Retrieving OS-repository data for OS upgrades

The OS repository stores all files that are needed for OS upgrade. You can manage these files through the OS repository form, and categorize them by image, device type and vendor type. Each vendor type has its own directory within the */var/opt/yce/os* directory, in which you can find the file.

If you want to perform an upgrade on a node, the question now is how to get these files from the OS repository to the node. There are multiple ways to do this.

The os_files scenario command

The fastest way to the files is by using the os_files scenario command. You just supply the vendor type and device type for your node, and if it's all configured correctly then it should return the os file that you need, or even the os files that you need in the case that you need multiple of them.

Let's take a very basic example. We have an Comware 7 node of the HP5920 model, and we want to transfer the os image currently set as production in the OS repository. All you have to do is create a command job as follows.

```
<os_files> := os_files -v <vendor_type> -t "HP5920"
<os_src> = "/var/opt/shared/public/os"

foreach <os_file> in <os_files>
    log -m "put os-image-file: <os_file>"
    file_put -n <node> -s "<os_src>/<vendor_type>/<os_file>" -t "/"
endeach
```

If you are certain that the OS upgrade only takes one file, and no more, you can even shorten it to this:

```
<os_file> := os_files -v <vendor_type> -t "HP5920"
log -m "os-image file is: <os_file>"

<source> := "/var/opt/shared/public/os/<vendor_type>/<os_file>"
file_put -n <node> -s "<source>" -t "/"
```

Advanced calls

If you want to add a bit more validation, you can combine this with the os_file_select scenario command:

```
<os_files> := os_files -v <vendor_type> -t "HP5920"

foreach <os_file> in <os_files>
```

```
    log -m "file <os_file>"

    <%os_file> := os_file_select -v <vendor_type> -n <os_file>
    log -m "File md5: <File_md5%os_file>
    log -m "File checksum: <File_crc%os_file>
    log -m "File size: <File_size%os_file>
    log -m "File minimum storage space required: <Storage_min%os_file>


    <source> := "/var/opt/shared/public/os/<vendor_type>/<os_file>"

    file_put -n <node> -s "<source>" -t "/"
endeach
```

You can now validate these values against the values you find on the node (for example, retrieve then through command parsing.

The Os_upgrade_file Relation

Another easy way to retrieve the files for an OS upgrade is by making use of the Os_upgrade_file relation. This is a standard relation, but it can be tweaked and modified as desired, and it collects the necessary data for the vendor type and device type that you specify.

To go back to our example of a Comware 7 node with model HP5920, you will need the following scenario code:

```
log -m "Os file name: <Os_file_name@Os_upgrade_file:device_type=HP5920>"

<source> :=
"/var/opt/shared/public/os/<vendor_type>/<Os_file_name@Os_upgrade_file:devic
e_type=HP5920>"

file_put -n <node> -s "<source>" -t "/"
```

The os_image_select and os_file_select scenario commands

If you want to have a few more options, you can use a combination of the os_image_select and the os_file_select scenario calls. os_image_select selects the os image you need based on a number of criteria, and os_file_select selects its os_file. This is a solution that works for OS images that have just one file.

```
<%os_image> := os_image_select -v <vendor_type> -t "HP5920"
if <error>
    stop
endif

log -m "Found os image <Os_name%os_image>"
```

```
<%os_file> := os_file_select -i <Id%os_image>
if <error>
    stop
endif

log -m "Found file <Os_file_name%os_file>"
log -m "File md5: <File_md5%os_file>
log -m "File checksum: <File_crc%os_file>
log -m "File size: <File_size%os_file>
log -m "File minimum storage space required: <Storage_min%os_file>

<source> := "/var/opt/shared/public/os/<vendor_type>/<Os_file_name%os_file>"

file_put -n <node> -s "<source>" -t "/"
```

All attributes of an os image and file can be references through these hash variables, which allows for validation. Also note that when no os image is found, the <error> variable is set. And by default, OS images that have the status of “production” will be selected. If you want to change that, you can use the -s option. Only os files with the status “active” will be returned. If they are not, then something is wrong with the repository and that will need to be fixed outside of the command job.

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:user:scenario_calls

Last update: **2024/07/03 12:31**