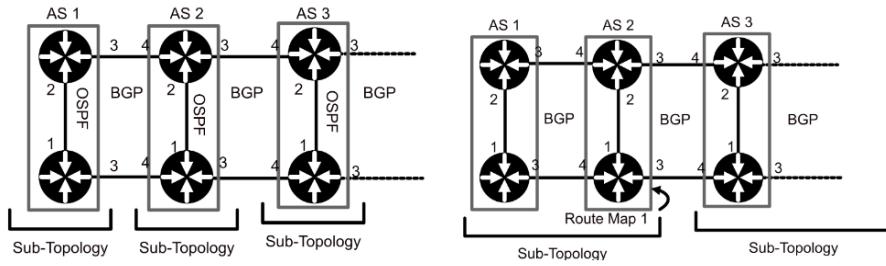


Figure 1: Topology

Step 2 Define Physical Topology

Using your identified sub-topology and interface directions, you will need to define a physical template so the system can produce correctly labelled configurations. A Sample Configuration can be found below.

In the **Make Wiring Scheme** page you can select the physical template you wish to edit or create a new one.

The first step is to add each device that is present in the sub-topology. When naming the device ensure the name matches the name that is given to the device according to the ATC Servers. That is Router or Switch with the first letter capitalised, followed by a space and its number.

For each device you will need to create each interface that is used in your topology (including the loopback interfaces). This will require you to enter the ‘direction’ and the interface name as according to the physical device.

NOTE: INTERFACE NAMING: interface names can be shorthand e.g. s0/1/0, partial e.g. ser0/1/0 or full Serial0/1/0, The supported interface types are serial, fast-ethernet, and gigabit-ethernet. So the name should comply such that if it were typed into the interface on the cisco ios command line, the interface name would be recognised. Please see Reasoning - Interface Naming in the System Documentation.

NOTE: you can specify subinterfaces such as s0/1/0.1 and s0/1/0.2 if you want multiple logical links on the same physical one, however if this is done, you can't have the parent interface s0/1/0 present in the physical topology.

If the interface is connected to another device click the arrow button to create a new connection. Enter the Device Name and the Direction of the interface that it is connected to. Once you click away a new connection is made on both devices. You can click the arrow button again to disconnect it, which will remove it from both devices.

By default the Internal check will be selected. However if the link is connecting one sub-topology to another you will need to change this to either next or previous. If the connection is intended to lead to the next replication of a sub-topology choose next. If the connection is intended to come from the previous sub-topology choose the previous. For more details see the Replication Step.

NOTE: Loopbacks do not need to be connected to themselves.

NOTE: Creating a new connection to a device that does not exist will be refused.

NOTE: Creating a connection to an interface that is either already in a connection or pending a connection (connect button pressed but no details entered) will be refused. You will need to disconnect the pending connection and retry.

Step 3 Create Configuration Template

This step is where the bulk of the settings are made. Here you will define addressing, routing and a variety of other configurations. Under Construct Configurations you will be able to select the template you want to edit as well as the wiring context which is just the physical template that you had just created.

Part 1 Define Models, Mappings, Networks and Interfaces

Models A Model is just an abstract representation of each device in the template. Each Device in the sub-topology will require its own Model. So the first thing to do is add each Model in the Models Box giving it a unique name. Once you have created the Models move back to the Mappings Box. Here you can tell the template what physical device name is represented by the Models you have created. Thus for each physical device in a kit you will need to tell the template what model should be applied to it.

Interfaces Now you can start creating your interfaces for the template, this is similar to the physical template, for each interface you will set it to a direction. The next drop down is for the network that you want the interface to be a part of. This is useful if you want to specify a simulated /24 subnet with loopbacks,

[Home](#) |
 [Update/Check Wiring Scheme](#) |
 [Construct Configuration](#) |
 [Create Replication Scheme](#) |
 [Upload Configuration](#) |
 [Physical Topology\physical.ini](#)

Name:	physical	Save					
Router 1 -							
Interfaces (direction name): [+/-]							
1	s0/1/0	->	Router 2	2	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
3	s0/1/1	->	Router 3	4	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
4	g0/0/0	->	Router 3	3	Internal	<input type="radio"/> To Next Kit	<input checked="" type="radio"/> To Previous Kit
5	lo0	->					
6	lo1	->					
7	l10	->					
Router 2 -							
Interfaces (direction name): [+/-]							
2	s0/1/0	->	Router 1	1	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
3	g0/0/1	->	Router 4	4	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
4	g0/0/0	->	Router 4	3	Internal	<input type="radio"/> To Next Kit	<input checked="" type="radio"/> To Previous Kit
5	lo0	->					
6	lo1	->					
7	l10	->					
Router 3 -							
Interfaces (direction name): [+/-]							
1	s0/2/0	->	Router 4	2	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
3	g0/0/0	->	Router 1	4	Internal	<input type="radio"/> To Next Kit	<input checked="" type="radio"/> To Previous Kit
4	s0/1/0	->	Router 1	3	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
5	lo0	->					
6	lo1	->					
7	l10	->					
Router 4 -							
Interfaces (direction name): [+/-]							
2	s0/1/0	->	Router 3	1	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
3	g0/0/0	->	Router 2	4	Internal	<input type="radio"/> To Next Kit	<input checked="" type="radio"/> To Previous Kit
4	g0/0/1	->	Router 2	3	Internal	<input checked="" type="radio"/> To Next Kit	<input type="radio"/> To Previous Kit
5	lo0	->					
6	lo1	->					
7	l10	->					

Figure 2: Physical Template

or ‘internal’ and ‘external’ links to have a different network summary route. You can specify networks in the Network Box, then in the interface dropdown you can set it to the index of the network. The last box allows you to specify the subnet mask in CIDR notation.

Connections Similarly to the physical template you can create connections to other models. Again you will want to specify whether the connection is internal, to the next replication or to the previous replication. An additional input labelled topologies back is present. Most of the time it will be one. However you can increase this value if you aim for the connection to be connected to a sub-topology that is before the most recent previous replication. See below for an example.

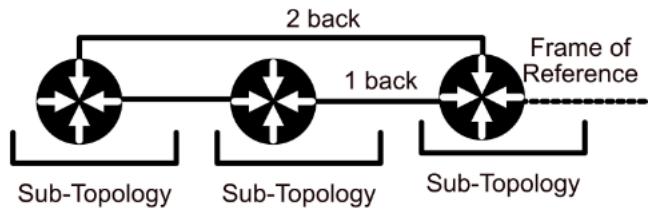


Figure 3: Connections Across Sub-topologies

NOTE: As there are four routers in a kit each with a unique name you will need to specify the four mappings. Even if your sub-topology has 2 models, if it is applied to each pair of devices in a kit you will need to specify mappings for all 4 devices.

Part 2 Create Routing Protocols

Next you will want to configure the routing protocols for the sub-topology. In the routing box add a new routing, and give it a unique name. This name is used to refer to the specific instance of the routing protocol (thus you can have multiple instances of OSPF running). In the drop down select you can decide whether you want BGP or OSPF as the routing protocol for this instance. Now in the Models box under Routing add the name for each model that the routing instance should be applied to.

NOTE: Make sure that you add the name of the Routing Instance to the list of Routing under each Model.

BGP To configure BGP you can set the router-id. Each router running BGP will increment the router-id inclusive of this first value. Similarly you can set the first AS number which will be incremented according to the instructions in the Increment AS After box. This input will usually contain the last model if you want each sub-topology to be a single autonomous system. However you can enter

multiple models separated by commas if you want multiple autonomous systems in each sub-topology. The update source can be left blank but if you want to set it you can enter the interface-direction for the interface that should be the update source. Finally you can add advertisements set to the interface-direction of each interface.

NOTE: BGP Neighbours are automatically made between Models that are directly connected and both the same instance of BGP.

OSPF To configure OSPF you can set the router-id. Each router running OSPF will increment the router-id inclusive of this first value. The process ID number needs to be set, and you can add advertisements set to the interface-direction of each interface.

NOTE: Only single area OSPF is supported.

Part 3 Creating Components

Components are created in a similar way to the routing protocols. You give each instance of a component a unique name, set it the configuration in the select and then make sure to add the component under the components heading of each Model you want it to apply to. Components also allow the ability to be restricted based on Room, Rack or Kit. Thus you can have configuration on every Model 2 in ATC328, in the yellow kit. For each of these inputs you can specify what containers the component Should be applied to. All of them “All”, A single colour “Black” or, multiple colours separated by commas “Green,Blue”.

NOTE: Make sure that you add the name of the Component Instance to the list of Components under each Model.

SYSLOG Here you can specify the Syslog Server Address, The Facility, the Log Level, The Source Interface for messages. If you want to specify an alternate protocol and port, you can set those to tcp/udp/default, and the port number under port. If the default setting is supplied the port input will be ignored.

SNMP Here you can specify the server address, the security names for snmp version 3 and whether traps should be enable. The location of snmp is automatically determined based of the room, rack, kit , and router name. The options should contain of new line spaced options of what should be included or excluded in snmp.

ACL The configured ACL will always be a named ACL; there is no support for numbered ACLs. You can specify whether the ACL is extended or not. In the rules you can paste all the specified rules for the ACL. ACLs can be attached to interfaces in or out, `interface_direction-in/out`.

Prefix List Prefix Lists are configured similarly to ACL's specifying name and rules.

Route Map With Route Maps you specify the name and Statements similarly to ACLs and Prefix lists. You can also specify whether the route map is to permit or deny, and the preference number. Under Attach To you can list the interface-directions and in out direction `interface_direction-in/out`. This will not apply the routemap on the interface instead it looks at the connection to determine what bgp neighbor would be present and will determine the update source if necessary and apply the routemap to the neighbor statement.

NOTE: For ACL, Prefix lists and Routemaps there are some keywords that can be used if you want the rules to be based on the context of the replication.

{interface-id-interface_direction} - gets the name of the interface on the current model

{interface-network-model_name-interface_direction} - This will get the network address of the interface, based on its address and subnet mask.

{interface-mask-model_name-interface_direction} - this will get the subnet mask in decimal dotted notation of the interface

{interface-slash-model_name-interface_direciton} - this will get the subnet mask in CIDR notation of the interface

{interface-wildcard-model_name-interface_direction} - this will get the wildcard based on the subnet mask of the interface.

{interface-address-model_name-interface_direction} - this will get the interface ip address

{network-index} - this will get the current network address of the index specified as seen in the network box, for the current replication (useful for getting summary routes). (You can determine the subnet mask as it will be the same in each replication).

{list-name} - this will insert the name of the prefix/acl or routemap specified into the list and will make sure it is valid.

{bgp-as-local+{num}|-{num}|empty} - this lets you get the local AS number using '{bgp-as-local}' or you can add or subtract to this to adjust it like so '{bgp-as-local-4}' or '{bgp-as-local+3}' This will just add or subtract to the as number that is returned. You can determine what Model's AS number will be returned based on when you have decided to increment the AS number in the configuration template in the bgp routing protocol.

NTP This just allows you to specify the NTP server address.

Template bgpospf.ini ▾ Wiring Context generic_top.ini ▾ Submit

Name bgpospf

Networks

Network 0 192.168.0.0
Network 1 10.10.0.0
Network 2 10.1.0.0

Add New Network | Remove Network

Mappings

Router: Router 1	Model: Model1
Router: Router 2	Model: Model2
Router: Router 3	Model: Model1
Router: Router 4	Model: Model2

Add New Mapping | Remove Mapping

Models

Model1

Interfaces (direction network): + [-]
5 0 ↘ 24 >
6 2 ↘ 32 >
1 1 ↘ 31 -> Model2 2 <input checked="" type="radio"/> Internal <input type="radio"/> To Next System <input type="radio"/> To Previous System
4 1 ↘ 31 -> Model1 3 <input type="radio"/> Internal <input type="radio"/> To Next System <input checked="" type="radio"/> To Previous System 1 Systems Back
3 1 ↘ 31 -> Model1 4 <input type="radio"/> Internal <input checked="" type="radio"/> To Next System <input type="radio"/> To Previous System
Routing: (title) + [-]
bgp_test
ospf_test
Components: (title) + [-]
prefix_test

Model2

Interfaces (direction network): + [-]
5 0 ↘ 24 >
6 2 ↘ 32 >
2 1 ↘ 31 -> Model1 1 <input checked="" type="radio"/> Internal <input type="radio"/> To Next System <input type="radio"/> To Previous System
4 1 ↘ 31 -> Model2 3 <input type="radio"/> Internal <input type="radio"/> To Next System <input checked="" type="radio"/> To Previous System 1 Systems Back
3 1 ↘ 31 -> Model2 4 <input type="radio"/> Internal <input checked="" type="radio"/> To Next System <input type="radio"/> To Previous System
Routing: (title) + [-]
bgp_test
ospf_test
Components: (title) + [-]
acl_test
routemap_test

Add New Model | Remove Model

Routing

bgp_test

BGP
Router ID 1.0.0.0
AS Start 65000 Increment AS After Model2
Advertisements (interface): + [-]
1 2 3 4 5

ospf_test

OSPF
Router ID 1.0.0.0
Process ID 1
Advertisements (interface): + [-]
1 2 5 6

Add New Routing | Remove Routing

Components

prefix_test

Prefix List
Room All Rack All Kit All
permit {interface-network-Model1-4}/{interface-slash-Model1-4}
Name PrefixList1 Rules

routemap_test

Route Map
Room All Rack Black Kit All
match ip address MATCH_ACL set ip default next-hop {interface-Model1-1} set default interface FastEthernet 0/1
Name ROUTE_MAP Permit <input checked="" type="checkbox"/> Num 10 Statements
Attached To (interface): + [-]
2

Add New Component | Remove Component

Step 3 Set Replication

Replication is where you will decide what devices are to be used to deploy your network, if deploying multiple networks to the same set of devices you can set what VRFs are to be used.

The first step will be to attach your templates to a VRF in the VRF Box. On the left side you will specify the name of the VRF and on the right dropdown you will select the template to attach. If you want the template deployed on the global instance, you can use the “Global” keyword.

NOTE: using the global keyword will mean there won’t be a VRF
Made and the template will deploy on the global router instance.

The remaining checkboxes will specify what devices are to be included in the deployment of the network. You can check each box for each Room, Rack, Kit and Device you want included. The arrows allow you to change the order of replication, this is useful if the rooms/racks/kits are connected in a different order.

NOTE: The order of replication implies that the previously replicated container is connected to the next replicated container. In the below example the replication order for kits is White, Purple, Orange, Green, Yellow. This implies that White Connects to Purple, Purple Connects to Orange and so on.

The checkboxes that are external to the fieldset allow for specific replication orders based on the parent container. This is useful if the connections of racks are different in one room to another, or if the connections of kits are different from one rack to another.

In the below example Rooms ATC328 and ATC329 have racks that are connected to each other differently. The settings specify a default connection under `replicate[racks] [all]` as being Black, Red, Blue, Green, Yellow. For ATC329 this order is overridden under the more specific setting `replicate[racks] [ATC329]` its order being Red, Blue,Green,Black,Yellow.

NOTE the last of a child container will connect to the first of the next child container according to how the parent containers are connected. The below example shows that ATC328 connected to ATC329. Which means the last container in ATC328 (Rack Yellow) will connect to the first container in ATC329 (rack Red).

NOTE: Most of the time you will want to use all routers and keep them in the same order as the connections are specified in the template. However if using a 3 Model topology with 3 devices, you will need to remove the unused router from the replication.

NOTE: When loading the replication file you will notice that the checkboxes do not load. If the file is not overwritten then it still contains the replication information, it just does not load it in the web interface.

Step 4 Deploy Configuration

Go to the Upload Configuration Page.

[Home](#) |
 [Update/Check Wiring Scheme](#) |
 [Construct Configuration](#) |
 [Create Replication Scheme](#) |
 [Upload Configuration](#)
 Replication File: replication.ini

Name:	replication	<input type="button" value="Save"/>
VRFs		
Global	bgpospf.ini	<input type="button" value="▼"/>
Management	management.ini	<input type="button" value="▼"/>
<input type="button" value="Add New VRF"/> <input type="button" value="Remove VRF"/>		
Replicate		
Rooms		
ATC328	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
ATC329	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
ATC330	<input type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
<input type="checkbox"/> ATC328 rack <input checked="" type="checkbox"/> ATC329 rack <input checked="" type="checkbox"/> All racks		
Racks		
replicate[Rack][ATC329]replicate[Rack][All]		
Red	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Blue	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Green	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Black	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Yellow	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
<input type="checkbox"/> Black kit <input type="checkbox"/> Red kit <input type="checkbox"/> Blue kit <input type="checkbox"/> Green kit <input type="checkbox"/> Yellow kit <input checked="" type="checkbox"/> All kits		
Kits		
replicate[Kit][All]		
White	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Purple	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Orange	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Green	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Yellow	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Routers		
Router 1	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Router 2	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Router 3	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>
Router 4	<input checked="" type="checkbox"/>	<input type="button" value="^"/> <input type="button" value="v"/>

Figure 4: Replication Template

1. First you will need to create the configuration files for each router. This is achieved in the first form by entering the replication file and physical wiring scheme, this will produce the config files to upload to the devices. The job name will be the name it is saved under to later upload.
2. If you want to check that the physical topology is wired as expected you can select the physical wiring scheme and replication of your choice to be validated on the machines. If there are disconnections the response will tell you. For more details See Physical Topology Check.
3. Using the final form you select the config files to upload by the same name as the job name from the previous step. Enter your credentials to upload to the ATC devices.

NOTE: You only need to create the configurations once after which they will be saved to be uploaded as needed.

[Home](#) | [Update Check Wiring Scheme](#) | [Construct Configuration](#) | [Create Replication Scheme](#) | [Upload Configuration](#)

Create Configurations

Replication File Wiring Context Job Name

Check Topology

Replication File Physical Topology Username Password

Upload Configurations

Job Username Password

Figure 5: Upload Form

Physical Topology Check

The physical topology check provides a list of issues, after conducting interface status checks, CDP checks and ping tests. It can determine the following things:

- Whether an interface is disconnected or down, and what interface it should be connected to.
- Whether an interface is connected to the wrong interface, and what interface it should be connected to.
- Whether an interface should be up, and is a loopback, or as a connection to a PC.
- Whether an interface does not exist i.e the name on the physical template does not exist on the device
- Whether an interface is connected, but can not ping.

Physical templates

The physical template is treated as the source of truth for the topology. So incorrectly naming devices will create an error that the device could not be connected to, and incorrectly naming interfaces will create an error that they don't exist on the devices.

Replication templates

The replication template dictates what devices are to be tested. The devices that will be actually tested will be the devices you booked that exist on the replication scheme. If you did not book all the devices on the replication template you will be warned. Similarly if you include devices on the replication template that aren't on the physical template you will be warned.

Interconnected devices

For topologies that involve devices connected across kits and racks, it will be able to determine whether you have connected devices incorrectly, however in some cases this is only discernable through a ping test, such as if you connect Router 1 to Router 2 using the correct interface but to the wrong kit. This is because CDP will look correct, and only a ping will identify that it is incorrectly connected. As ping tests are not implemented for the switches, interconnected Switches are not supported.