



Enterprise Onboarding in CUPS

- [Feature Revision History, on page 1](#)
- [Feature Description, on page 1](#)
- [How it Works, on page 3](#)
- [Enterprise Onboarding in CUPS OAM Support, on page 20](#)

Feature Revision History



Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In CUPS architecture, User Planes (SAEGW-U) are grouped into a logical concept called User Plane Group (UP Group) and controlled by a Control Plane (CP) node. An APN is associated with a UP Group, and the UP for IP pool is selected based on least-used User Plane.

During configuration of new APNs and IP pools, the operator must decide on a UP Group to be used. The information required to decide the UP Group is not exposed by the system and the process is tedious and error prone. Also, the number of contexts, APNs, VRFs, and IP pools are reduced both on CP and UP in CUPS architecture as compared with ASR 5500. This also limits the addition of new APNs and IP Pools to the right context and UP Group.

The Intelligent Onboarding (IOB) tool automates the procedure of choosing the right UP Group and SGi context for the new APN to be added. The tool gathers current resources that are configured (number of UP Groups, UPs per group, existing contexts, APNs, and IP pools) in the CUPS system. It then determines if the system can absorb the new configuration and determines the UP Group that can support without breaching the system limits. In line with this, the new configuration is applied by the tool.

Operational Use Case

The Enterprise requires an operator to add, modify, and/or delete a user with information based on APN and IP pools. The tool generates and applies the required configuration to add, modify, or delete an APN in the CUPS environment.

The following operations can be performed:

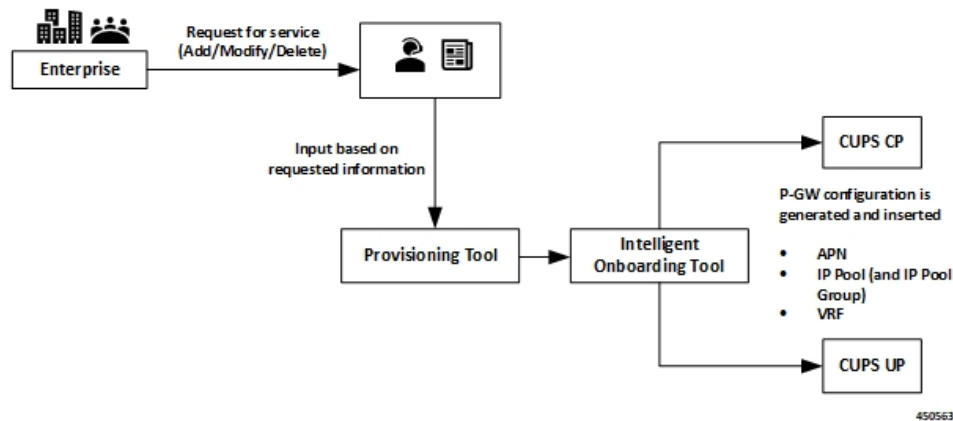
- Enterprise Addition: A new APN is added with required number of IPv4/IPv6 pools.
- Enterprise Modification: IP pools can be added/deleted for an existing APN.
- Enterprise Deletion: An APN will be deleted.

In 21.20.13 and later releases, the IOB tool also supports the onboarding of one or more virtual APNs in one operation. As part of this operation, one or more existing APNs can be modified to reference these new virtual APNs. Similarly, the tool also supports deleting the set of virtual APNs onboarded together and simultaneously removing existing references to those APNs from other APNs.

Architecture

On ASR 5500, Enterprise addition consists of adding a new APN. For CUPS, along with the APN configuration, we must include the correct UP Group and SGi context configuration.

The IOB tool takes inputs from the Provisioning tool, chooses the best suited UP Group and SGi context for the APN, and configures the CP and UP. The IOB tool also allows modification of the APN configuration (adding/deleting the IP pools) and Deletion of an APN.



If onboarding multiple APNs, the APN configuration section must list:

- The configuration for all onboarding APNs and
- APNs referring to them (in case of virtual APNs)

In the preceding scenario, all the APNs get onboarded onto the same UP group and SGi context.

Installation

The IOB tool is shipped as Linux executable. All dependencies, like Pexpect and connection management library, are packaged into the standalone .exe file.

The tool is shipped with StarOS images and signed with the same keys that are used for StarOS VPC-SI image.

The executable tool requires the following environment:

- RedHat Enterprise Linux 7.6 (or CentOS equivalent) 64-bit installation
- OpenSSL version 1.0.2.k-fips
- The following shared libraries are installed under /lib64 (these are typically present in a standard RHEL or CentOS installation):
 - libdl.so.2
 - libz.so.1
 - libc.so.6
 - ld-linux-x86-64.so.2
- Read, write, execute permissions for /tmp directory. While executing, the tool creates a temporary directory under /tmp, extracts sections of the executable to this temporary directory and executes the sections.
- Sufficient disk space for the tool and the log files (current usage is approximately 10 MB)
- IP connectivity to CPs and UPs on which onboarding is to be done. Password-based SSH is used for connections.

How it Works

The IOB tool is a standalone application that leverages StarOS CLIs to collect the system level resources, read the configurations, check the errors, SRP information, and so on. The input parameters to the IOB tool include addressing and login credentials for CPs and UPs, details of the operation (add/modify/delete), and the specific configuration to be applied. Since the contexts to apply the configuration to may not be known beforehand, the input configuration specifies a dummy context as a placeholder. The IOB tool substitutes that dummy context with the specific context that is chosen prior to applying the configuration.

Also, as part of Enterprise Onboarding solution, a new CLI command is introduced, and an existing CLI command is modified. For details, see *Enterprise Onboarding in CUPS OA&M Support* section.

The IOB tool goes through the following steps:

- **Pre-processing:** This is performed to ensure that the system is in stable state to proceed with the onboarding operation. On successful validation, the IOB tool collects the current resource usage information from the system.
- **Context and UP Group Selection:** The IOB tool applies the onboarding algorithm to select a context and UP Group to onboard the APNs.
- **Configuration:** Based on the operation to be performed, an algorithm is applied using the data collected in the Pre-processing step. The configuration is then applied on CP and UP. For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- **Post-processing:** Post configuration checks are performed to validate the system for any errors. For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- **Logging:** The entire operation is logged. The logging mechanism captures the output of the operation, history of the operation, Warnings/Error messages, and any other information that helps in debugging.

Pre-Processing

Pre-processing step helps in understanding the status of the CUPS system where the onboarding operation is being performed. In the pre-processing stage, following checks are performed irrespective of the operation:

- Verify if all CP and UP management IPs are reachable:
 - Ping Active/Standby management IPs of all the CPs.
 - Ping Active/Standby management IPs of all the UPs.
- Collect the resources information (APN, IP Pools, VRF, and Context) based on the output of:
 - **show ip user-plane verbose**
 - **show cups-resources session summary**
- **Add Operation:**
 - On Control Plane node, following checks are performed:
 - Verifies that the VRF, APN, and IP pool to be onboarded is not configured in the system. If onboarding one or more virtual APNs, then the APNs that refer to these virtual APNs must be already present on the system. The tool uses the presence of the following configuration in the APN to distinguish these APNs.


```
virtual-apn gcdr apn-name-to-be-included Gn
```

So, given an input configuration with one or more APNs, then any APN that is already present in the system must include the preceding configuration. Otherwise, the tool assumes that the APN isn't present and hence fails the preaudit step.
 - Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.
 - After context and UP Group selection, on User Plane node, the following pre-processing checks are performed on all the UPs of the selected UP Group:
 - Verifies that the VRF to be onboarded doesn't exist in the system. If it exists, then the pre-processing fails and onboarding is aborted.
 - Verifies that there is no configuration difference between Active/Standby UPs using **show srp info**.
 - Verifies if SGi context is mapped in the UP Groups.
- **Modify Operation:**
 - On Control Plane node, following checks are performed:
 - Verifies that the VRF to be modified exists in the system.

- Verifies that the APN to be modified exists in the system.
 - Verifies that the IP pools, deleted as part of modify operation, exists in the system. Any IP pool that is added as part of modify operation, doesn't exist in the system.
 - Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.
- **Delete Operation:**
 - On Control Plane node, following checks are performed:
 - Verifies that the VRF to be deleted exists in the system.
 - Verifies that the APN(s) to be deleted exists in the system.
 - Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.
 - On User Plane node, following checks are performed:
 - Verifies that the VRF to be deleted exists in the system.
 - Verifies that there is no configuration difference between Active/Standby UPs using **show srp info**.

CP and UP Configuration

On successful pre-processing, the tool performs the Add/Modify/Delete operation as per the input and applies the configuration on CP and UP. For ICSR setups, the configuration is applied on both Active and Standby CP and UPs.

- Add operation: The algorithm chooses the right SGi context and UP group for the APN to be added.
 - On Control Plane node, following steps are performed:
 - The chosen SGi context and the UP Group are added to the APN configuration, which goes as input to the tool. In case of onboarding virtual APNs, only the onboard virtual APNs get updated with UP Group and IP context. The APNs that refer to them (which is already present in the system) just gets updated with any **virtual-apn preference ..** configuration that is present in the input file.
 - The updated configuration is then applied to the CP node.
 - On User Plane node:
 - The IOB tool replaces the dummy SGi context with chosen context, and applies the resulting configuration to all the UPs in the chosen UP Group.
 - Applies VRF configurations to all the UPs in the UP Group.
 - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- Modify Operation: Configuration is modified to add or delete the IP pools.

- On Control Plane node:
 - For the given APN configuration, IP pool configuration is modified to add/delete the IP pools. If any IP pools are deleted, then prior to deletion, the tool:

- Busyouts the pool.

- Clears existing subscribers for that pool per pace-out interval. The pace-out interval is calculated based on the size of the pool.

For IPv6 pools, the formula is:

$$\text{Pace-out interval} = (2^{(64 - \text{pool size})} * 2 - 2) / 500$$

So, a /48 pool will get a pace-out interval of $(2^{(64 - 48)} * 2 - 2) / 500 = (2^{16} * 2 - 2) / 500 = 131070 / 500 = 262$ seconds

For IPv4 pools, the formula is:

$$\text{Pace-out interval} = (2^{(32 - \text{pool size})} * 2 - 2) / 500$$

So, a /21 pool will get a pace-out interval of $(2^{(32 - 21)} * 2 - 2) / 500 = (2^{11} * 2 - 2) / 500 = 4094 / 500 = 8$ seconds

- For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- Delete Operation: Deletes the APN.
 - On Control Plane node:
 - IP pools and VRFs, associated with the APN, are deleted.

Prior to deleting any APN, the IOB tool verifies if any user is attached to the given APN. If any user exists, it exits from the tool and displays an error message "Please clear the subscribers then run the DELETE_ENTERPRISE else it will delete the APN".
 - APN configuration is deleted.
 - Deleting the virtual APNs removes only the virtual APNs and references to the virtual APNs. The APNs that refer to them are expected to remain in the system. Otherwise, post audit will fail.

- On User Plane node, VRF configurations are deleted.

The IOB tool doesn't rollback to the previous configuration on a failure. It, however, tries to delete as much of the relevant configuration as possible to minimize the amount of manual clean-up required.

Post-Processing

After the configurations are pushed to CP and UP, checks are performed to validate configuration changes.

- Add Operation:
 - On Control Plane node, following checks are performed:

- Verifies configured VRF with **show ip vrf** *vrf_name*: To verify if the VRF configuration is applied in the CUPS system.
 - Verifies that the chosen context is shown with **show configuration apn** *apn_name*: To verify if the context has been associated with the APN that is added. This verification takes place for each APN that is onboard. If there are virtual APNs onboard, then this verification takes place only for each virtual APN.
 - Verifies that the chosen UP Group is shown under **show configuration apn** *apn_name*: To verify if the UP Group has been associated with the APN that is added.
 - If there are virtual APNs onboard, then the tool verifies that all the references to the virtual APNs from other APNs as per the input configuration (**virtual-apn preference** *<preference>* **apn** *<virtual apn>* and so on) are present and correct.
 - Saves configuration using **save configuration** *file_path / file_name*: After successful addition of new enterprise, checks if the respective configuration files are stored in the given path as mentioned in "CUPSinfo.txt" file.
 - Synchronize configuration on CPs with **filesystem synchronize**: After successful addition of new enterprise, verifies the file synchronization.
 - Verifies that there is no configuration difference between CPs using **show srp info**: SRP validation in ICSR setup: After successful addition of new enterprise, the IOB tool checks for SRP validation with "Primary" and "secondary" status, "Last Peer Configuration Error", "Connection State", along with "Number of Sessmgrs".
- On User Plane node, following checks are performed:
 - Verifies configured VRF with **show ip vrf** *vrf_name*: To verify the VRF configuration applied in the CUPS system.
 - Verifies Route Distinguisher using **show ip vrf** *vrf_name*: To verify the Route Distinguisher configuration applied in the CUPS system.
 - Save configuration using **save configuration** *file_path / file_name*.
 - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup.
 - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- Modify Operation:
 - On Control Plane node, following checks are performed:
 - Verifies that the modified changes are applied to the CUPS system.
 - Verifies that the changes to IP pool are reflected in the system
 - Saves configuration using **save configuration** *file_path / file_name*.
 - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup
 - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- Delete Operation:
 - On Control Plane node, following checks are performed:
 - This verification takes place for each APN after the delete operation. The delete operation for the virtual APN removes only the references to the virtual APN and retains the APNs that refer to them. Removal of the latter APNs causes the post processing to fail.
 - Verifies if the VRF configuration is deleted from the CUPS system.
 - Saves configuration using **save configuration file_path / file_name**.
 - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup
 - On User Plane node, following checks are performed:
 - Verifies if the VRF configuration is deleted from the CUPS system.
 - Saves configuration using **save configuration file_path / file_name**.
 - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup.

Add Operation

The Add operation configures a new APN for the enterprise customer. The tool also supports onboarding multiple APNs in one operation, provided they share the same SGi context and VRF configuration. In this case, the onboarding APNs may or may not share the IP pool information (supports both the conditions). All the onboarding APNs map to the same SGi context and UP group in the preceding scenario. The algorithm chooses the right SGi context and UP Group, and maps them to the APN by taking the system parameters into consideration.

Algorithm Logic:

- Check for System Limits (done against CP limits mentioned in [System Limits, on page 18](#)). In case of onboarding virtual APNs, the tool only considers the virtual APNs as new APNs for APN limit calculation. The APNs referring to the virtual APNs are already present in the system and are hence already included in the current count of APNs in the system.
- Rank UP Groups based on number of APNs configured with low numbers on top.
- Sort the SGi contexts based on the number of VRFs configured in ascending order.
- Exclude VIP UP Groups and Contexts from the list.
- Pick a UP Group from top of the list (least-used):
 - Get a Context that is mapped to UP Group (if no Contexts are mapped, pick from the sorted list).
 - Check the number of VRFs, IPv4, IPv6 pools, and total pool size.
 - Choose the Context if checks fall within thresholds; else, repeat for next Context.
 - Pick suitable Context within limits; if none found, exit algorithm.
 - For this UP Group, iterate through UPs and check total IP pool limits.

- If successful, choose the UP Group and Context.
- Iterate through all UP Groups.
- At each step, while checking against the thresholds, print error messages.
- Prepare the configuration with chosen Context and UP Group and apply.

Modify Operation

Modify operation allows the onboarded Enterprise customer to increase/decrease the subscribers by Adding more IP pools or deleting the existing IP pools.

Delete Operation

Delete operation removes a previously onboarded Enterprise. During this operation, the IOB tool cleans up the IP pools, VRFs, and APNs that are used for the Enterprise.

To delete an Enterprise, the following procedure must be followed as there may be active subscribers on the system:

- Busyout the IP Pools: This is performed to block the new subscribers. Invoke IOB tool and perform the Busyout operation using the MODIFY operation.
- Clear Subscribers: The Provisioning tool clears the active subscribers.
- Delete the Enterprise: Invoke the IOB tool and perform the enterprise removal using the DELETE operation.

Password Encryption

The IOB tool expects passwords in the "CUPSInfo.txt" input file to be RSA encrypted and converted to base64 format. Encryption is done using OpenSSL (currently, version 1.0.2.k is supported) commands and RSA public key. The IOB tool must be provided the path to the corresponding RSA private key so that it can decrypt the passwords. The decrypted passwords are stored only in the IOB tool's RAM. The detailed steps for encryption and decryption are described below:

1. Verify that the OpenSSL with the correct version is installed on the target machine:
 - "openssl version" should indicate that the version is 1.0.2.k-fips.
2. Generate RSA private and public key pair:
 - a. RSA private key:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:4096
```

Where:

- "private_key.pem" represents the generated private key file in PEM format. This is used for decryption and has to be stored securely.

- 4096 is the key length in bits. Either 2048 or 4096 can be used. Multiple passwords may need to be encrypted and so, 4096 is recommended. Generally, the larger the key size, the larger the size of data it can encrypt. However, it also takes longer to encrypt/decrypt.

b. RSA public key:

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Where:

- "private_key.pem" is the private key generated in Step (a).
- "public_key.pem" is the file that contains the corresponding public key.

3. For each password that needs to be encrypted, do the following:

- a.** Type the password in plaintext in a text file using an editor. Don't hit enter at the end of the line. It should have just the password in a single line. In this example, the file is named as "pp1".

b. Execute:

```
openssl pkeyutl -encrypt -inkey public_key.pem -pubin -in pp1 -out encrypted_pp1
```

Where:

- "public_key.pem" is the public key generated in Step 2b.
- "pp1" is the file containing the single password in plaintext.
- "encrypted_pp1" contains the password in encrypted form.

Delete "pp1" created in *Step 3a* to avoid accidental exposure.

- c.** "encrypted_pp1" contains the key in raw binary form. Convert it to base64 as follows:

```
base64 encrypted_pp1
```

- d.** The above command (Step 3c) will output the base64 encoded encrypted password to the terminal. Copy and paste this into the "CUPSinfo.txt" file that contains the credentials supplied to the IOB tool. While copying, make sure to remove any line breaks or spaces. The entire password should be a single line.
- e.** "encrypted_pp1" can be deleted at this point.



Note *Step 3* must be performed for each password, one at a time, using the same public key/private key pair for all the passwords.

After "CUPSinfo.txt" file is updated with all the encrypted base64 passwords, the IOB tool is ready to be run. When running the script, specify an additional parameter: **-k** *absolute path to private_key.pem created in Step 2a*>.

Onboarding Application – Usage and Input Parameters

The application is compiled to create a standalone .exe. The application can be run on a RedHat Enterprise Linux machine.

The Onboarding Application can be run with below syntax:

```
./intelligent_onboarding -o <OP_Type_Parameter_File> -i <CUPS_Info_File> -k
<Path_to_Pvt_Key_file> [ -l <Path_to_store_logfiles> ] [ -p ] [ --context_selection_from_cp
] [ -v ]
```

Options:

- **-o:** [Mandatory] Provide the input parameter file specific to the operation being invoked.
After successful onboarding, the IOB tool deletes the file.
- **-i:** [Mandatory] This option is used for "CUPSinfo.txt" file which has the details of CUPS system.
- **-k:** [Mandatory] Absolute path to the private key file. The tool uses this to decrypt the previously encrypted passwords. This private key file must correspond to the public key that is used to encrypt the passwords.
- **-p:** [Optional] When included, few pre-audit and post-audit checks are bypassed to reduce the time taken for Add/Modify/Delete operation.
- **-l:** [Optional] Provide absolute path to store the logs.

When this keyword is not specified, the log files are created in the directory from which the IOB tool is invoked.

- **--context_selection_from_cp:** [Optional] When specified, the tool bases its context selection solely on the list of contexts available on the CP. The tool assumes that the selected context is also available on the UPs and does not validate this. This is an optimization. The default behavior is to examine contexts configured on CP and UP and select from contexts common to both.
- **-v:** [Optional] Displays the version of the IOB executable.

If IOB tool is executed without the **-v** option, the version is displayed that is similar to:

```
#####
#
#           WELCOME TO ENTERPRISE ONBOARDING           #
#                   Version 21.20.9.private                   #
#
#####
```

NOTE: The version is displayed in the log file and terminal output as well.

CUPSinfo.txt

Onboarding application must know the system-level details to carry out the onboarding operations. The "CUPSinfo.txt" file has the IP addresses for CP and UP nodes and configurable threshold values. "Skip_UPGroup" and "Skip_Context" refers to the UP Groups and contexts that must not be considered for onboarding algorithm. For example, VIP groups and contexts that cannot be used for other enterprises. The file specifies a path where the configuration must be saved. The passwords in this file must be specified in RSA encrypted, base 64 format.

In 21.20.9 and earlier releases, the entry order of CP and UP inputs were:

```
//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Primary_config_path,Secondary_config_path
```

```
//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Sx-IP-Address,Primary_config_path,Secondary_config_path
```

In 21.20.10 and later releases, the entry order of CP and UP inputs are:

```
//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Primary_config_path,Secondary_config_path,Login,Password
//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Sx-IP-Address,Primary_config_path,Secondary_config_path,Login,Password
```

Sample CUPSinfo.txt File

For 21.20.9 and earlier releases:

```
//Threshold for Warning, input as percentage values

CPContext_threshold = {vrf_threshold:80, ipv4_threshold:80, ipv6_threshold:80}
CPSystem_threshold = {vrf_threshold:80, total_pool_threshold:80, apn_threshold:80}
UPContext_threshold = {vrf_threshold:80, ipv4_threshold:80, ipv6_threshold:80}
UPSystem_threshold = {vrf_threshold:80, apn_threshold:80, total_pool_threshold:80}
UPBudgeted_Sessions_threshold = {budgeted_threshold:80}

SKIP_UPGroup =
SKIP_Context =

//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Primary_config_path,Secondary_config_path
cups_di_cp1,Control_Plane,209.165.200.225,209.165.200.225,<login_id>,<password>,
/flash/209.165.200.225-cups-vpp-saegw-global-control-plane.cfg,
/flash/209.165.200.225-cups-vpp-saegw-global-control-plane.cfg

//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Sx-IP-Address,Primary_config_path,Secondary_config_path
cups_di_up0,User_Plane,209.165.200.230,209.165.200.230,<login_id>,<password>,
209.165.200.238,/flash/209.165.200.230-cups-vpp-saegw-global-user-plane-.cfg,
/flash/209.165.200.230-cups-vpp-saegw-global-user-plane.cfg
cups_di_up1,User_Plane,209.165.200.235,209.165.200.235,<login_id>,<password>,
209.165.200.242,/flash/209.165.200.235-cups-vpp-saegw-global-user-plane.cfg,
/flash/209.165.200.235-cups-vpp-saegw-global-user-plane.cfg
```

In 21.20.10 and later releases:

```
//Threshold for Warning, input as percentage values

CPContext_threshold = {vrf_threshold:98, ipv4_threshold:98, ipv6_threshold:98}
CPSystem_threshold = {vrf_threshold:98, total_pool_threshold:98, apn_threshold:98}
UPContext_threshold = {vrf_threshold:98, ipv4_threshold:98, ipv6_threshold:98}
UPSystem_threshold = {vrf_threshold:98, apn_threshold:98, total_pool_threshold:98}
UPBudgeted_Sessions_threshold = {budgeted_threshold:80}

SKIP_UPGroup =
SKIP_Context =

//Control_Plane: Host,Node,Primary_IP,Secondary-IP,Primary_config_path,
Secondary_config_path,Login,Password
cups_di_cp1,Control_Plane,209.165.200.225,209.165.200.225,/flash/209.165.200.225-CP01.cfg,
/flash/209.165.200.225-CP02.cfg,<login_id>,<password>
//User_Plane: Host,Node,Primary_IP,Secondary-IP,Sx-IP-Address,Primary_config_path,
Secondary_config_path,Login,Password
cups_si_up1,User_Plane,209.165.200.235,209.165.200.235,209.165.200.242,/flash/209.165.200.235-UP01.cfg,
/flash/209.165.200.235-UP02.cfg,<login_id>,<password>
```

ADD_ENTERPRISE_INPUT_PARAMETERS.txt

This file provides the configuration information when an APN is added. It provides the IP pool information and VRF information. The context provided is dummy and the actual context is determined as part of the algorithm. The IP pools doesn't support chunks.

Sample ADD_ENTERPRISE_INPUT_PARAMETERS.txt

Following is the example configuration for onboarding a single APN.

```
OpType = "ADD_ENTERPRISE"

CP_APN_Config = '''Config
context APN
    apn starent.com
ip address pool name starent_ipv4_pool_group_01
ipv6 address prefix-pool starent_ipv6_pool_group_01
    exit
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGI_Context = '''Config
    context dummy-SGI
ip vrf MPN00001
ip pool starent_ip_pool_v4_001 209.165.200.225 255.255.255.250 private 0 no-chunk-pool
group-name starent_ipv4_pool_group_01 vrf MPN00001
ip pool starent_ip_pool_v4_002 209.165.200.228 255.255.255.250 private 0 no-chunk-pool
group-name starent_ipv4_pool_group_01 vrf MPN00001

ipv6 pool starent_ip_pool_v6_001 prefix 2001:1:1::/48 private 0 no-chunk-pool group-name
starent_ipv6_pool_group_01 vrf MPN00001

    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config= '''config
context dummy-SGI
ip vrf MPN00001
ip maximum-routes 100
exit
router bgp 65101
ip vrf MPN00001
route-distinguisher 65101 11100001
route-target both 65101 11100001
exit
address-family ipv4 vrf MPN00001
redistribute connected
exit
address-family ipv6 vrf MPN00001
redistribute connected
exit
exit
exit
exit'''
```

Following is the example configuration for onboarding multiple virtual APNs in one ADD operation.

```
OpType = "ADD_ENTERPRISE"

CP_APN_Config = '''Config
```

```

context APN
apn virtual1
    ip address pool name apn2_ipv4_pool_group_01
    ipv6 address prefix-pool apn2_ipv6_pool_group_01
exit
apn virtual2
    ip address pool name apn2_ipv4_pool_group_02
    ipv6 address prefix-pool apn2_ipv6_pool_group_02
exit
apn virtual3
    ip address pool name apn2_ipv4_pool_group_03
    ipv6 address prefix-pool apn2_ipv6_pool_group_03
exit
apn virtual4
    ip address pool name apn2_ipv4_pool_group_04
    ipv6 address prefix-pool apn2_ipv6_pool_group_04
exit
apn virtual5
    ip address pool name apn2_ipv4_pool_group_05
    ipv6 address prefix-pool apn2_ipv6_pool_group_05
exit
apn virtual6
    ip address pool name apn2_ipv4_pool_group_06
    ipv6 address prefix-pool apn2_ipv6_pool_group_06
exit
apn virtual7
    ip address pool name apn2_ipv4_pool_group_07
    ipv6 address prefix-pool apn2_ipv6_pool_group_07
exit
apn virtual8
    ip address pool name apn2_ipv4_pool_group_08
    ipv6 address prefix-pool apn2_ipv6_pool_group_08
exit
apn virtual9
    ip address pool name apn2_ipv4_pool_group_09
    ipv6 address prefix-pool apn2_ipv6_pool_group_09
exit
apn virtual10
    ip address pool name apn2_ipv4_pool_group_10
    ipv6 address prefix-pool apn2_ipv6_pool_group_10
exit
apn real1
    virtual-apn preference 1 apn virtual2 domain virtual2
    virtual-apn preference 2 apn virtual3 domain virtual3
    virtual-apn preference 3 apn virtual4 domain virtual4
exit
apn real2
    virtual-apn preference 3 apn virtual5 domain virtual5
    virtual-apn preference 6 apn virtual6 domain virtual6
    virtual-apn preference 9 apn virtual7 domain virtual7
exit
apn real3
    virtual-apn preference 2 apn virtual6 domain virtual6
    virtual-apn preference 5 apn virtual7 domain virtual7
    virtual-apn preference 8 apn virtual8 domain virtual8
exit
apn real4
    virtual-apn preference 2 apn virtual8 domain virtual8
    virtual-apn preference 3 apn virtual9 domain virtual9
    virtual-apn preference 5 apn virtual10 domain virtual10
exit
apn real5
    virtual-apn preference 7 apn virtual10 domain virtual10
    virtual-apn preference 8 apn virtual11 domain virtual11

```

```

        virtual-apn preference 9 apn virtual2 domain virtual2
    exit
    apn real6
        virtual-apn preference 11 apn virtual10 domain virtual10
        virtual-apn preference 12 apn virtual11 domain virtual11
        virtual-apn preference 13 apn virtual2 domain virtual2
    exit
    apn real7
        virtual-apn preference 12 apn virtual2 domain virtual2
        virtual-apn preference 13 apn virtual3 domain virtual3
    exit
    apn real8
        virtual-apn preference 12 apn virtual7 domain virtual7
    exit
    apn real9
        virtual-apn preference 12 apn virtual5 domain virtual5
        virtual-apn preference 13 apn virtual6 domain virtual6
        virtual-apn preference 14 apn virtual7 domain virtual7
        virtual-apn preference 15 apn virtual8 domain virtual8
        virtual-apn preference 16 apn virtual9 domain virtual9
        virtual-apn preference 17 apn virtual10 domain virtual10
        virtual-apn preference 18 apn virtual2 domain virtual2
        virtual-apn preference 19 apn virtual3 domain virtual3
    exit
    apn real10
        virtual-apn preference 1 apn virtual11 domain virtual11
    exit
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''Config
    context dummy-SGi
        ip vrf MPN00002
        ip pool apn2_ip_pool_v4_001 209.165.201.1 255.255.255.224 private 0 group-name
apn2_ipv4_pool_group_01 vrf MPN00002 no-chunk-pool
        ip pool apn2_ip_pool_v4_002 209.165.201.3 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_02 vrf MPN00002
        ip pool apn2_ip_pool_v4_003 209.165.201.5 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_03 vrf MPN00002
        ip pool apn2_ip_pool_v4_004 209.165.201.7 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_04 vrf MPN00002
        ip pool apn2_ip_pool_v4_005 209.165.201.9 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_05 vrf MPN00002
        ip pool apn2_ip_pool_v4_006 209.165.201.11 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_06 vrf MPN00002
        ip pool apn2_ip_pool_v4_007 209.165.201.13 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_07 vrf MPN00002
        ip pool apn2_ip_pool_v4_008 209.165.201.15 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_08 vrf MPN00002
        ip pool apn2_ip_pool_v4_009 209.165.201.17 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_09 vrf MPN00002
        ip pool apn2_ip_pool_v4_010 209.165.201.19 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_10 vrf MPN00002

        ipv6 pool apn2_ip_pool_v6_001 prefix 2001:268:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_01 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_002 prefix 2001:278:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_02 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_003 prefix 2001:288:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_03 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_004 prefix 2001:298:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_04 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_005 prefix 2001:2A8:1::/48 private 0 no-chunk-

```

MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt

```

pool group-name apn2_ipv6_pool_group_05 vrf MPN00002
  ipv6 pool apn2_ip_pool_v6_006 prefix 2001:2B8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_06 vrf MPN00002
  ipv6 pool apn2_ip_pool_v6_007 prefix 2001:2C8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_07 vrf MPN00002
  ipv6 pool apn2_ip_pool_v6_008 prefix 2001:2D8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_08 vrf MPN00002
  ipv6 pool apn2_ip_pool_v6_009 prefix 2001:2E8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_09 vrf MPN00002
  ipv6 pool apn2_ip_pool_v6_010 prefix 2001:2F8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_10 vrf MPN00002
  exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config = '''config
  context dummy-SGI
    ip vrf MPN00002
      ip maximum-routes 100
    exit
  router bgp 65101
    ip vrf MPN00002
      route-distinguisher 65101 11100002
      route-target both 65101 11100002
    exit
  address-family ipv4 vrf MPN00002
    redistribute connected
  exit
  address-family ipv6 vrf MPN00002
    redistribute connected
  exit
  exit
exit
exit'''

```

MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt

This file provides the IP pools that must be added to or deleted from an existing enterprise. The context name is determined based on the pool name.

Sample MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt

```

OpType = "MODIFY_ENTERPRISE"
CP_APN_Config = '''Config
  context APN
    apn cisco.com
  exit
  exit
exit'''
CP_SGi_Context = '''Config
  context dummy-SGi
    no ip pool cisco_ip_pool_v4_002 209.165.202.129 255.255.255.224 private 0 no-chunk-pool
  group-name starent_ipv4_pool_group_01 vrf MPN00001
  ip pool starent_ip_pool_v4_003 209.165.202.132 255.255.255.224 private 0 no-chunk-pool
  group-name starent_ipv4_pool_group_01 vrf MPN00001
  exit
exit'''

```


DELETE_ENTERPRISE_INPUT_PARAMETERS.txt

This input file must contain APN, SGI context, and VRF details when the request is to remove the enterprise.

Sample DELETE_ENTERPRISE_INPUT_PARAMETERS.txt

Following is the example configuration for deleting a single APN.

```
OpType= "DELETE_ENTERPRISE"

CP_APN_Config = '''config
    context APN
        no apn cisco.com
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGI_Context = '''config
    context dummy-SGI
no ip vrf MPN00001

    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config = '''config
    router bgp 65101
        no ip vrf MPN00001
    exit
exit'''
```

Following is the example configuration for deleting multiple virtual APNs in one DELETE operation.

```
CP_APN_Config = '''config
    context APN
        no apn virtual1
        no apn virtual2
        no apn virtual3
        no apn virtual4
        no apn virtual5
        no apn virtual6
        no apn virtual7
        no apn virtual8
        no apn virtual9
        no apn virtual10

        apn real1
            no virtual-apn preference 1
            no virtual-apn preference 2
            no virtual-apn preference 3
        exit
        apn real2
            no virtual-apn preference 3
            no virtual-apn preference 6
            no virtual-apn preference 9
        exit
        apn real3
            no virtual-apn preference 2
            no virtual-apn preference 5
            no virtual-apn preference 8
        exit
        apn real4
```

```

        no virtual-apn preference 2
        no virtual-apn preference 3
        no virtual-apn preference 5
    exit
    apn real5
        no virtual-apn preference 9
        no virtual-apn preference 8
        no virtual-apn preference 7
    exit
    apn real6
        no virtual-apn preference 13
        no virtual-apn preference 11
        no virtual-apn preference 12
    exit
    apn real7
        no virtual-apn preference 12
        no virtual-apn preference 13
    exit
    apn real8
        no virtual-apn preference 12
    exit
    apn real9
        no virtual-apn preference 19
        no virtual-apn preference 17
        no virtual-apn preference 13
        no virtual-apn preference 12
        no virtual-apn preference 15
        no virtual-apn preference 14
        no virtual-apn preference 16
        no virtual-apn preference 18
    exit
    apn real10
        no virtual-apn preference 1
    exit
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''config
    context dummy-SGi
        no ip vrf MPN00002

    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config = '''config
    router bgp 65101
        no ip vrf MPN00002
    exit
exit'''

```

System Limits

The following table depicts the maximum limits on ASR 5500 and CUPS.

Table 1: System Limits

Parameter	ASR 5500	Control Plane	User Plane
VRF Limit	300 per context 2048 per chassis	<ul style="list-style-type: none"> • 300 per context: Derived from the output of show ip user-plane verbose CLI command. • 1500 per chassis: Derived from the output of show ip user-plane verbose CLI command that is added across all contexts. 	205 VRF (with default routes); Derived from the output of show ip user-plane verbose CLI command. Must calculate per UP.
IP Pool Limit	IPv4: 2000 per context IPv6: 256 IPv6 per context 5000 per chassis (combined IPv4 and IPv6)	<p>IPv4: 2000 per context - Derived from the output of show ip user-plane verbose CLI command.</p> <p>IPv6: 256 IPv6 per context</p> <p>3400 per chassis (combined IPv4 and IPv6) - Derived from the output of show ip user-plane verbose CLI command.</p>	<p>Total of 600 IP pools per context per UP group:</p> <ul style="list-style-type: none"> • Total of 600 IP pools can consist a maximum of 256 IPv6 IP pools. • Total of 600 IP pools can consist a maximum of 600 IPv4 IP pools. <p>Derived from the output of show ip user-plane verbose CLI command. Must calculate the value from the output (Max 600 IPv4 pools, Max 256 IPv6 pools).</p>
APN Limit	2048	Total of 1500 for the system: Derived from the output of show cups-resource session summary CLI command.	205 per UP: Derived from the output of show cups-resource session summary CLI command. Must calculate per UP.

**Note**

- The IOB tool allows onboarding (OpType: ADD_ENTERPRISE) multiple APNs provided all those APNs share the "CP_SGi_Context" and "UP_VRF_Config" section of the input file. The APNs may potentially use different IP pool groups, but all those pool groups must be present in a single context in the "CP_SGi_Context" section of the input file. Also, the APNs must share a VRF. In such a case all those APNs onboard to the same UP group and SGi context.
- The tool supports deletion (OpType: DELETE_ENTERPRISE) of multiple APNs provided that all APNs share the "UP_VRF_Config" and "CP_SGi_Context" sections of the input configuration. The tool deletes the VRF and pools at the end of the operation. The intended use case for multiple APN deletes is to delete APNs that were onboarded together. The APNs onboarded together must be deleted together - the tool does not support separate deletion of APNs that were onboarded together.
- The tool does not support modification (OpType: MODIFY_ENTERPRISE) of multiple APNs in one operation. Only one APN can be modified at a time.
- The CUPSinfo.txt file is considered as the primary UP information. If any UP Groups are added in the system, but are not present in the file, then they are excluded from onboarding.

Enterprise Onboarding in CUPS OAM Support

This section describes operations, administration, and maintenance information for this feature.

Show Commands

show cups-resource session summary

This CLI command is introduced in support of the Enterprise Onboarding in CUPS solution. The output of this CLI command displays system-level resources on CP.

NOTES:

- Group Name Column displayed in output is the name of UP Group.
- Sx-IP shows the IP address of UP configured under the UP Group.
- APN, Active-Sessions, and LCI details are for the UP Group.

show ip user-plane verbose

The output of this CLI command is enhanced to display Total Pool Kernel Routes and Max Pool Kernel Routes fields. The dynamic IPv4 and IPv6 pool count is replaced with total IPv4 and IPv6 pool count. The output of this CLI command displays the context and UP Group it belongs to, and also adds information on number of IP pools and VRFs for that UP.

Error Codes

The following list of error codes is available in support of Enterprise Onboarding in CUPS feature.

Error Code	Description
1001	Indicates that the parsing of Input files has failed.
1002	Indicates that the parsing of Input_parameters file has failed.
1003	Indicates that the parsing of CUPSinfo file has failed.
1004	Indicates the inability to decrypt the passwords.
1005	Indicates that OpType is not present in input parameters.
1006	Indicates that the required configurations are not available in Input_parameters file for a given OpType.
1101	Indicates that the system pre-processing has failed.
1102	Indicates that the CPs pre-audit has failed for a given OpType.
1103	Indicates that the UPs pre-audit has failed for <UP_name>.
1107	Indicates that the tool is unable to update the CP_APN_Config section with the selected SGI context and UP Group. This indicates an error in the input configuration file.
1108	Indicates that the input file contains specified multiple APNs in a MODIFY_ENTERPRISE operation. This is not supported.
1301	Indicates that the CONTEXT and UPGROUP are not available for selection.
1401	Indicates the inability to find <context_name> and <group_name> from the CUPS system.
1501	Indicates the inability to get <context_name> from the output of show apn CLI command.
1601	Indicates that the configurations have failed for <control/user plane name> <connection state>.
1602	Indicates that the rollback configurations have failed for <control/user plane name> .
1701	Indicates that the CP post-audit has failed for <control plane name> <connection state>.
1702	Indicates that the UP post-audit has failed for <user plane name> <connection state>.
1703	Indicates that the Sx re-association has failed.

