Migration of a GMM customer

1. Do an assessment of the customer in GMM
   1. Check the current templates / advanced templated that are in use
   2. Check for app mgmt. / SRA and other features customer uses in GMM
2. Ask Deva to create the top level org with the same name that exists in GMM
3. If the customer has a org hierarchy in GMM, use the Python script orgs\_migration.py to create the organization structure (<https://cto-github.cisco.com/IOTNM/gmm-iotoc-migration>)
4. Now start creating the templates / groups in the new IoTOD orgs
   1. Identify with eCVD template to use as the initial template
   2. For 809s / 807s, use the 829 eCVD template and adjust the interfaces to match 809
   3. If there are additional configs in the GMM templates / advanced templates that are not covered by the eCVD, append all this additional config to the template in IoTOD.
   4. Keep the names of templates/groups in IoTOD same as the ones in GMM (Note: Special characters are not allowed in IoTOD names)
   5. Update the values as necessary in the form view of the group template based on the values in the GMM template.
5. For each of the orgs in IoTOD/GMM, run the devices\_migration.py ((<https://cto-github.cisco.com/IOTNM/gmm-iotoc-migration>)) to generate a csv file with the list of devices and all the config properties in GMM.
   1. Delete any columns that are not applicable for IoTOD or the ones that are already covered in the group level property of IoTOD.
   2. Update the column names if required to match the property names in IoTOD
   3. Use this csv file to bulk add the devices into IoTOD
   4. All devices should show up in Unheard state
6. In the GMM orgs, create the IoTOD migration template. You need to be admin of the GMM org to do this step. Either ask Praveen or customer to add you as admin.
   1. Create an advanced template of type Router with this eem script
   2. Create a template based on the model with the name starting with “IoTOD migration” which uses this Advanced Router template. This name will help GMM avoid unnecessary reboot while applying this template.

*event manager applet migrate*

*event timer countdown time 5 maxrun 300*

*action 1.0 cli command "enable"*

*! get the current bootimage with path*

*action 2.1 cli command "show ver | i System image file"*

*action 2.2 regexp "flash:.[^\"]\*" "$\_cli\_result" bootimage*

*! extract the boot image name*

*action 2.3 string replace "$bootimage" 5 5 "/"*

*action 2.4 regexp "[^/]\*$" $\_string\_result bootfile*

*! disable file prompt and create backup directory*

*action 3.0 cli command "conf t"*

*action 3.1 cli command "file prompt quiet"*

*action 3.2 cli command "end"*

*action 3.5 cli command "mkdir gmm\_backup"*

*! Find the GMM files / old images to backup*

*! Filename must not contain whitespace*

*! Some non-standard files are not covered, like user renamed image files*

*action 4.0 cli command "dir flash: | i ir8.\*mz|ir1101-universalk9|golden.bin|base.\*cfg|golden.cfg"*

*! flag to determine if bootimage is in root dir*

*action 4.01 set fileinroot "0"*

*action 4.1 foreach line "$\_cli\_result" "\n"*

*! trim the carriage return*

*action 4.2 string trimright "$line" "\r"*

*! skip the last line, which is a prompt*

*action 4.3 if $\_string\_result ne $line*

*action 4.4 regexp "[^ ]\*$" "$\_string\_result" file*

*! if it is the bootimage, set flag and skip move*

*action 4.5 if $file eq $bootfile*

*action 4.51 syslog msg "Keeping $bootfile already in flash:"*

*action 4.6 set fileinroot "1"*

*! otherwise, move it*

*action 4.7 else*

*action 4.71 syslog msg "moving $file to gmm\_backup/"*

*action 4.8 cli command "rename $file gmm\_backup/$file"*

*action 4.9 end*

*action 4.91 end*

*action 4.99 end*

*! if boot image not in root, copy it over*

*action 5.0 if $fileinroot ne "1"*

*action 5.1 syslog msg "Moving $bootimage to flash:$bootfile ..."*

*action 5.2 cli command "rename $bootimage flash:$bootfile"*

*action 5.4 end*

*! Sanity check. Boot image better be in root or bail out*

*action 8 cli command "dir flash:$bootfile"*

*action 8.1 string match "\* $bootfile\*" "$\_cli\_result"*

*action 8.2 if $\_string\_result ne "1"*

*action 8.3 syslog msg "bootimage '$bootfile' not found! Quiting..."*

*action 8.4 exit*

*action 8.5 end*

*action 8.6 syslog msg "Ready for Migration"*

*action 8.7 wait 10*

*! Finally, do erase and reload*

*action 9.0 cli command "show logging | redirect flash:migrate.log" pattern "confirm|#"*

*action 9.1 cli command ""*

*action 9.2 cli command "erase nvram:" pattern "confirm|#"*

*action 9.3 cli command ""*

*action 9.9 reload*

*end*

1. Add users to the IoTOD orgs
   1. Use the Python script users\_migration.py to add the users. This script will copy over all the users from GMM org to the IoTOD org – An admin in GMM will get Device Operator role and operator in GMM will get Restricted Operator role in IoTOD.
   2. Customer will now get the welcome email from IoTOD
2. Confirm that customer has setup the PnP connect with the right profile and added all the serial numbers to it.
3. If the customer is using Application Management in GMM, for each organizations in GMM use the GMM app migration script ([https://cto-github.cisco.com/IOTNM/gmm-iotoc-migration](https://cto-github.cisco.com/IOTNM/gmm-iotoc-migration/tree/app_migration_dev) ) to export all the necessary details regarding all the apps those are installed or running in GMM. This script would generate a tar.gz containing the details of the devices along with all the app config details, templates, policies and installation details which could then be used for reference and further steps in migration to IoTOD. For running the script, you should run the bellow command:

**python migrate.py export-gmm-app-details –-base-url <gmm-url> --org-id <org Id> --api-key <GMM-api-key>**

Example:

python .\migrate.py export-gmm-app-details --base-url=https://jokerdev.iotspdev.io/api/v2/ --org-id=2766 --api-key=1ee8a76a80ddf8a4.BTBZ5rabU9rOMyJekwVJ4PCaKLrjgSmF5WxnIvYZzp4

if the script successfully ran then you should see the log the message at the end like bellow:

2021-07-29 12:10:03,638 - App Migration:: - INFO - GMM Apps details has been successfully exported in C:\Debabrata\PythonDevelopments\iot\gmm-iotoc-migration\archive\gmm\_org\_2766\_20210729-121003.tar.gz

As mentioned in the output it will generate tar.gz file appending the current date and time with and its org id.

The exported tar file will have a folder called devices which contains <serial\_number.json> files and the migrate-gmm-app command in step 12 will use the installation values and app specific params from these files.

NOTE: If you want to change any app configuration before reinstalling onto IOD, then do the following?:

* Extract the tar.gz file
* Go to the devices folder, find the JSON file corresponding to the device that needs configuration change
* Modify the configuration (Eg: resource values)
* Save the files
* Create the tar.gz file back from the modified folder contents
* Use the modified tar.gz file as input during step 13

1. In GMM, uninstall the currently running applications from the set of devices that are selected for migration.
2. If the customer wants to re-install the applications that were present in GMM
   1. Upload the necessary application packages into the appropriate IoTOD tenant corresponding to the GMM organization. This is the mandatory step before running the app migration script in IoTOD (step 13).
3. In GMM, apply the IoTOD migration template to the test gateways.
   1. Select (checkbox) the gateway ready for migration
   2. Click Networking -> Change template
   3. Click on Saved templates tab
   4. Select the “IoTOD migration…” template and click save and update gateway in confirmation dialog.
   5. This will apply the eem script on the gateway.
   6. EEM script will backup the base GMM config in case a manual roll back needs to be done.
   7. All existing config will be erased and device will be reloaded.
   8. Any private APN that was configured on the device will NOT be lost in this process. So, cellular connectivity will work fine during PnP process.

Once PnP kicks in, it is regular on-boarding process in IoTOD.

When the device comes up successfully, customer can validate the device to ensure that all required features (subnet config, site-to-site VPN, users etc.) work as expected.

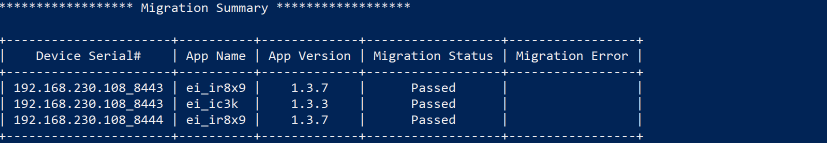
1. If the customer wants to re-install the applications that were present in GMM, follow the below step (Note: This will not migrate the data/state of the application from GMM, but will freshly install the application with the same configuration that existed in GMM)
   1. Now for each organization use the tar.gz file generated in step 9 to run IoTOD app migration script. The script needs a csv file as input. Create a csv file containing all the serial numbers of devices for which the applications need to be re-installed (Even though the exported file contains information about all devices, this CSV file can be used to migrate the devices in subsets/batches).

Run the bellow command for app migration:

migrate.py install-gmm-app-to-iod [OPTIONS] GMM\_EXPORT\_TAR

Example:

python .\migrate.py install-gmm-app-to-iod --device-file=device\_file\_test.csv .\archive\gmm\_org\_2414.tar.gz



For details about running the script, follow the README here <https://cto-github.cisco.com/IOTNM/gmm-iotoc-migration/>

Once script is successfully run, check that the applications those have been re-installed are running with proper configuration in the corresponding device.

Potential issues that can come up during migration

1. The device needs special config to get internet connectivity – In this case, the eem script that erases all config will not work and an alternate solution will need to be figured out.
   1. For Penske, we had to create a config file that was needed to setup internet connectivity. Appended the pnp\_profile config to this file.
   2. Copied this config file to the device using GMM/FND.
   3. Modified the eem script to copy this config file to startup-config and reload instead of erasing all config.
2. There could be platform/pnp issues with the ios version currently running on the devices.
   1. Seen this with COTA and LADWP (issues were different but PnP would not work)
   2. Do a firmware upgrade to the latest version from GMM before kicking off the migration.
   3. If the required version is not available in the GMM org, Praveen can enable it.
3. There could be IoTOD defects that you run into during migration
   1. Work with IoTOD team to see if workarounds are available
   2. Raise blocker defects if needed
4. There could be eCVD defects that you run into during migration
   1. Work with eCVD team to resolve the configuration issues
5. If the application stores data under /data directory or /data/appdata, the script will not migrate this data, but only will re-install the application using the GMM configuration. So, re-installing these applications using the migration script may not work (the exact behavior will depends on how the application uses the stored data)
6. If application using secure storage, then the script will not migrate the data stored in secure storage. So, re-installing these applications using the migration script may not work (the exact behavior will depends on how the application uses secure storage).