Automate AIX updates and patch management in NIM environment with ‘Puppet AIX and VIOS Automation’ module

Oct 26th,2018

Updating IBM® AIX® system *in a large-scale infrastructure is now possible with Puppet®.*
Setting attributes in a Puppet manifest file enables automatic updates of AIX NIM standalone clients in push mode. Specific NIM resources built automatically from SUMA downloads or built following FLTRVC recommendations can be used.

Introduction

This article details how to use Puppet to automate AIX updates and patch management in NIM environment.
This service is delivered as a Puppet module called ‘**Puppet AIX and VIOS Automation’** module. This module enables to automatically upgrade NIM standalone clients to latest or specific SP (Service Pack) or TL (Technology Level) and/or apply recommended fixes: security (SEC) and high impact pervasive (HIPER) fixes.

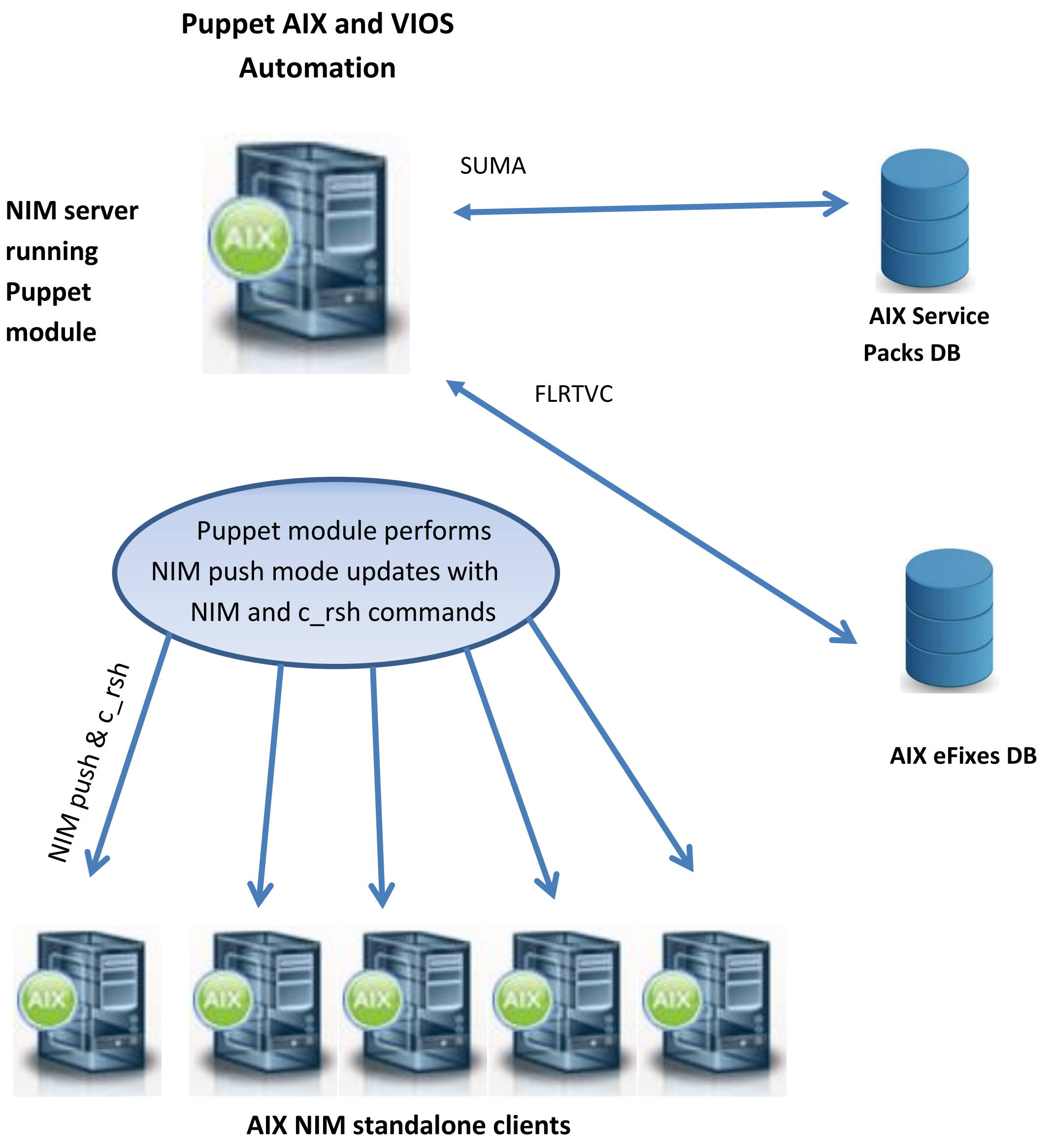
Another article called “Automate VIOS updates in NUM environment with ‘Puppet AIX and VIOS Automation’ module” completes the present documentation as far as VIOS updates are concerned.

Main components involved in ‘**Puppet AIX and VIOS Automation’** module are described first, and general explanations are provided. Functionalities offered by this module are then listed. Configurations steps of NIM, Puppet, and ‘Puppet AIX and VIOS Automation’ module are provided, and a step is provided to verify installation is correct. Then starts the internal description of the module and how to use it. If syntax of new ‘download’, ‘patchmngt’ and ‘fix’ custom types is not repeated as already given in README.md, several examples are provided to show how to individually run these steps, or how to combine them to perform AIX updates and patch management.

See table of contents at the end.

Main components

This service is delivered as a Puppet module called ‘**Puppet AIX and VIOS Automation’** module.


Puppet AIX and VIOS Automation
NIM server running Puppet module
AIX
SUMA
AIX Service Packs DB
FLRTVC
AIX eFixes DB
Puppet module performs NIM push mode updates with NIM and c_rsh commands
NIM push & c_rsh
AIX NIM standalone clients

The diagram above describes the main components involved by AIX patch management with Puppet.

At the bottom are the AIX NIM standalone clients: they are the targets of the updates performed by 'Puppet AIX and VIOS Automation' module. They can be automatically updated and patched on an individual basis.

In the middle left is the NIM server, on which Puppet Agent needs to be installed. 'Puppet AIX and VIOS Automation' module runs on this instance of Puppet. Connections between NIM server and NIM standalone clients is exclusively performed through NIM's nimsh protocol, (c_rsh is used, but no ssh required): updates are performed from NIM server to NIM client by using NIM push mode.

In the middle right are two external databases:

- the AIX Service Packs DB (IBM Fix Central) accessed through SUMA (Service Update Management Assistant) command running on the NIM server,
- the AIX eFixes DB accessed through http/https/ftp protocols to apply FLRTVC (Fix Level Recommendation Tool Vulnerability Checker) recommendations.

The NIM server needs to be able to connect to these external DB.

'Puppet AIX and VIOS Automation' module enables to upgrade the NIM clients to latest or specific SP (Service Pack) or TL (Technology Level) and/or apply recommended fixes to these NIM clients. The NIM server must be able to connect to IBM Fix Central to download SPs or TLs through **SUMA** (Service Update Management Assistant). **NIM** (either by NFS or HTTP) is used from NIM server to NIM clients to perform updates and to apply recommended **FLRTVC** (Fix Level Recommendation Tool Vulnerability Checker) patches: security (SEC) and high impact pervasive (HIPER) fixes.

Software component

This service for managing AIX system updates with Puppet® is an Open Source development available on **AIXOSS GitHub repository**. It is added to the existing AIX systems update offers built on Ansible® and Chef® platforms.

Functionalities offered by 'Puppet AIX and VIOS Automation' module

'Puppet AIX and VIOS Automation' module enables update of AIX system on standalone logical partitions. This update in a large-scale infrastructure is orchestrated through 'Puppet AIX and VIOS Automation' module and rely on the underlayer on functionalities provided by NIM server.

By 'update', we mean updates of AIX levels and updates of AIX eFixes from FLRTVC alerts.

'Puppet AIX and VIOS Automation' module offers the possibility to automate:

- The download of AIX Technology Levels (TL's) and Service Packs (SP's) from AIX fix server by using SUMA commands.

- The download of recommended security and High Impact Pervasive AIX fixes through FLRTVC tool.
- The creation of NIM lpp_source resources from downloaded update files (.bff from SUMA, epkg.Z from FLRTVC).
- The installation of NIM lpp_source resources from resources automatically created for the update or from a specific resource defined by the user as ssh, rpm…
- The removing of eFixes.

'Puppet AIX and VIOS Automation' module automatically install FLRTVC tools, if this tool is not yet installed on the NIM server

With 'Puppet AIX and VIOS Automation' module, the updates can be **applied**, **committed**, or **rejected**.

With 'Puppet AIX and VIOS Automation' module, NIM, SUMA and FLRTVC commands are totally hidden to the end user.

The unique entry point for the end user to declare its resource and the desired state of these resources is located in the manifest file: **./aixautomation/manifests/init.pp**

In order to write customized **./aixautomation/manifests/init.pp**, the **README.md** file provides a full reference for the three custom-types : 'download', 'patchmngt' and 'fix', and their attributes. Semantics of attributes is provided, as well as details about values: mandatory or optional values, possible values, default values

In addition to this README.md file, the user can refer to commented samples provided into './examples/init.pp'.

For an update with Puppet AIX Automation, the user must define specific properties depending on the custom type he/she wants to use, some of them are explained below as an introduction:

- Levels of update requested to download: this is specified
  - For SUMA through two parameters:
    - 'from' (meaning from a specific AIX TL or SP): identifies the current level,
    - 'to' (meaning to a specific AIX TL or SP): identifies as level the level desired.
  - For FLRTVC thru the 'level' parameter indicating the type of eFix:
    - 'hiper' for HIPER eFixes,
    - 'sec' for Security eFixes,

    - ‘all’.
- Local directory to store the results of downloads from AIX Fix central.
- NIM lpp_source resource to create or to use in a NIM operation.
- List of AIX systems on which to perform actions, these are called ‘targets’.
- Type of the NIM operation: install, update, remove
- …

Several resources can be declared, using DSL language, in the same manifest file (./manifests/init.pp), so that several actions are triggered one after the other. The output of an action can be the input of the following one: typically, the ‘download’ resource declaration can create a NIM lpp_source resource which is used later by a ‘patchmngt’ resource declaration. As far as ‘fix’ resource declaration is concerned, it can perform download of resources, creation of NIM lpp_source resource, and installation of this NIM resource on the ‘targets’.

The functional flow of ‘Puppet AIX and VIOS Automation’ module update consists of:

(i) Download requested update files from AIX Fix central,

(ii) Check if the update is applicable for the defined system targets,

(iii) Create the appropriate lpp_source resources,

(iv) Install the lpp_source resources through NIM operations.

Configuration

NIM configuration

NIM configuration between NIM server and NIM clients

The NIM server needs to be at a level at least as high as the highest level of its clients. It can address several clients with different AIX releases and levels.

The NIM master connection with the AIX standalone client partitions must be set with “nimsh” and “c_rsh” connection must be available.

Below the commands to run on clients to activate nimsh connection:
(“quimby01” = name of AIX partition standalone client and “fattony01” name of NIM master).

```
# mv /etc/niminfo /etc/niminfo.last
# niminit -aname=quimby01 -apif_name=en0 -amaster=fattony01 -aplatform=chrp -acable_type1=N/A
-a connect=nimsh
# nimclient –C
```

To verify the connection, run a c_rsh command from the NIM master, for example:

```
 #/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/usr/bin/oslevel -s"
7100-03-07-1614
```

NIM configuration between NIM server and external DB

The NIM master must have access to the internet to download fixes and updates through HTTP and FTP protocol.

Puppet configuration

Puppet Agent installation

Installation of Puppet Agent needs to be done on the server hosting the NIM server.

- Get Puppet Agent 5.3.5 from: https://s3.amazonaws.com/puppet-agents/2017.3/puppet-agent/5.3.5/repos/aix/7.1/PC1/ppc/puppet-agent-5.3.5-1.aix7.1.ppc.rpm

- Install Puppet Agent package with rpm tool:

```
# rpm -hiv puppet-agent-5.3.5-1.aix7.1.ppc.rpm
Preparing...        ############################################ [100%]
1:puppet-agent   ############################################ [100%]
```

Verification of Puppet Agent installation

The files installed with Puppet Agent can be shown with:

```
# rpm -qa | grep puppet | xargs rpm -ql
/etc/puppetlabs
/etc/puppetlabs/code
/etc/puppetlabs/code/environments
/etc/puppetlabs/code/environments/production
/etc/puppetlabs/code/environments/production/data
/etc/puppetlabs/code/environments/production/environment.conf
/etc/puppetlabs/code/environments/production/hiera.yaml
...
/etc/puppetlabs/mcollective/rpc-help.erb
/etc/puppetlabs/mcollective/server.cfg
/etc/puppetlabs/puppet
/etc/puppetlabs/puppet/auth.conf
/etc/puppetlabs/puppet/hiera.yaml
/etc/puppetlabs/puppet/puppet.conf
/etc/puppetlabs/pxp-agent
```

```
/etc/puppetlabs/pxp-agent/modules
/opt/freeware/doc/puppet-agent-5.3.5
/opt/freeware/doc/puppet-agent-5.3.5/bill-of-materials
/opt/puppetlabs
/opt/puppetlabs/bin
/opt/puppetlabs/bin/facter
/opt/puppetlabs/bin/hiera
...
```

Puppet Agent installation impacts 3 directories:

/etc/puppetlabs
/opt/freeware/doc/puppet-agent-5.3.5
/opt/puppetlabs

Puppet binaries are located in the directory: **/opt/puppetlabs/bin**

Check Puppet version:

```
# /opt/puppetlabs/bin/puppet –version
5.3.5
```

Puppet comes with its own ruby (therefore no ruby installation is required):

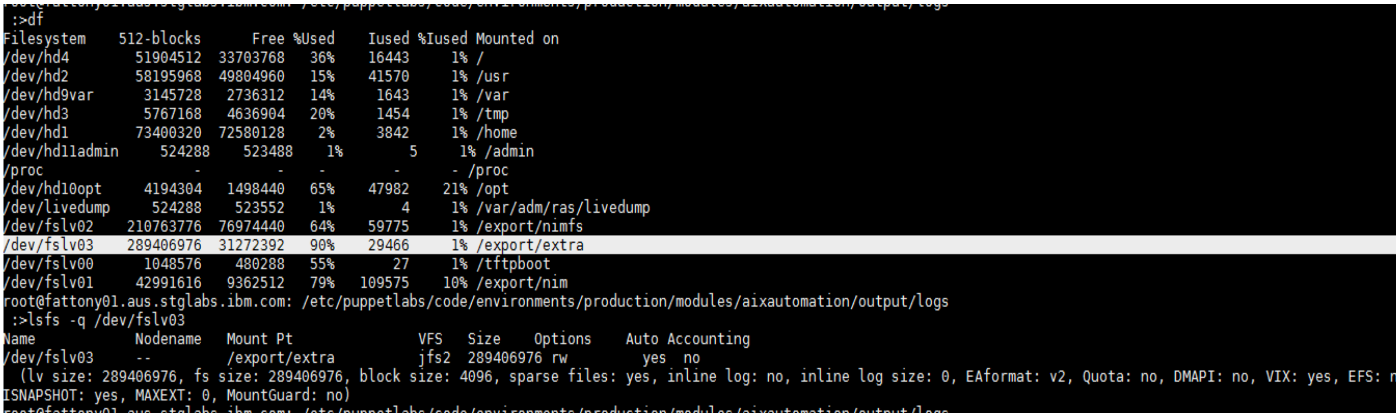
```
#  /opt/puppetlabs/puppet/bin/ruby --version
ruby 2.4.3p205 (2017-12-14 revision 61247) [powerpc-aix7.1.0.0]
```

‘Puppet AIX and VIOS Automation’ module configuration

‘Puppet AIX and VIOS Automation’ module requires disk space to perform downloads

Puppet AIX Automation patch management module requires enough available disk space to store the updates downloaded from AIX Fix Central, and to store downloaded eFixes. By default, downloads are performed into '/tmp’, but it is recommended to use a dedicated exportable (jfs or jfs2) file system with 30 GB or more, to store all the files needed for the updates and uploaded automatically by Puppet AIX Automation patch management module.

Below for example, the characteristics of the file system used for our test to store the files to download from IBM Fix central: jfs2 filesystem mounted on **/export/extra**



```
 :>df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4        51904512  33703768   36%    16443     1% /
/dev/hd2        58195968  49804960   15%    41570     1% /usr
/dev/hd9var      3145728   2736312   14%     1643     1% /var
/dev/hd3         5767168   4636904   20%     1454     1% /tmp
/dev/hd1        73400320  72580128    2%     3842     1% /home
/dev/hd11admin      524288    523488    1%        5     1% /admin
/proc                  -         -    -         -     - /proc
/dev/hd10opt     4194304   1498440   65%    47982    21% /opt
/dev/livedump     524288    523552    1%        4     1% /var/adm/ras/livedump
/dev/fslv02    210763776  76974440   64%    59775     1% /export/nimfs
/dev/fslv03    289406976  31272392   90%    29466     1% /export/extra
/dev/fslv00      1048576    480288   55%       27     1% /tftpboot
/dev/fslv01     42991616   9362512   79%   109575    10% /export/nim
root@fattony01.aus.stglabs.ibm.com: /etc/puppetlabs/code/environments/production/modules/aixautomation/output/logs
 :>lsfs -q /dev/fslv03
Name            Nodename   Mount Pt               VFS   Size    Options    Auto Accounting
/dev/fslv03     --         /export/extra          jfs2  289406976 rw         yes  no
  (lv size: 289406976, fs size: 289406976, block size: 4096, sparse files: yes, inline log: no, inline log size: 0, EAformat: v2, Quota: no, DMAPI: no, VIX: yes, EFS: n
ISNAPSHOT: yes, MAXEXT: 0, MountGuard: no)
```

'Puppet AIX and VIOS Automation' module installation

'Puppet AIX and VIOS Automation' module is available on **AIXOSS GitHub repository**.
The **AIXOSS GitHub repository** contains Open Source Software ported to AIX. It also contains scripts to use with Open Source Software to perform specific AIX tasks.

Get 'Puppet AIX and VIOS Automation' module from AIXOSS GitHub repository: https://github/aixoss/aix-puppet

by using following command:

```
# git clone https://github.com/aixoss/aix-puppet.git aixautomation
```

or by downloading directly the zip file.

Copy AIX and VIOS Automation to **/etc/puppetlabs/code/environments/production/modules/** repository. The name of the directory in which 'Puppet AIX and VIOS Automation' module is installed needs to be: '**aixautomation**'. If you get the module from zip file, you must rename '**aix-puppet-master**' to '**aixautomation**'.

Find below commands and outputs to perform this installation:

```
# cd /etc/puppetlabs/code/environments/production/modules
# git clone https://github.com/aixoss/aix-puppet.git aixautomation
Cloning into 'aixautomation'...
remote: Counting objects: 574, done.
remote: Compressing objects: 100% (164/164), done.
Receiving objects:  80% (460/574)remote: Total 574 (delta 285), reused 574 (delta 285), pack-reused 0
Receiving objects: 100% (574/574), 154.63 KiB | 0 bytes/s, done.
Resolving deltas: 100% (285/285), done.
Checking connectivity... done.
# ls -l
total 8
drwxr-xr-x    8 root     system         4096 Aug 06 04:02 aixautomation/
```

Verification of ‘Puppet AIX and VIOS Automation’module installation

The Puppet manifest file contains declaration resources in a Domain Specific Language (or DSL), and is the only entry point to use AIX and VIOS Automation module. The structure, the syntax, the semantic of the manifest file is detailed below, and several examples are provided hereafter.

‘Puppet AIX and VIOS Automation’ module contains a manifest file ‘./manifests/init.pp’ that can be immediately run to verify the installation and configuration of AIX Puppet, and the installation of AIX and VIOS Automation.

This minimal ‘./manifests/init.pp’ checks which packages need to be downloaded to update an AIX system from Technology Level "7200-01" to Service Pack "7200-01-03-1720". It performs no download, and no update. This very first operation of a SUMA command in preview mode is at no risk!

The manifest file “./manifests/init.pp” is displayed below:

```
# How to launch:
#  /opt/puppetlabs/bin/puppet apply \
#  --debug --modulepath=/etc/puppetlabs/code/environments/production/modules/ -e "include
aixautomation"
# This will trigger this suma command:
#  /usr/sbin/suma -x  -a RqType=SP -a RqName=7200-01-03-1720 -a FilterML=7200-01-02-1717 \
#   -a DisplayName="Downloading lppsources into /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-
1720" \
#   -a Action=Preview -a DLTarget=/tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720 \
#   -a FilterDir=/tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720
class aixautomation {
  download { "test suma-preview":
    ensure  => present,
    type       => "SP",
    # /tmp should be changed to perform a ‘download’ to a more appropriate directory
    #   in dedicated file system.
    # It is not necessary to change it to perform a ‘preview’.
    root     => "/tmp",
    from     => "7200-01-02-1717",
    to         => "7200-01-03-1720",
    to_step => "preview",
  }
}
```

Structure of ‘Puppet AIX and VIOS Automation’ module

Below is described the structure of ‘Puppet AIX and VIOS Automation’ module source tree, that can be found after installation into:

/etc/puppetlabs/code/environments/production/modules/aixautomation

‘Puppet AIX and VIOS Automation’ module implements facters that are necessary to gather facts before running anything:

- ‘applied_manifest’: to get, display and parse manifest files
- ‘props’: to set global properties
- ‘servicepacks’: to get SUMA metadata and generate a local database with SP available per TL
- ‘standalones’: to get data on NIM standalone clients and verify they can be used.

‘Puppet AIX and VIOS Automation’ module implements three custom types and their service providers:

- ‘download’ custom type implemented by the ‘suma’ provider based on SUMA services,
- ‘fix’ custom type implemented by the ‘flrtvc’ provider based on FLRTVC services,
- ‘patchmngt’ custom type implemented by the ‘nimpush’ provider based on NIM push.

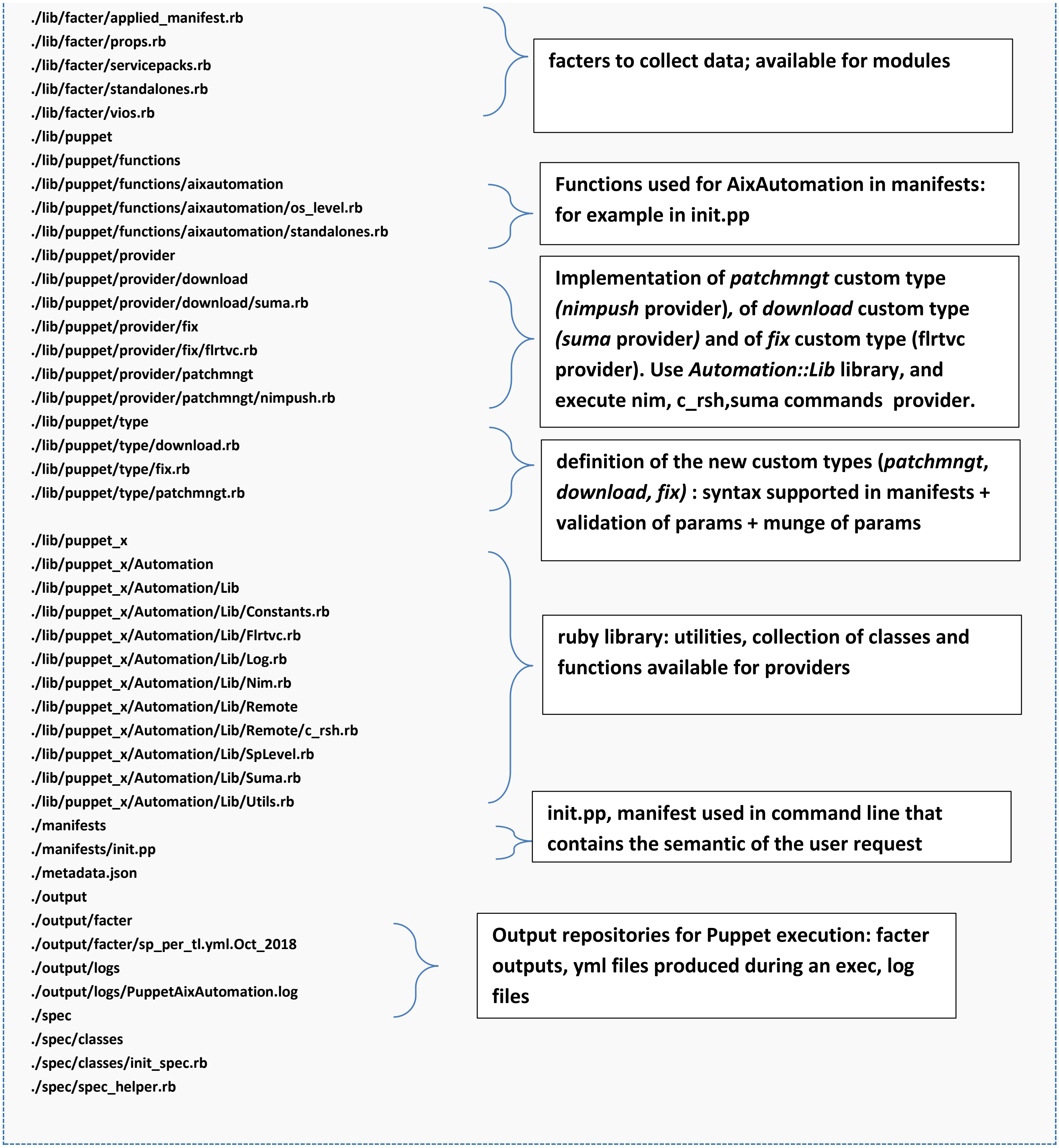
‘Puppet AIX and VIOS Automation’ module provides a library:

- Automation::Lib

‘Puppet AIX and VIOS Automation’ module provides samples of manifest files and documentation

- ./examples/init.pp
- ./manifests/init.pp
- README.md
- TODO.md

```
# cd /etc/puppetlabs/code/environments/production/modules/aixautomation
# cd find .
./Gemfile
./README.md
./Rakefile
./TODO.md
./examples
./examples/init.pp
./lib
./lib/facter
```



```
./lib/facter/applied_manifest.rb
./lib/facter/props.rb
./lib/facter/servicepacks.rb
./lib/facter/standalones.rb
./lib/facter/vios.rb
./lib/puppet
./lib/puppet/functions
./lib/puppet/functions/aixautomation
./lib/puppet/functions/aixautomation/os_level.rb
./lib/puppet/functions/aixautomation/standalones.rb
./lib/puppet/provider
./lib/puppet/provider/download
./lib/puppet/provider/download/suma.rb
./lib/puppet/provider/fix
./lib/puppet/provider/fix/flrtvc.rb
./lib/puppet/provider/patchmngt
./lib/puppet/provider/patchmngt/nimpush.rb
./lib/puppet/type
./lib/puppet/type/download.rb
./lib/puppet/type/fix.rb
./lib/puppet/type/patchmngt.rb

./lib/puppet_x
./lib/puppet_x/Automation
./lib/puppet_x/Automation/Lib
./lib/puppet_x/Automation/Lib/Constants.rb
./lib/puppet_x/Automation/Lib/Flrtvc.rb
./lib/puppet_x/Automation/Lib/Log.rb
./lib/puppet_x/Automation/Lib/Nim.rb
./lib/puppet_x/Automation/Lib/Remote
./lib/puppet_x/Automation/Lib/Remote/c_rsh.rb
./lib/puppet_x/Automation/Lib/SpLevel.rb
./lib/puppet_x/Automation/Lib/Suma.rb
./lib/puppet_x/Automation/Lib/Utils.rb
./manifests
./manifests/init.pp
./metadata.json
./output
./output/facter
./output/facter/sp_per_tl.yml.Oct_2018
./output/logs
./output/logs/PuppetAixAutomation.log
./spec
./spec/classes
./spec/classes/init_spec.rb
./spec/spec_helper.rb
```

facters to collect data; available for modules

Functions used for AixAutomation in manifests: for example in init.pp

Implementation of *patchmngt* custom type *(nimpush* provider)*,* of *download* custom type *(suma* provider*)* and of *fix* custom type (flrtvc provider). Use *Automation::Lib* library, and execute nim, c_rsh,suma commands provider.

definition of the new custom types (*patchmngt, download, fix)* : syntax supported in manifests + validation of params + munge of params

ruby library: utilities, collection of classes and functions available for providers

init.pp, manifest used in command line that contains the semantic of the user request

Output repositories for Puppet execution: facter outputs, yml files produced during an exec, log files

Run Puppet AIX Automation

How to launch

The user declarations are located in the manifest file init.pp:

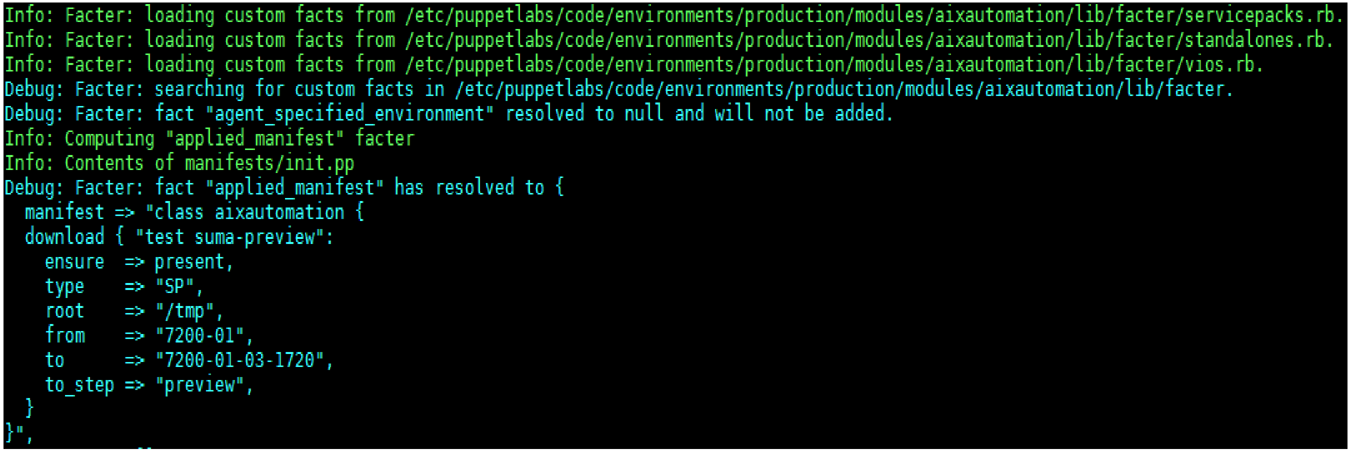
./aixautomation/manifests/init.pp

On NIM master, Puppet AIX Automation is launched with the following command:

```
# cd /etc/puppetlabs/code/environments/production/modules
#/opt/puppetlabs/bin/puppet apply --debug --
modulepath=/etc/puppetlabs/code/environments/production/modules/ -e "include aixautomation"
clude aixautomation"
```

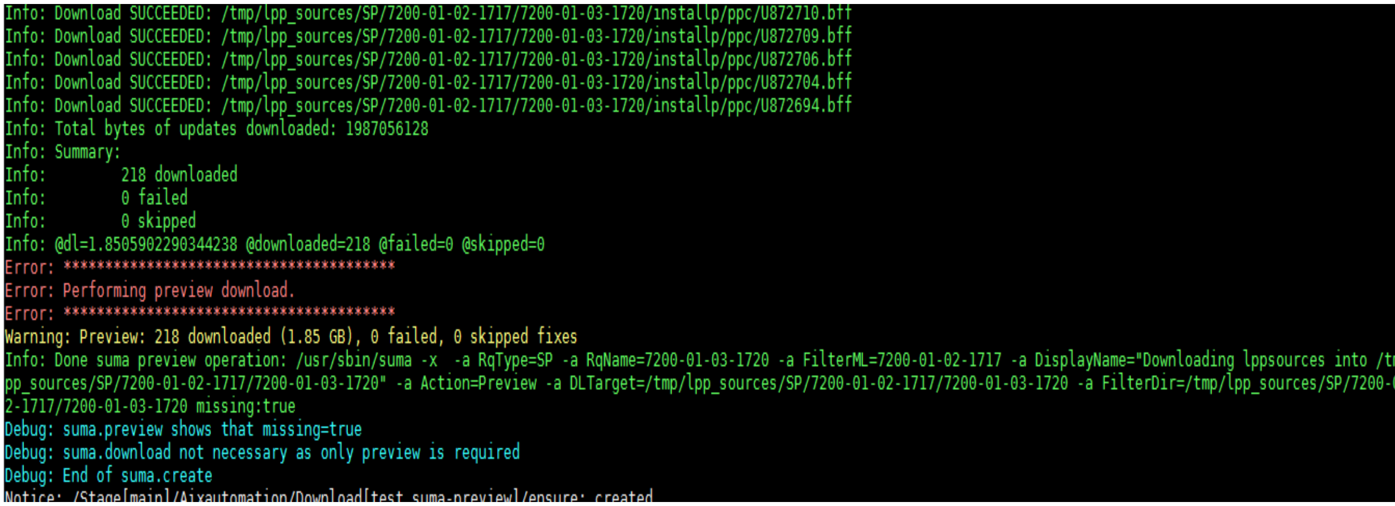
The standard output shows:

- the manifest file being run:



```
Info: Facter: loading custom facts from /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter/servicepacks.rb.
Info: Facter: loading custom facts from /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter/standalones.rb.
Info: Facter: loading custom facts from /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter/vios.rb.
Debug: Facter: searching for custom facts in /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter.
Debug: Facter: fact "agent_specified_environment" resolved to null and will not be added.
Info: Computing "applied_manifest" facter
Info: Contents of manifests/init.pp
Debug: Facter: fact "applied_manifest" has resolved to {
  manifest => "class aixautomation {
  download { "test suma-preview":
    ensure  => present,
    type    => "SP",
    root    => "/tmp",
    from    => "7200-01",
    to      => "7200-01-03-1720",
    to_step => "preview",
  }
}",
```

- the result of the SUMA downloads in preview mode:



```
Info: Download SUCCEEDED: /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720/installp/ppc/U872710.bff
Info: Download SUCCEEDED: /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720/installp/ppc/U872709.bff
Info: Download SUCCEEDED: /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720/installp/ppc/U872706.bff
Info: Download SUCCEEDED: /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720/installp/ppc/U872704.bff
Info: Download SUCCEEDED: /tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720/installp/ppc/U872694.bff
Info: Total bytes of updates downloaded: 1987056128
Info: Summary:
Info:         218 downloaded
Info:         0 failed
Info:         0 skipped
Info: @dl=1.8505902290344238 @downloaded=218 @failed=0 @skipped=0
Error: ****************************************
Error: Performing preview download.
Error: ****************************************
Warning: Preview: 218 downloaded (1.85 GB), 0 failed, 0 skipped fixes
Info: Done suma preview operation: /usr/sbin/suma -x  -a RqType=SP -a RqName=7200-01-03-1720 -a FilterML=7200-01-02-1717 -a DisplayName="Downloading lppsources into /tr
pp_sources/SP/7200-01-02-1717/7200-01-03-1720" -a Action=Preview -a DLTarget=/tmp/lpp_sources/SP/7200-01-02-1717/7200-01-03-1720 -a FilterDir=/tmp/lpp_sources/SP/7200-0
2-1717/7200-01-03-1720 missing:true
Debug: suma.preview shows that missing=true
Debug: suma.download not necessary as only preview is required
Debug: End of suma.create
Notice: /Stage[main]/Aixautomation/Download[test_suma-preview]/ensure: created
```

Remark: The "Error: Performing preview download" message is an output from SUMA on stderr, so we catch it as error, but it is just an information message that you can discard.

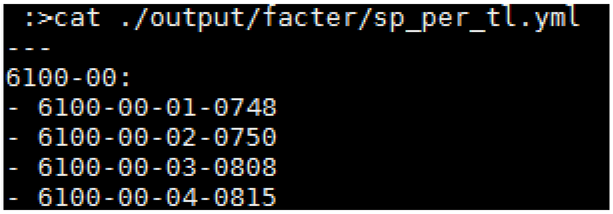
The log file of 'Puppet AIX and VIOS Automation' module is located in: **/etc/puppetlabs/code/environments/production/modules/aixautomation/output/logs/PuppetAixAutomation.log**

Explanation of what is triggered during the first run

Now are provided explanation of what is triggered during the first run (or if sp_per_tl.yml is removed): download of SUMA metadata, parsing of these metadata and generation of a local database with all Service Packs available per Technical Level.

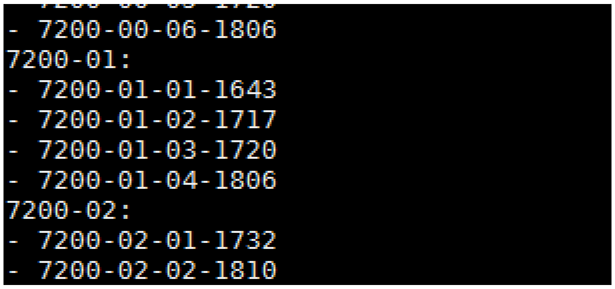
At the first run, Puppet AIX Automation builds an yml file that contains all the AIX Service Pack numbers sorted by AIX Technical Level. This file is built by iteration with SUMA metadata requests.

⇨ **aixautomation/output/facter/sp_per_tl.yml**



```
 :>cat ./output/facter/sp_per_tl.yml
---
6100-00:
- 6100-00-01-0748
- 6100-00-02-0750
- 6100-00-03-0808
- 6100-00-04-0815
```

………



```
- 7200-00-06-1806
7200-01:
- 7200-01-01-1643
- 7200-01-02-1717
- 7200-01-03-1720
- 7200-01-04-1806
7200-02:
- 7200-02-01-1732
- 7200-02-02-1810
```

This reference file of Service Packs available per Technical Level is used during validation of manifest files written by customer to verify attributes of 'download' resource declaration.

Note: a reference **sp_per_tl** file is provided with the source:

./output/facter/sp_per_tl.yml.Oct_2018

If you want to skip the step of building './output/facter/sp_per_tl.yml', or if you encounter problems to generate it, you can rename **sp_per_tl.yml.Oct_2018** file to **sp_per_tl.yml** before running 'Puppet AIX and VIOS Automation' module.

```
# cd /etc/puppetlabs/code/environments/production/modules/aixautomation/
# mv output/facter/sp_per_tl.yml.Oct_2018 output/facter/sp_per_tl.yml
```

If you want the last reference of Service packs from SUMA data base, let Puppet AIX Automation build the file sp_per_tl.yml on the first run, or remove it to recreate a new sp_per_tl reference with the next run.

Some explanations related to the runtime steps

‘Puppet AIX and VIOS Automation’ module runs the following steps during the update:

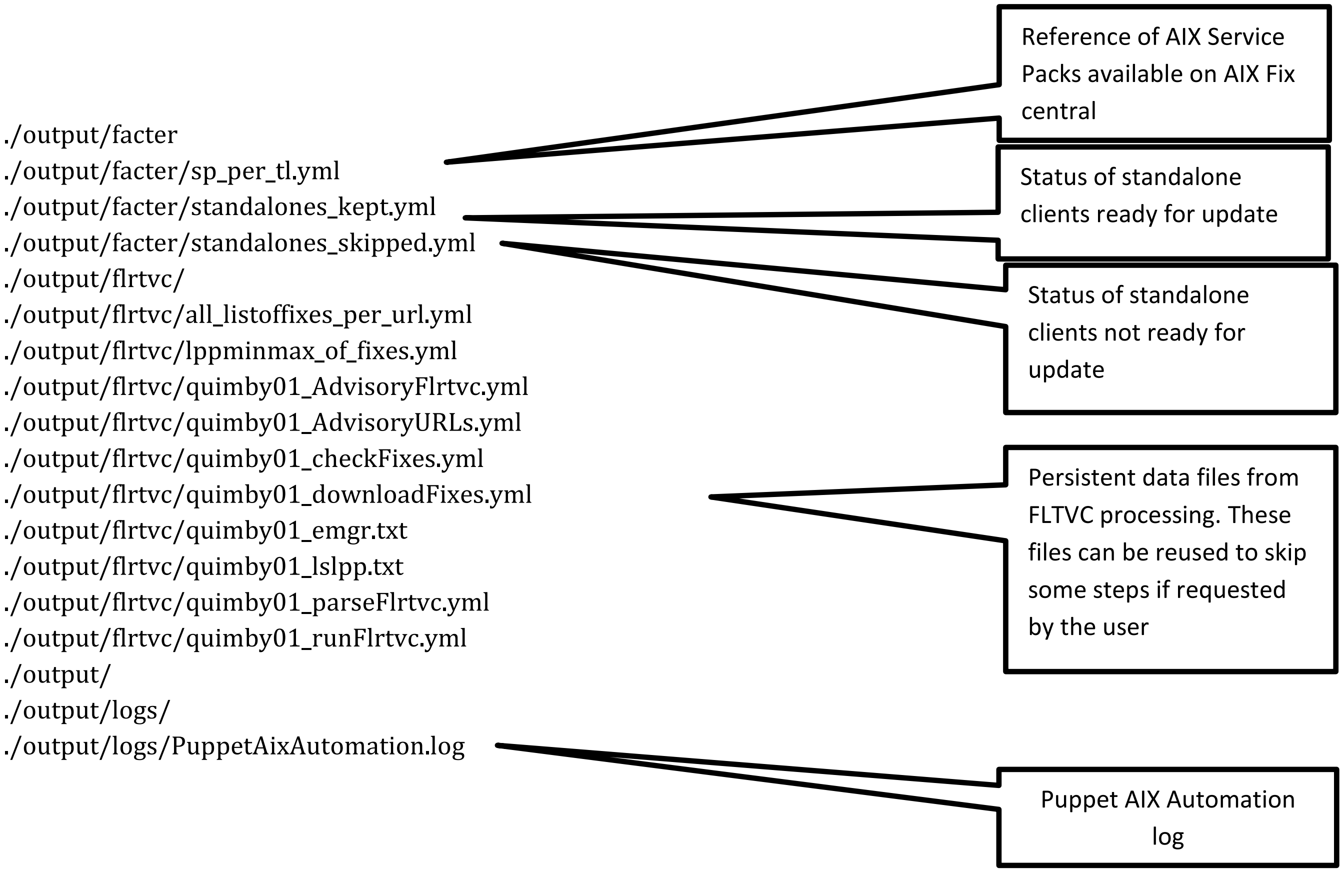
- Computing facters:
    - control and collect data from the manifest file,
    - collect data needed for the update from the configuration properties, for example the installation directory: ('/etc/puppetlabs/code/environments/production/modules') and the output directory ('/etc/puppetlabs/code/environments/production/modules/aixautomation/output'),
    - collect data from NIM configuration clients as AIX Service Pack numbers, level of eFix installed, status of connectivity, (only the data on NIM clients listed in the manifest file are checked),
    - collect data about all Services Pack numbers available from AIX Fix Central data base (generate sp_per_tl.yml file used as reference of available update).
- Validation of the custom types of the manifest: check the consistency of attributes used in the manifest file.
- Running the manifest: process the user resource declarations in the order they are found into the manifest file.

Output files – Logs

‘Puppet AIX and VIOS Automation’ module outputs files into:

./aixautomation/output/
./aixautomation/output/facter: output of facter
./aixautomation/output/flrtvc: output of flrtvc

The log file from Puppet AIX Automation process is in: **./output/logs/PuppetAixAutomation.log**
The log file is automatically rotated and 12 files of 1MB are kept.



```
./output/facter
./output/facter/sp_per_tl.yml
./output/facter/standalones_kept.yml
./output/facter/standalones_skipped.yml
./output/flrtvc/
./output/flrtvc/all_listoffixes_per_url.yml
./output/flrtvc/lppminmax_of_fixes.yml
./output/flrtvc/quimby01_AdvisoryFlrtvc.yml
./output/flrtvc/quimby01_AdvisoryURLs.yml
./output/flrtvc/quimby01_checkFixes.yml
./output/flrtvc/quimby01_downloadFixes.yml
./output/flrtvc/quimby01_emgr.txt
./output/flrtvc/quimby01_lslpp.txt
./output/flrtvc/quimby01_parseFlrtvc.yml
./output/flrtvc/quimby01_runFlrtvc.yml
./output/
./output/logs/
./output/logs/PuppetAixAutomation.log
```

Reference of AIX Service Packs available on AIX Fix central

Status of standalone clients ready for update

Status of standalone clients not ready for update

Persistent data files from FLTVC processing. These files can be reused to skip some steps if requested by the user

Puppet AIX Automation log

Remark: as a side effect of FLRTVC tool, file named apar.csv can be found into the directory where the ‘puppet apply’ command is launched. This file should cause no worry.

Use cases for updating AIX systems with ‘Puppet AIX and VIOS Automation’ module

This section describes some nominal use cases for updating AIX systems with ‘Puppet AIX and VIOS Automation’ module and describes the manifest file init.pp needed for each use case.

Hereunder examples were taken mostly from examples provided into:
./aixautomation/examples/init.pp

General explanation of structure of manifest file: init.pp

```
class aixautomation {
 download { " download request with SUMA provider":
  provider => suma,
  ensure => present,
  name => "test PAA_SP_7200-01_7200-01-03-1720",
  type => "SP",
  root => "/exports/extra/test-puppet/suma",
  from => "7200-01",
  to => "7200-01-03-1720",
  to_step => "download",
  lpp_source => "PAA_SP_7200-01_7200-01-03-1720",
 }
 patchmngt { "Patch management request with nimpush provider":
  provider   => nimpush,
  ensure     => present,
  name       => "update to 7200-01-03-1720",
  action     => "update",
  targets    => "client1 client2 client3",
  sync       => "yes",
  lpp_source => "PAA_SP_7200-01_7200-01-03-1720",
 }
 fix { "eFix_install with flrtvc provider ":
  provider => flrtvc,
  name     => "eFix_install",
  ensure   => present,
  to_step  => "installResource",
  targets  => "client1 client2 client3",
  force    => "yes",
  root     => "/exports/extra/test-puppet/flrtvc",
 }
}
```

Class automation to encapsulate the request

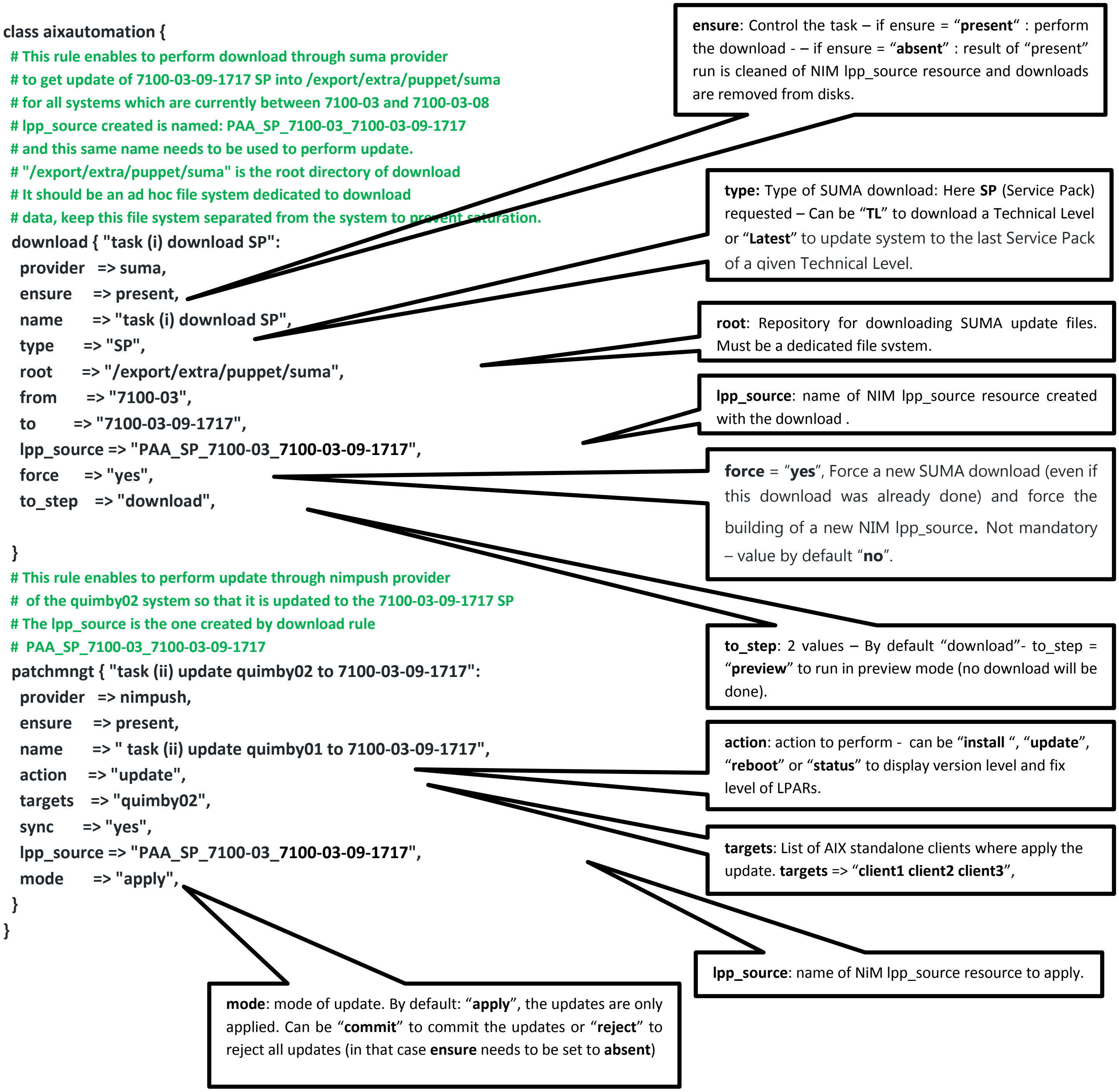
Attributes of custom type ‘download’: resource declaration to download SUMA resources for an update. A NIM lpp_source resource will be created as result of this resource declaration.

Attributes of custom type ‘patchmngt’. The lpp_source resource defined will be installed on the list of NIM standalone clients, called ‘targets’, as result of this resource declaration.

Attributes of custom type ‘fix’. The FLRTVC eFixes will be installed on the list of NIM standalone clients, called ‘targets’. This resource declaration automates the download of eFixes, the creation of the corresponding NIM lpp_source resource and the installation of this NIM resource through NIM operation.

Update AIX systems from AIX Technology Level 7100-03 to 7100-03-09-1717

Use case for updating AIX systems from TL “7100-03” to “7100-03-09-1717”. The user must define 2 tasks into the init.pp file: (i) One task to download update file with SUMA provider, (ii) one task with nimpush provider to install the update as lpp_source resource previously created by task (i).



```
class aixautomation {
 # This rule enables to perform download through suma provider
 # to get update of 7100-03-09-1717 SP into /export/extra/puppet/suma
 # for all systems which are currently between 7100-03 and 7100-03-08
 # lpp_source created is named: PAA_SP_7100-03_7100-03-09-1717
 # and this same name needs to be used to perform update.
 # "/export/extra/puppet/suma" is the root directory of download
 # It should be an ad hoc file system dedicated to download
 # data, keep this file system separated from the system to prevent saturation.
 download { "task (i) download SP":
   provider   => suma,
   ensure     => present,
   name       => "task (i) download SP",
   type       => "SP",
   root       => "/export/extra/puppet/suma",
   from       => "7100-03",
   to         => "7100-03-09-1717",
   lpp_source => "PAA_SP_7100-03_7100-03-09-1717",
   force      => "yes",
   to_step    => "download",

 }
 # This rule enables to perform update through nimpush provider
 #  of the quimby02 system so that it is updated to the 7100-03-09-1717 SP
 # The lpp_source is the one created by download rule
 #  PAA_SP_7100-03_7100-03-09-1717
 patchmngt { "task (ii) update quimby02 to 7100-03-09-1717":
   provider   => nimpush,
   ensure     => present,
   name       => " task (ii) update quimby01 to 7100-03-09-1717",
   action     => "update",
   targets    => "quimby02",
   sync       => "yes",
   lpp_source => "PAA_SP_7100-03_7100-03-09-1717",
   mode       => "apply",
 }
}
```

ensure: Control the task – if ensure = “**present**“ : perform the download - – if ensure = “**absent**” : result of “present” run is cleaned of NIM lpp_source resource and downloads are removed from disks.

type: Type of SUMA download: Here **SP** (Service Pack) requested – Can be “**TL**” to download a Technical Level or “**Latest**” to update system to the last Service Pack of a given Technical Level.

root: Repository for downloading SUMA update files. Must be a dedicated file system.

lpp_source: name of NIM lpp_source resource created with the download .

force = “**yes**”, Force a new SUMA download (even if this download was already done) and force the building of a new NIM lpp_source. Not mandatory – value by default “**no**”.

to_step: 2 values – By default “download”- to_step = “**preview**” to run in preview mode (no download will be done).

action: action to perform - can be “**install** “, “**update**”, “**reboot**” or “**status**” to display version level and fix level of LPARs.

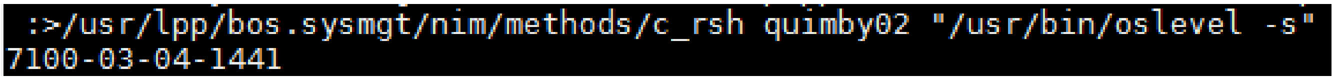
targets: List of AIX standalone clients where apply the update. **targets** => “**client1 client2 client3**”,

lpp_source: name of NiM lpp_source resource to apply.

mode: mode of update. By default: “**apply**”, the updates are only applied. Can be “**commit**” to commit the updates or “**reject**” to reject all updates (in that case **ensure** needs to be set to **absent**)

Steps of AIX system update from SUMA download

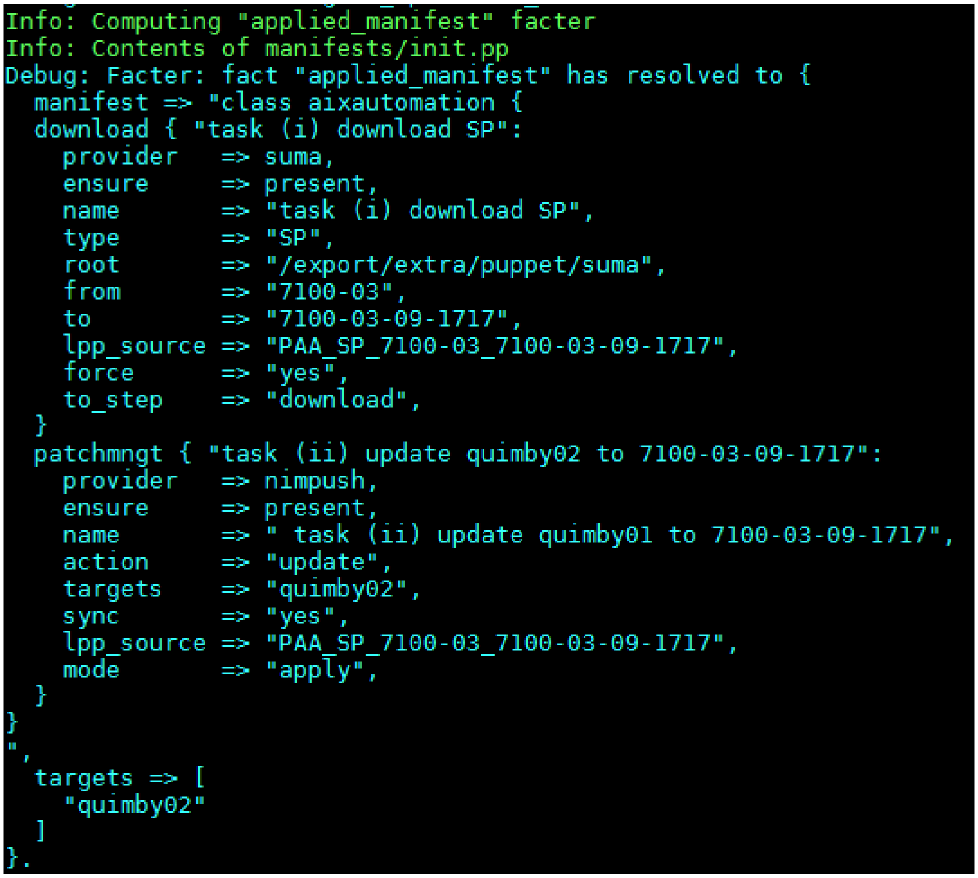
- Manual verification of ‘quimby02’ AIX oslevel before update



```
 :>/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby02 "/usr/bin/oslevel -s"
7100-03-04-1441
```

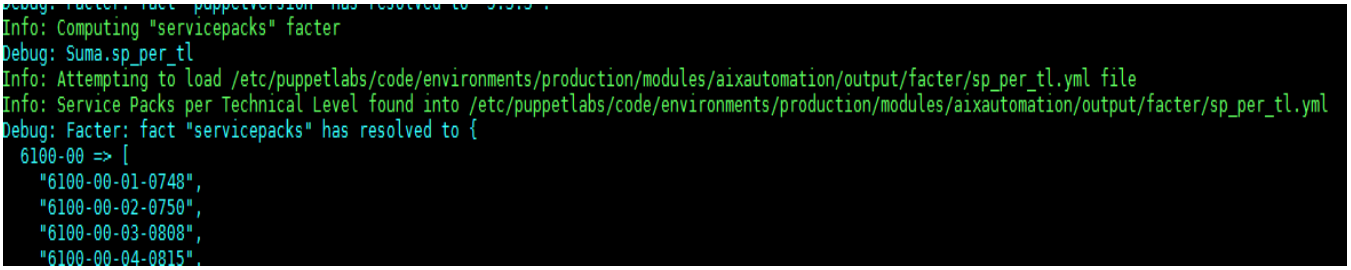
- Run ‘Puppet AIX and VIOS Automation’ module
  #/opt/puppetlabs/bin/puppet apply --debug --modulepath=/etc/puppetlabs/code/environments/production/modules/ -e "include aixautomation"

- Reminder about the manifest file ‘init.pp’ which is currently running



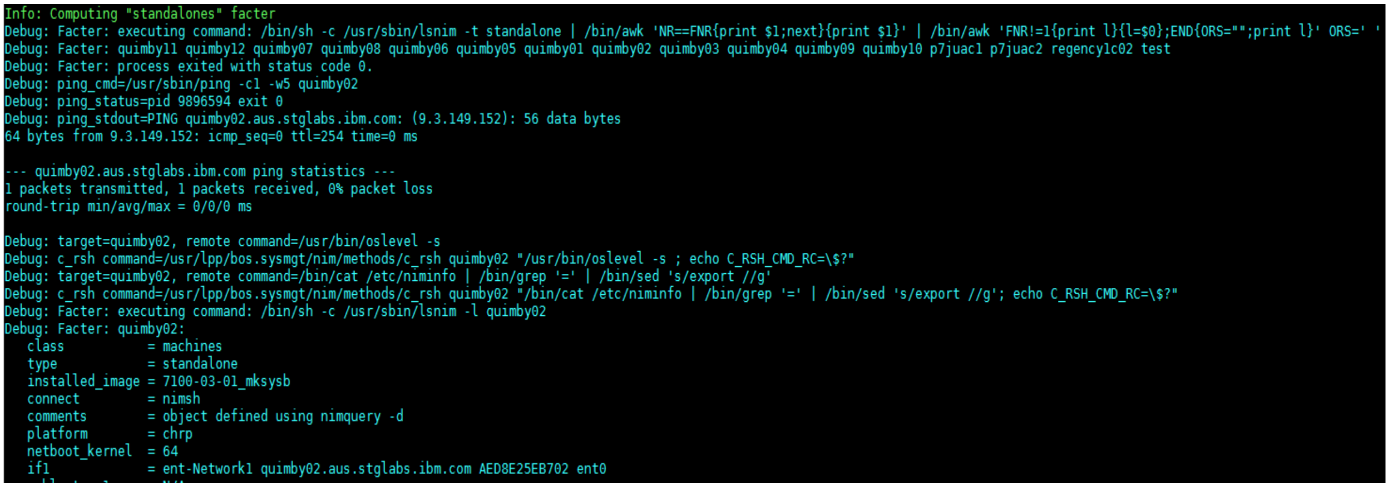
```
Info: Computing "applied_manifest" facter
Info: Contents of manifests/init.pp
Debug: Facter: fact "applied_manifest" has resolved to {
  manifest => "class aixautomation {
  download { "task (i) download SP":
    provider   => suma,
    ensure     => present,
    name       => "task (i) download SP",
    type       => "SP",
    root       => "/export/extra/puppet/suma",
    from       => "7100-03",
    to         => "7100-03-09-1717",
    lpp_source => "PAA_SP_7100-03_7100-03-09-1717",
    force      => "yes",
    to_step    => "download",
  }
  patchmngt { "task (ii) update quimby02 to 7100-03-09-1717":
    provider   => nimpush,
    ensure     => present,
    name       => " task (ii) update quimby01 to 7100-03-09-1717",
    action     => "update",
    targets    => "quimby02",
    sync       => "yes",
    lpp_source => "PAA_SP_7100-03_7100-03-09-1717",
    mode       => "apply",
  }
}
",
  targets => [
    "quimby02"
  ]
}.
```

- Computing facters
    - Check and build the reference file about AIX Service Pack per Technical Level: **sp_per_tl.yml**

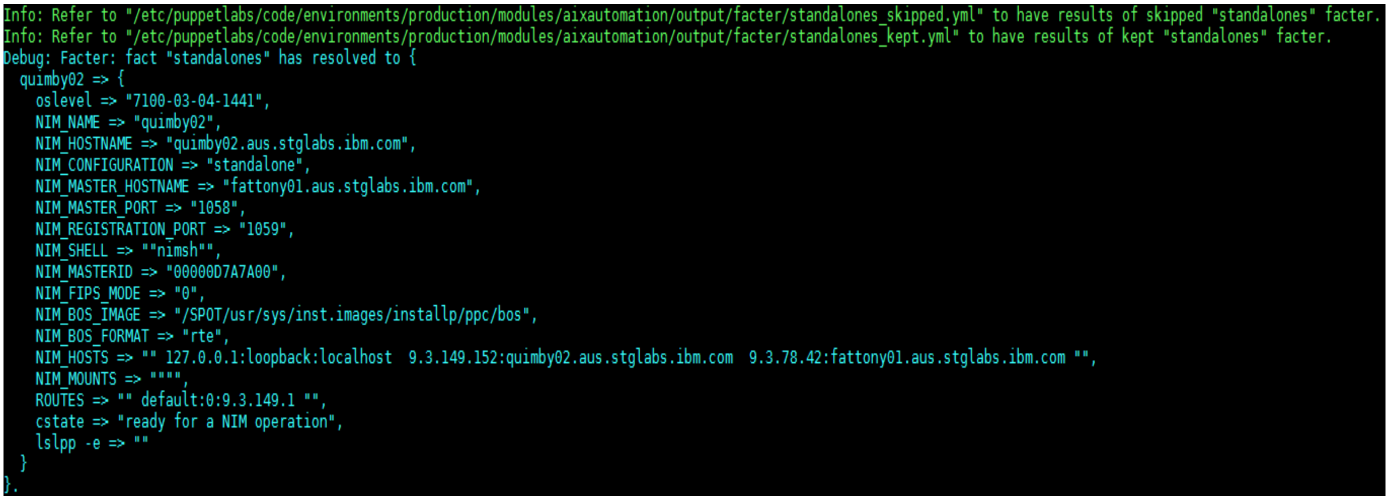


```
Debug: Facter: fact "puppetversion" has resolved to "5.3.3".
Info: Computing "servicepacks" facter
Debug: Suma.sp_per_tl
Info: Attempting to load /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/sp_per_tl.yml file
Info: Service Packs per Technical Level found into /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/sp_per_tl.yml
Debug: Facter: fact "servicepacks" has resolved to {
  6100-00 => [
    "6100-00-01-0748",
    "6100-00-02-0750",
    "6100-00-03-0808",
    "6100-00-04-0815",
```

- Computing standalone facters: collect information about standalone clients (called ‘targets’ in manifest file), and verify their connectivity and usability.

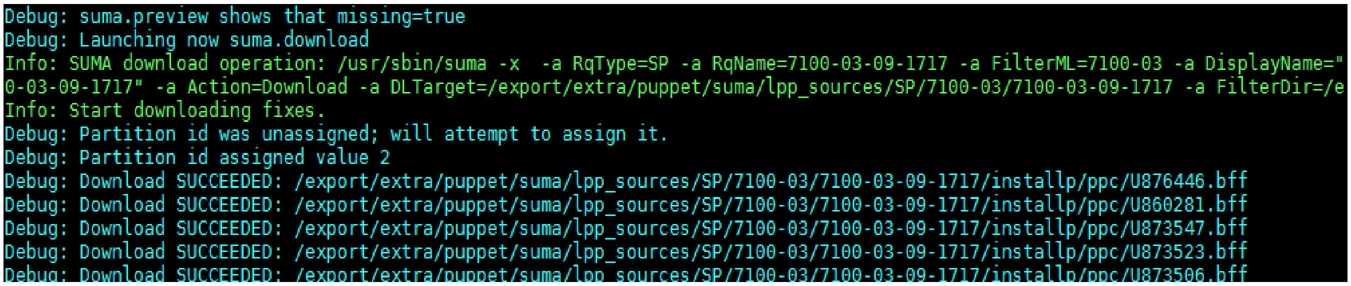


```
Info: Computing "standalones" facter
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -t standalone | /bin/awk 'NR==FNR{print $1;next}{print $1}' | /bin/awk 'FNR!=1{print l}{l=$0};END{ORS="";print l}' ORS=' '
Debug: Facter: quimby11 quimby12 quimby07 quimby08 quimby06 quimby05 quimby01 quimby02 quimby03 quimby04 quimby09 quimby10 p7juac1 p7juac2 regency1c02 test
Debug: Facter: process exited with status code 0.
Debug: ping_cmd=/usr/sbin/ping -c1 -w5 quimby02
Debug: ping_status=pid 9896594 exit 0
Debug: ping_stdout=PING quimby02.aus.stglabs.ibm.com: (9.3.149.152): 56 data bytes
64 bytes from 9.3.149.152: icmp_seq=0 ttl=254 time=0 ms

--- quimby02.aus.stglabs.ibm.com ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

Debug: target=quimby02, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby02 "/usr/bin/oslevel -s ; echo C_RSH_CMD_RC=\$?"
Debug: target=quimby02, remote command=/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g'
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby02 "/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g'; echo C_RSH_CMD_RC=\$?"
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -l quimby02
Debug: Facter: quimby02:
   class          = machines
   type           = standalone
   installed_image = 7100-03-01_mksysb
   connect        = nimsh
   comments       = object defined using nimquery -d
   platform       = chrp
   netboot_kernel  = 64
   if1            = ent-Network1 quimby02.aus.stglabs.ibm.com AED8E25EB702 ent0
```



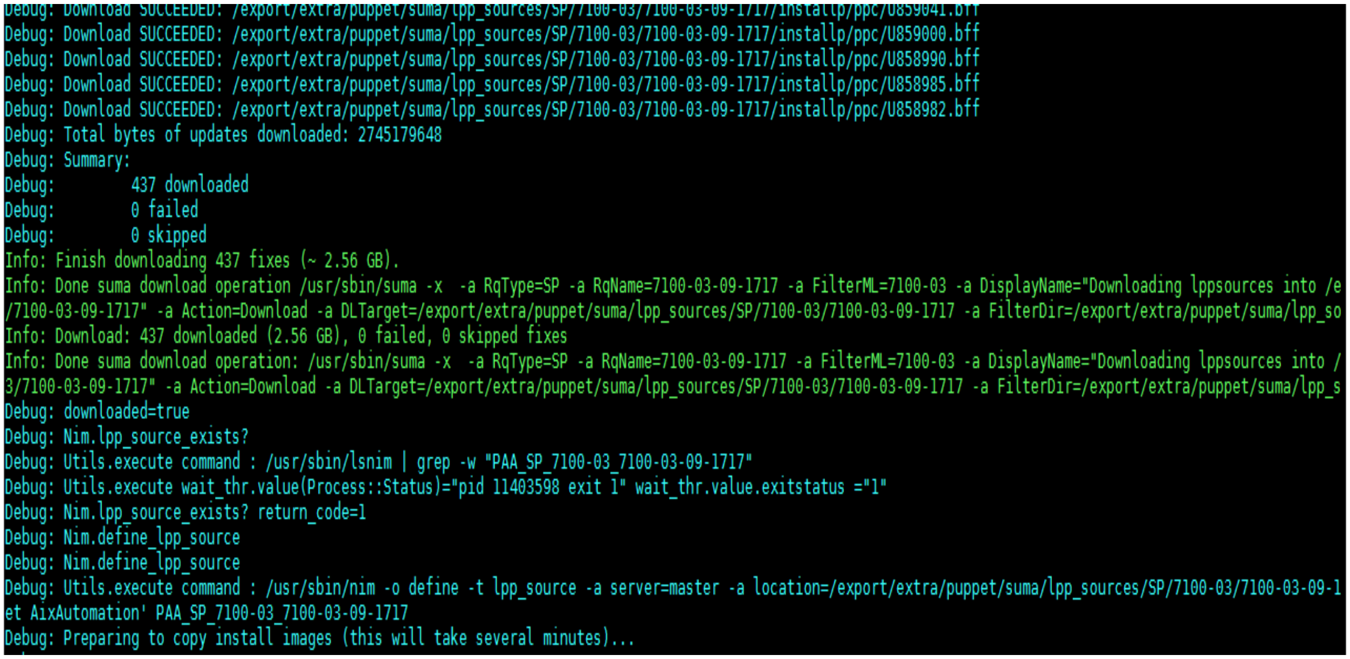
```
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/standalones_skipped.yml" to have results of skipped "standalones" facter.
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/standalones_kept.yml" to have results of kept "standalones" facter.
Debug: Facter: fact "standalones" has resolved to {
  quimby02 => {
    oslevel => "7100-03-04-1441",
    NIM_NAME => "quimby02",
    NIM_HOSTNAME => "quimby02.aus.stglabs.ibm.com",
    NIM_CONFIGURATION => "standalone",
    NIM_MASTER_HOSTNAME => "fattony01.aus.stglabs.ibm.com",
    NIM_MASTER_PORT => "1058",
    NIM_REGISTRATION_PORT => "1059",
    NIM_SHELL => ""nimsh"",
    NIM_MASTERID => "00000D7A7A00",
    NIM_FIPS_MODE => "0",
    NIM_BOS_IMAGE => "/SPOT/usr/sys/inst.images/installp/ppc/bos",
    NIM_BOS_FORMAT => "rte",
    NIM_HOSTS => "" 127.0.0.1:loopback:localhost  9.3.149.152:quimby02.aus.stglabs.ibm.com  9.3.78.42:fattony01.aus.stglabs.ibm.com "",
    NIM_MOUNTS => """",
    ROUTES => "" default:0:9.3.149.1 "",
    cstate => "ready for a NIM operation",
    lslpp -e => ""
  }
}.
```

- Performing SUMA download



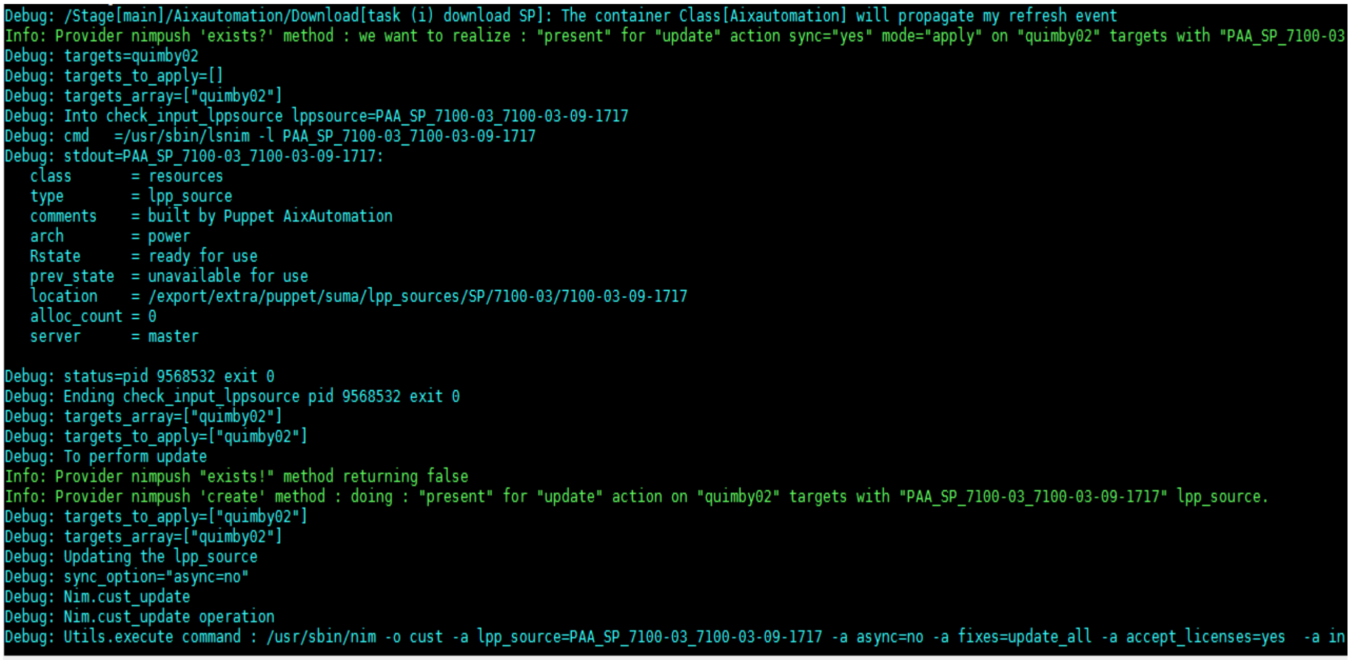
```
Debug: suma.preview shows that missing=true
Debug: Launching now suma.download
Info: SUMA download operation: /usr/sbin/suma -x  -a RqType=SP -a RqName=7100-03-09-1717 -a FilterML=7100-03 -a DisplayName="
0-03-09-1717" -a Action=Download -a DLTarget=/export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717 -a FilterDir=/e
Info: Start downloading fixes.
Debug: Partition id was unassigned; will attempt to assign it.
Debug: Partition id assigned value 2
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U876446.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U860281.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U873547.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U873523.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U873506.bff
```

- Creating NIM lpp_source resource from SUMA download.



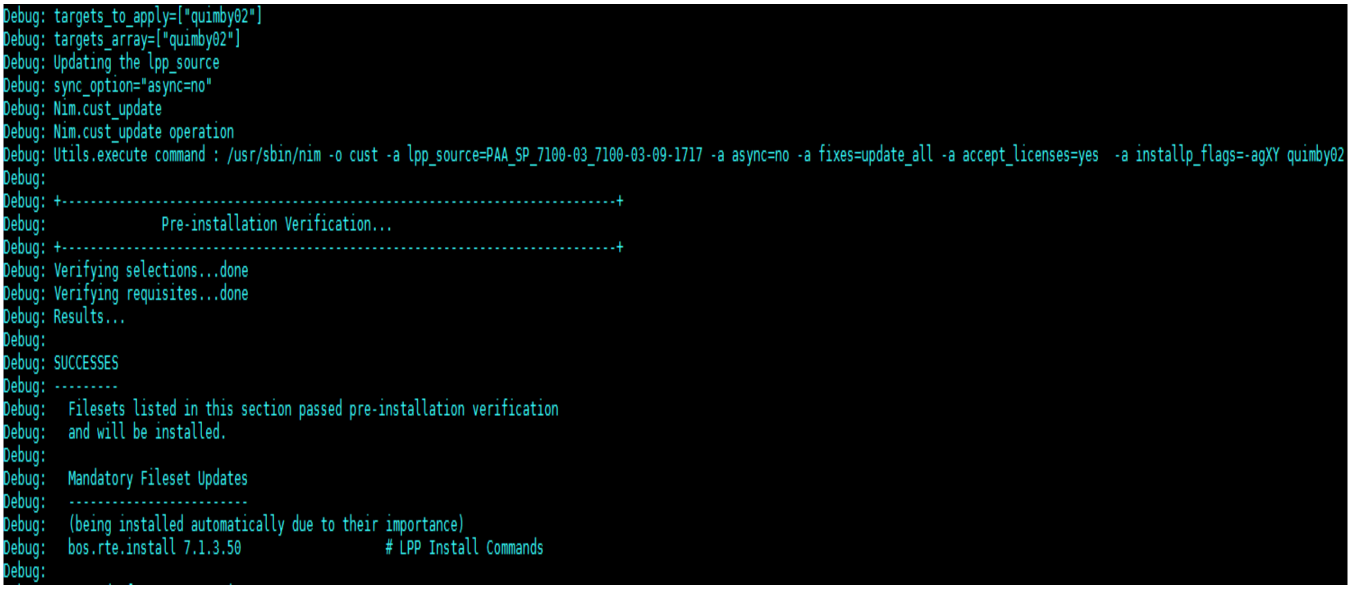
```
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U859041.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U859000.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U858990.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U858985.bff
Debug: Download SUCCEEDED: /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717/installp/ppc/U858982.bff
Debug: Total bytes of updates downloaded: 2745179648
Debug: Summary:
Debug:         437 downloaded
Debug:         0 failed
Debug:         0 skipped
Info: Finish downloading 437 fixes (~ 2.56 GB).
Info: Done suma download operation /usr/sbin/suma -x  -a RqType=SP -a RqName=7100-03-09-1717 -a FilterML=7100-03 -a DisplayName="Downloading lppsources into /e
/7100-03-09-1717" -a Action=Download -a DLTarget=/export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717 -a FilterDir=/export/extra/puppet/suma/lpp_so
Info: Download: 437 downloaded (2.56 GB), 0 failed, 0 skipped fixes
Info: Done suma download operation: /usr/sbin/suma -x  -a RqType=SP -a RqName=7100-03-09-1717 -a FilterML=7100-03 -a DisplayName="Downloading lppsources into /
3/7100-03-09-1717" -a Action=Download -a DLTarget=/export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717 -a FilterDir=/export/extra/puppet/suma/lpp_s
Debug: downloaded=true
Debug: Nim.lpp_source_exists?
Debug: Utils.execute command : /usr/sbin/lsnim | grep -w "PAA_SP_7100-03_7100-03-09-1717"
Debug: Utils.execute wait_thr.value(Process::Status)="pid 11403598 exit 1" wait_thr.value.exitstatus ="1"
Debug: Nim.lpp_source_exists? return_code=1
Debug: Nim.define_lpp_source
Debug: Nim.define_lpp_source
Debug: Utils.execute command : /usr/sbin/nim -o define -t lpp_source -a server=master -a location=/export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1
et AixAutomation' PAA_SP_7100-03_7100-03-09-1717
Debug: Preparing to copy install images (this will take several minutes)...
```

- ➢ Performing NIM operation to update with the NIM lpp_source resource previously built from SUMA download.

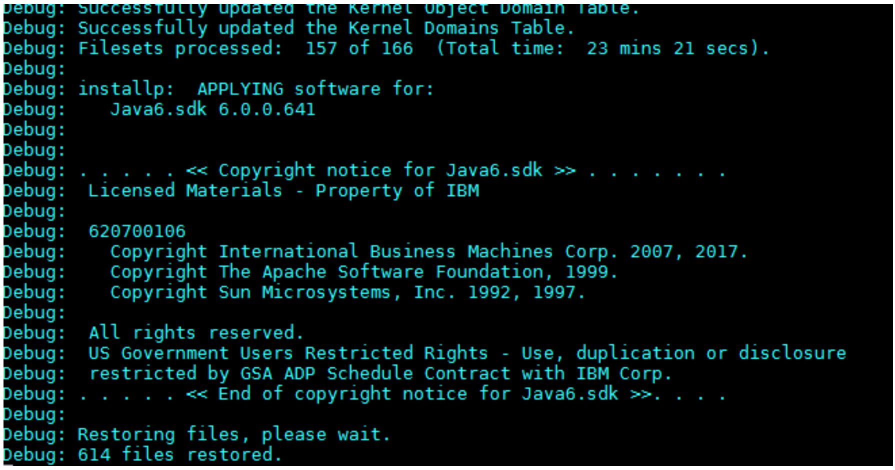


```
Debug: /Stage[main]/Aixautomation/Download[task (i) download SP]: The container Class[Aixautomation] will propagate my refresh event
Info: Provider nimpush 'exists?' method : we want to realize : "present" for "update" action sync="yes" mode="apply" on "quimby02" targets with "PAA_SP_7100-03
Debug: targets=quimby02
Debug: targets_to_apply=[]
Debug: targets_array=["quimby02"]
Debug: Into check_input_lppsource lppsource=PAA_SP_7100-03_7100-03-09-1717
Debug: cmd   =/usr/sbin/lsnim -l PAA_SP_7100-03_7100-03-09-1717
Debug: stdout=PAA_SP_7100-03_7100-03-09-1717:
   class       = resources
   type        = lpp_source
   comments    = built by Puppet AixAutomation
   arch        = power
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /export/extra/puppet/suma/lpp_sources/SP/7100-03/7100-03-09-1717
   alloc_count = 0
   server      = master

Debug: status=pid 9568532 exit 0
Debug: Ending check_input_lppsource pid 9568532 exit 0
Debug: targets_array=["quimby02"]
Debug: targets_to_apply=["quimby02"]
Debug: To perform update
Info: Provider nimpush "exists!" method returning false
Info: Provider nimpush 'create' method : doing : "present" for "update" action on "quimby02" targets with "PAA_SP_7100-03_7100-03-09-1717" lpp_source.
Debug: targets_to_apply=["quimby02"]
Debug: targets_array=["quimby02"]
Debug: Updating the lpp_source
Debug: sync_option="async=no"
Debug: Nim.cust_update
Debug: Nim.cust_update operation
Debug: Utils.execute command : /usr/sbin/nim -o cust -a lpp_source=PAA_SP_7100-03_7100-03-09-1717 -a async=no -a fixes=update_all -a accept_licenses=yes  -a in
```

- ➢ Starting the update installation



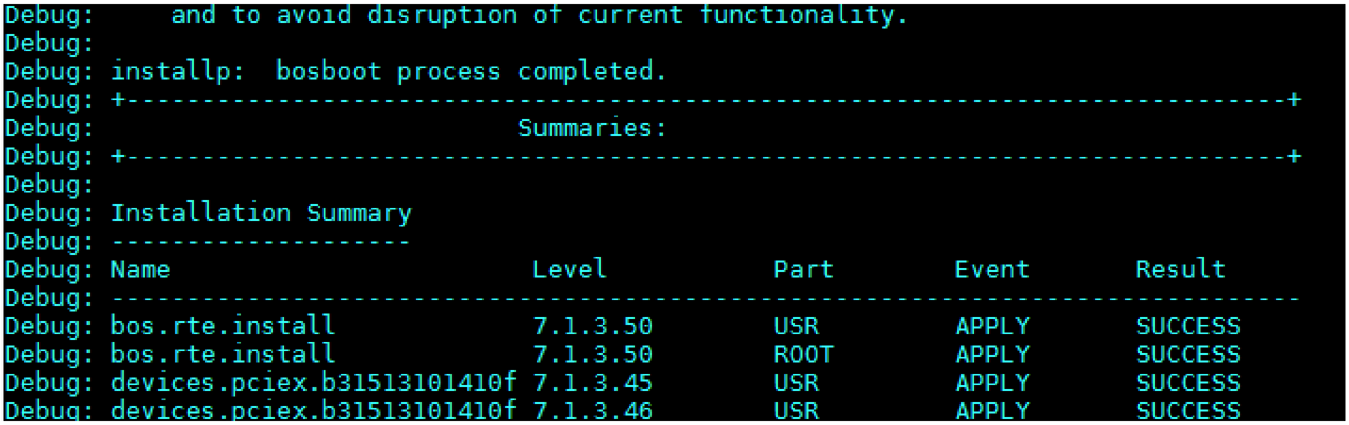
```
Debug: targets_to_apply=["quimby02"]
Debug: targets_array=["quimby02"]
Debug: Updating the lpp_source
Debug: sync_option="async=no"
Debug: Nim.cust_update
Debug: Nim.cust_update operation
Debug: Utils.execute command : /usr/sbin/nim -o cust -a lpp_source=PAA_SP_7100-03_7100-03-09-1717 -a async=no -a fixes=update_all -a accept_licenses=yes  -a installp_flags=-agXY quimby02
Debug:
Debug: +-----------------------------------------------------------------------------+
Debug:                 Pre-installation Verification...
Debug: +-----------------------------------------------------------------------------+
Debug: Verifying selections...done
Debug: Verifying requisites...done
Debug: Results...
Debug:
Debug: SUCCESSES
Debug: ---------
Debug:   Filesets listed in this section passed pre-installation verification
Debug:   and will be installed.
Debug:
Debug:   Mandatory Fileset Updates
Debug:   -------------------------
Debug:   (being installed automatically due to their importance)
Debug:   bos.rte.install 7.1.3.50                   # LPP Install Commands
Debug:
```



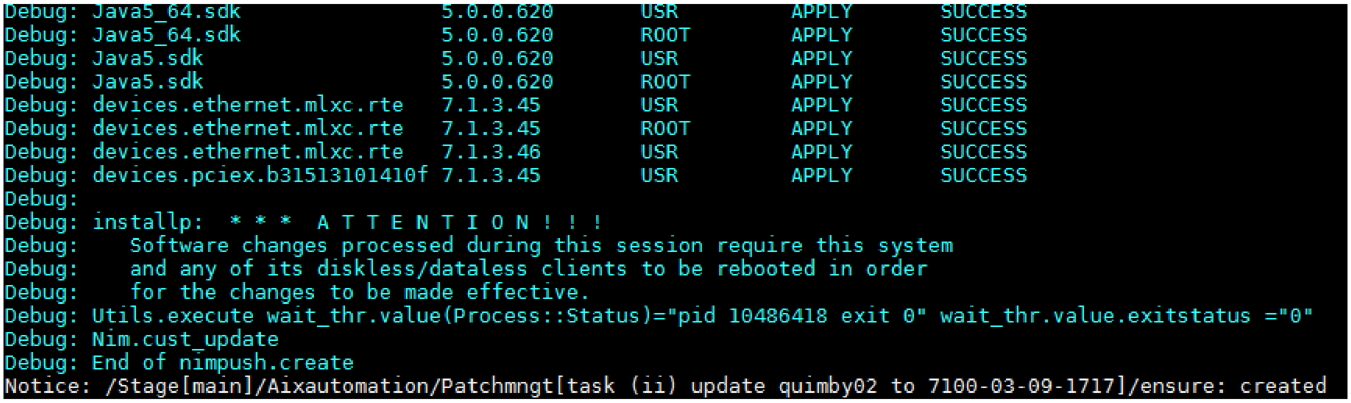
```
Debug: Successfully updated the Kernel Object Domain Table.
Debug: Successfully updated the Kernel Domains Table.
Debug: Filesets processed:  157 of 166  (Total time:  23 mins 21 secs).
Debug:
Debug: installp:  APPLYING software for:
Debug:    Java6.sdk 6.0.0.641
Debug:
Debug:
Debug: . . . . . << Copyright notice for Java6.sdk >> . . . . . . .
Debug:  Licensed Materials - Property of IBM
Debug:
Debug:  620700106
Debug:    Copyright International Business Machines Corp. 2007, 2017.
Debug:    Copyright The Apache Software Foundation, 1999.
Debug:    Copyright Sun Microsystems, Inc. 1992, 1997.
Debug:
Debug:  All rights reserved.
Debug:  US Government Users Restricted Rights - Use, duplication or disclosure
Debug:  restricted by GSA ADP Schedule Contract with IBM Corp.
Debug: . . . . . << End of copyright notice for Java6.sdk >>. . . .
Debug:
Debug: Restoring files, please wait.
Debug: 614 files restored.
```

....

- ➢ Finalization of installation - Status of the update



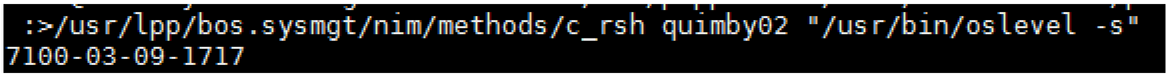
```
Debug:     and to avoid disruption of current functionality.
Debug:
Debug: installp:  bosboot process completed.
Debug: +-----------------------------------------------------------------------------+
Debug:                              Summaries:
Debug: +-----------------------------------------------------------------------------+
Debug:
Debug: Installation Summary
Debug: --------------------
Debug: Name                        Level           Part        Event       Result
Debug: -------------------------------------------------------------------------------
Debug: bos.rte.install             7.1.3.50        USR         APPLY       SUCCESS
Debug: bos.rte.install             7.1.3.50        ROOT        APPLY       SUCCESS
Debug: devices.pciex.b315131014l0f 7.1.3.45        USR         APPLY       SUCCESS
Debug: devices.pciex.b315131014l0f 7.1.3.46        USR         APPLY       SUCCESS
```



```
Debug: Java5_64.sdk                5.0.0.620       USR         APPLY       SUCCESS
Debug: Java5_64.sdk                5.0.0.620       ROOT        APPLY       SUCCESS
Debug: Java5.sdk                   5.0.0.620       USR         APPLY       SUCCESS
Debug: Java5.sdk                   5.0.0.620       ROOT        APPLY       SUCCESS
Debug: devices.ethernet.mlxc.rte   7.1.3.45        USR         APPLY       SUCCESS
Debug: devices.ethernet.mlxc.rte   7.1.3.45        ROOT        APPLY       SUCCESS
Debug: devices.ethernet.mlxc.rte   7.1.3.46        USR         APPLY       SUCCESS
Debug: devices.pciex.b315131014l0f 7.1.3.45        USR         APPLY       SUCCESS
Debug:
Debug: installp:  * * *  A T T E N T I O N ! ! !
Debug:    Software changes processed during this session require this system
Debug:    and any of its diskless/dataless clients to be rebooted in order
Debug:    for the changes to be made effective.
Debug: Utils.execute wait_thr.value(Process::Status)="pid 10486418 exit 0" wait_thr.value.exitstatus ="0"
Debug: Nim.cust_update
Debug: End of nimpush.create
Notice: /Stage[main]/Aixautomation/Patchmngt[task (ii) update quimby02 to 7100-03-09-1717]/ensure: created
```

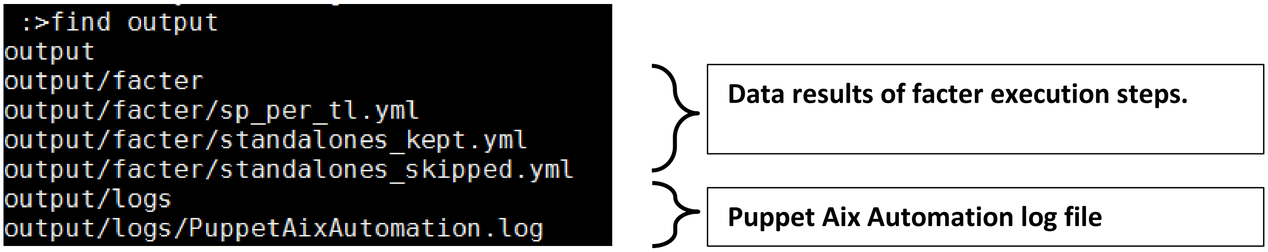
Verification of AIX oslevel after update for the NIM client quimby02

Successfully updated to **7100-03-09-1717**



```
 :>/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby02 "/usr/bin/oslevel -s"
7100-03-09-1717
```

Output files from ‘Puppet AIX and VIOS Automation’ module update with SUMA download and NIM push operations



```
 :>find output
output
output/facter
output/facter/sp_per_tl.yml
output/facter/standalones_kept.yml
output/facter/standalones_skipped.yml
output/logs
output/logs/PuppetAixAutomation.log
```

Data results of facter execution steps.

Puppet Aix Automation log file

Update AIX systems with FLRTVC eFixes.

Use case for updating AIX systems eFixes using FLRTVC recommendations. The user must define a resource declaration into the init.pp: “task of fix installation”

```
class aixautomation {
  # This rule enables to install iFix through flrtvc provider,
  #  on the quimby01 and quimby02 systems so that they are fixed as best as possible.
  # Setting ensure to 'present'
  # List of appropriate eFix is computed for each system, and then applied.
  # Possible steps are : runFlrtvc, parseFlrtvc, downloadFixes, checkFixes, buildResource
  #  installResource.
  # Clean is by default set to 'yes', but if you want to spare time and reuse previous
  # result of computation for each step, you can set force to 'no'.
  # "/export/extra/puppet/flrtvc" is suggested as root directory of download
  # It should be an ad hoc file system dedicated to download
  #  data, keep this file system separated from the system to prevent saturation

  fix { "task fix installation":
    provider => flrtvc,
    name     => " task fix installation ",
    ensure   => present,
    to_step  => "installResource",
    targets  => "quimby01 quimby02",
    force    => "yes",
    root     => "/export/extra/puppet/flrtvc",
  }
}
```

ensure: control the task – if ensure = “**present**“ : eFixes are installed– if ensure = “**absent**” : eFixes are removed.

to_step: to control the flow of the execution steps: Possible values by chronologic execution order: **installFlrtvc, runFlrtvc, parseFlrtvc, downloadFixes, checkFixes, buildResource, installResource.** By default, **installResource** is assumed to install the FLRTVC eFixes (all steps are done).

targets: list of AIX standalone clients where eFixes installation will be done.

force = “yes” to force all FLRTVC steps to be done. By default **yes** is assumed. If **force = “no”,** persisted data from previous run will be used (data from ./output/flrtvc/ repository)

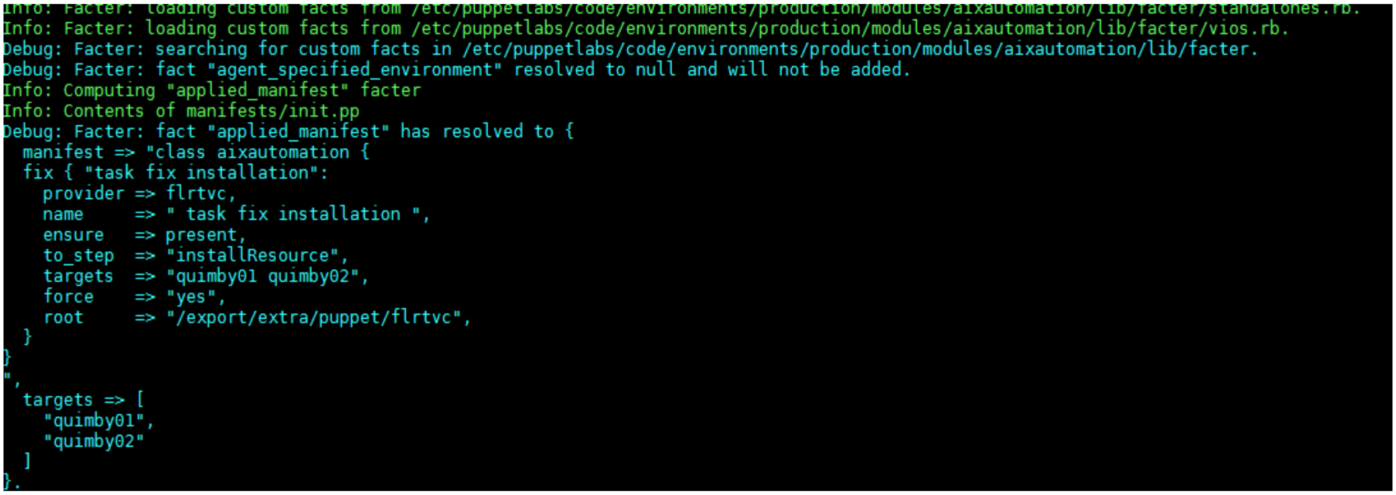
root: repository for downloading SUMA update files. Must be a dedicated file system.

Steps of FLRTV update

- Run ‘Puppet AIX and VIOS Automation’ module

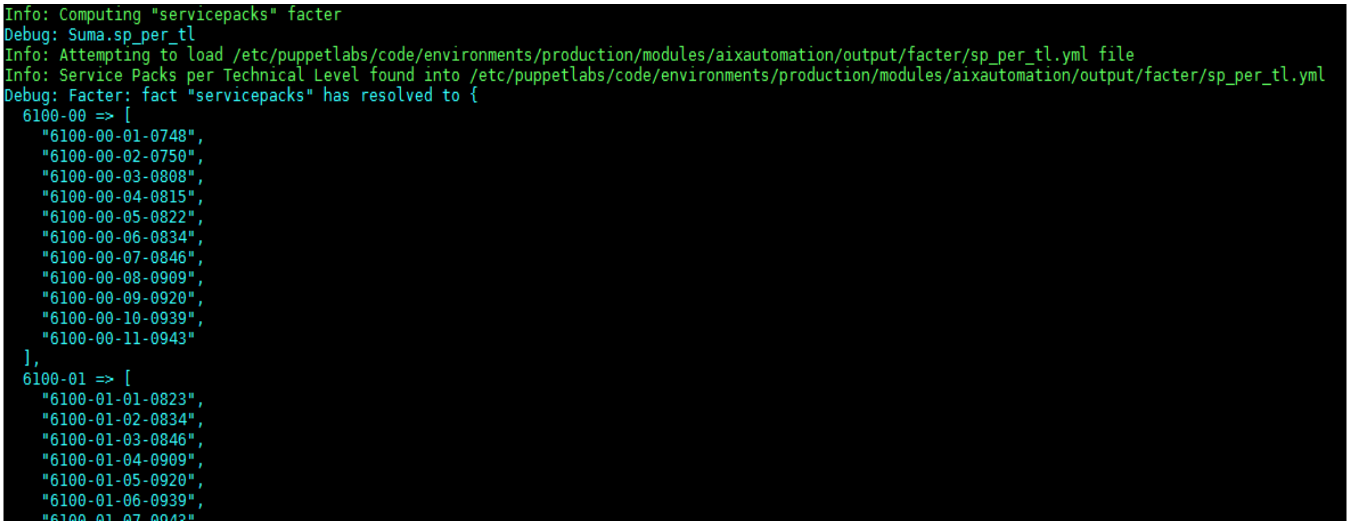
  #/opt/puppetlabs/bin/puppet apply --debug --modulepath=/etc/puppetlabs/code/environments/production/modules/ -e "include aixautomation"

- Reminder about the manifest file ‘init.pp’ which is being used by runtime



```
Info: Facter: loading custom facts from /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter/standalones.rb.
Info: Facter: loading custom facts from /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter/vios.rb.
Debug: Facter: searching for custom facts in /etc/puppetlabs/code/environments/production/modules/aixautomation/lib/facter.
Debug: Facter: fact "agent_specified_environment" resolved to null and will not be added.
Info: Computing "applied_manifest" facter
Info: Contents of manifests/init.pp
Debug: Facter: fact "applied_manifest" has resolved to {
  manifest => "class aixautomation {
  fix { "task fix installation":
    provider => flrtvc,
    name     => " task fix installation ",
    ensure   => present,
    to_step  => "installResource",
    targets  => "quimby01 quimby02",
    force    => "yes",
    root     => "/export/extra/puppet/flrtvc",
  }
}
",
  targets => [
    "quimby01",
    "quimby02"
  ]
}.
```

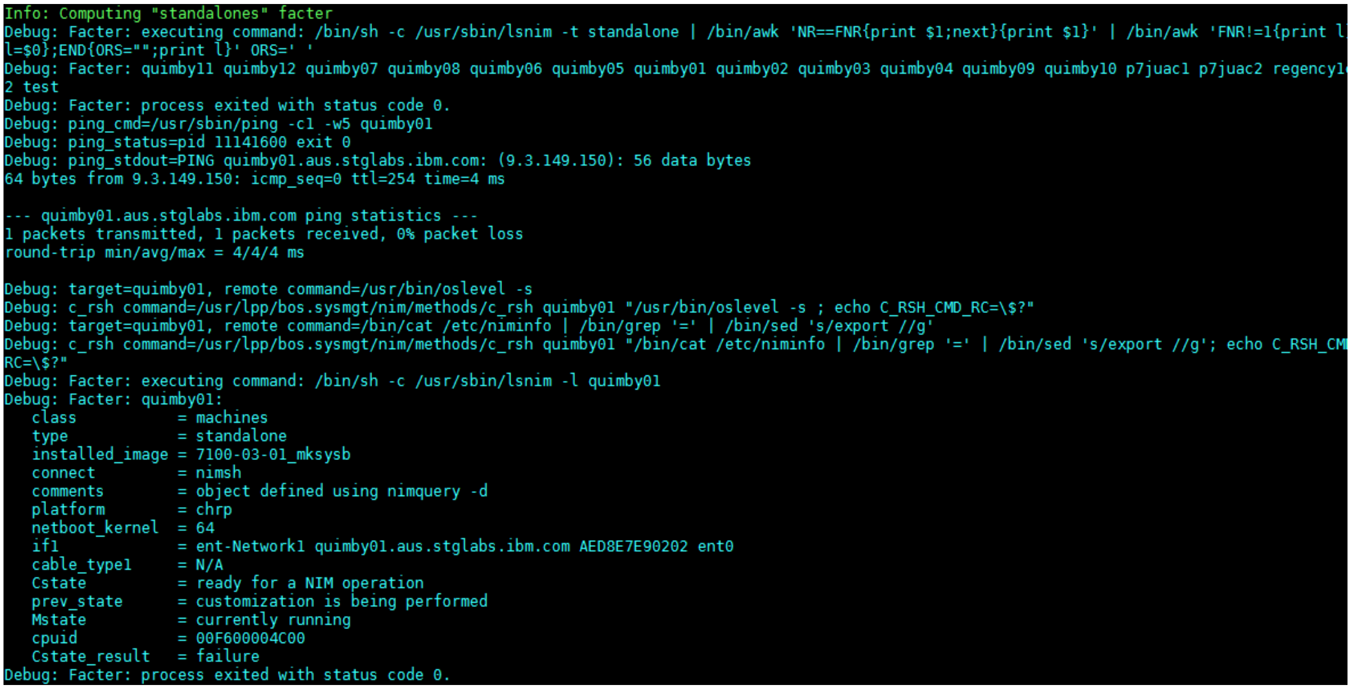
- Computing facters
  - ⇨ Check and build the reference file about AIX Service Pack per Technical Level: **sp_per_tl.yml**



```
Info: Computing "servicepacks" facter
Debug: Suma.sp_per_tl
Info: Attempting to load /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/sp_per_tl.yml file
Info: Service Packs per Technical Level found into /etc/puppetlabs/code/environments/production/modules/aixautomation/output/facter/sp_per_tl.yml
Debug: Facter: fact "servicepacks" has resolved to {
  6100-00 => [
    "6100-00-01-0748",
    "6100-00-02-0750",
    "6100-00-03-0808",
    "6100-00-04-0815",
    "6100-00-05-0822",
    "6100-00-06-0834",
    "6100-00-07-0846",
    "6100-00-08-0909",
    "6100-00-09-0920",
    "6100-00-10-0939",
    "6100-00-11-0943"
  ],
  6100-01 => [
    "6100-01-01-0823",
    "6100-01-02-0834",
    "6100-01-03-0846",
    "6100-01-04-0909",
    "6100-01-05-0920",
    "6100-01-06-0939",
    "6100-01-07-0943"
```

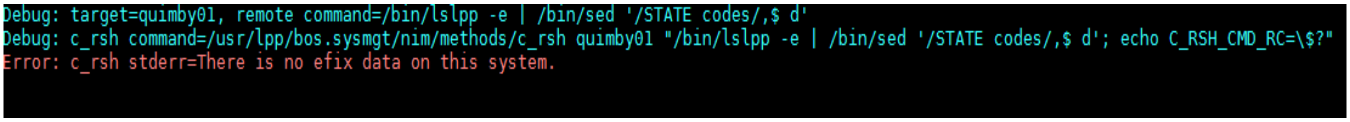
  - ⇨ Computing standalones facters: collect information about standalone clients (called ‘targets’ in the manifest file).

- Collect NIM Status for each target

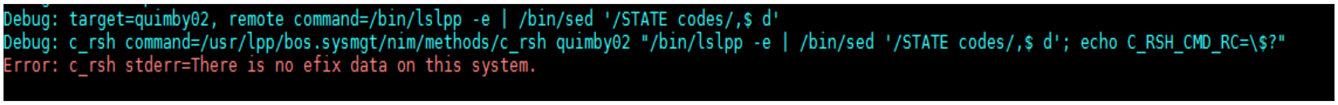


```
Info: Computing "standalones" facter
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -t standalone | /bin/awk 'NR==FNR{print $1;next}{print $1}' | /bin/awk 'FNR!=1{print l
l=$0};END{ORS="";print l}' ORS=' '
Debug: Facter: quimby11 quimby12 quimby07 quimby08 quimby06 quimby05 quimby01 quimby02 quimby03 quimby04 quimby09 quimby10 p7juac1 p7juac2 regency1
2 test
Debug: Facter: process exited with status code 0.
Debug: ping_cmd=/usr/sbin/ping -c1 -w5 quimby01
Debug: ping_status=pid 11141600 exit 0
Debug: ping_stdout=PING quimby01.aus.stglabs.ibm.com: (9.3.149.150): 56 data bytes
64 bytes from 9.3.149.150: icmp_seq=0 ttl=254 time=4 ms

--- quimby01.aus.stglabs.ibm.com ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 4/4/4 ms

Debug: target=quimby01, remote command=/usr/bin/oslevel -s
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/usr/bin/oslevel -s ; echo C_RSH_CMD_RC=\$?"
Debug: target=quimby01, remote command=/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g'
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/bin/cat /etc/niminfo | /bin/grep '=' | /bin/sed 's/export //g'; echo C_RSH_CM
RC=\$?"
Debug: Facter: executing command: /bin/sh -c /usr/sbin/lsnim -l quimby01
Debug: Facter: quimby01:
   class           = machines
   type            = standalone
   installed_image = 7100-03-01_mksysb
   connect         = nimsh
   comments        = object defined using nimquery -d
   platform        = chrp
   netboot_kernel  = 64
   if1             = ent-Network1 quimby01.aus.stglabs.ibm.com AED8E7E90202 ent0
   cable_type1     = N/A
   Cstate          = ready for a NIM operation
   prev_state      = customization is being performed
   Mstate          = currently running
   cpuid           = 00F600004C00
   Cstate_result   = failure
Debug: Facter: process exited with status code 0.
```

- Collecting eFix Status for each target



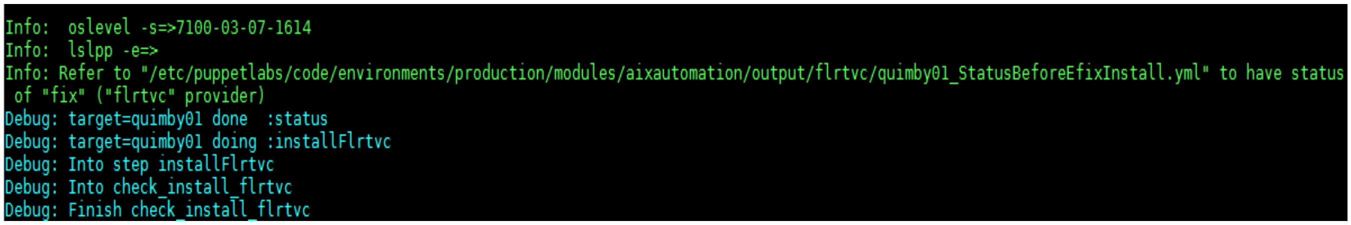
```
Debug: target=quimby01, remote command=/bin/lslpp -e | /bin/sed '/STATE codes/,$ d'
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/bin/lslpp -e | /bin/sed '/STATE codes/,$ d'; echo C_RSH_CMD_RC=\$?"
Error: c_rsh stderr=There is no efix data on this system.
```



```
Debug: target=quimby02, remote command=/bin/lslpp -e | /bin/sed '/STATE codes/,$ d'
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby02 "/bin/lslpp -e | /bin/sed '/STATE codes/,$ d'; echo C_RSH_CMD_RC=\$?"
Error: c_rsh stderr=There is no efix data on this system.
```

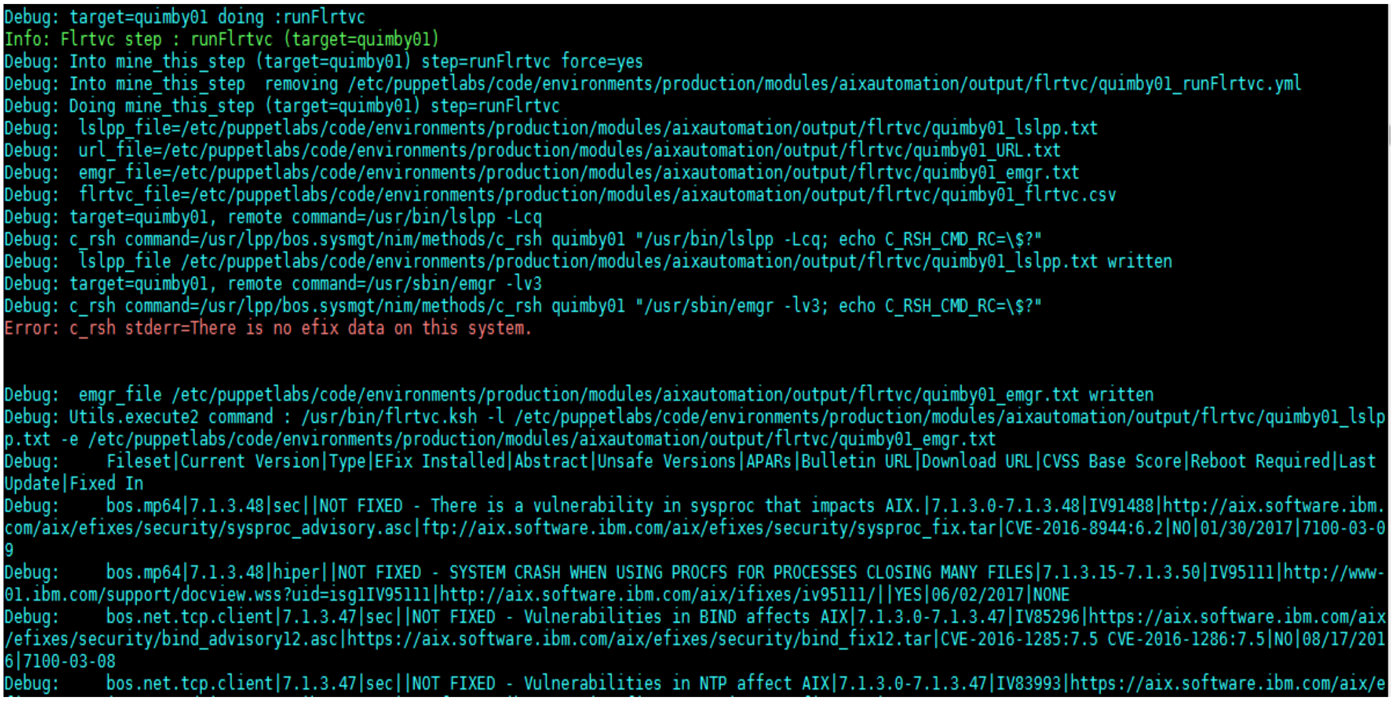
⇨ Computing manifest file: running FLRTVC eFix resource declaration.

- Checking FLRTVC installation: if ‘flrtvc.ksh’ is not installed, it is automatically downloaded and installed.



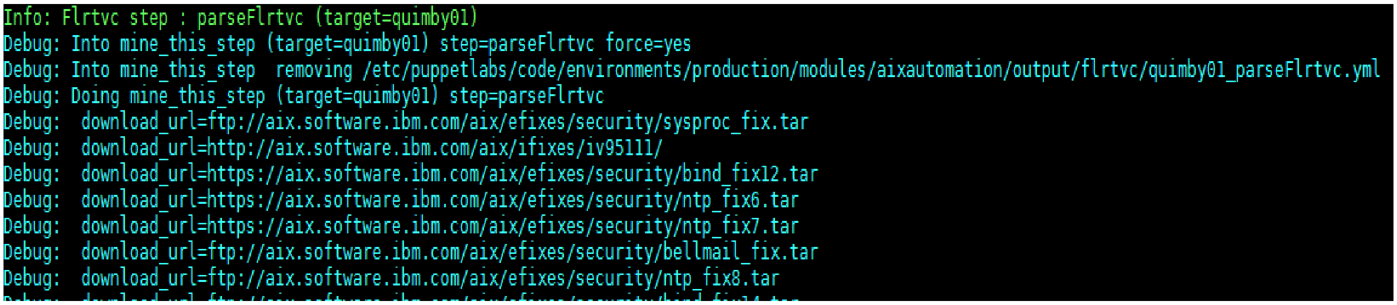
```
Info:  oslevel -s=>7100-03-07-1614
Info:  lslpp -e=>
Info: Refer to "/etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_StatusBeforeEfixInstall.yml" to have status
 of "fix" ("flrtvc" provider)
Debug: target=quimby01 done  :status
Debug: target=quimby01 doing :installFlrtvc
Debug: Into step installFlrtvc
Debug: Into check_install_flrtvc
Debug: Finish check_install_flrtvc
```

- Launching FLRTVC recommendations tools to know which eFixes are necessary to download for each target, for that we collect information from each target, to provide these information as input to ‘flrtvc.ksh’ script.



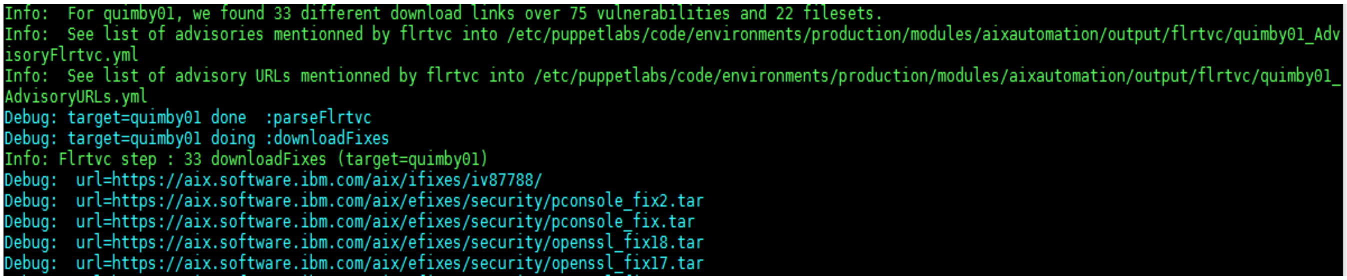
```
Debug: target=quimby01 doing :runFlrtvc
Info: Flrtvc step : runFlrtvc (target=quimby01)
Debug: Into mine_this_step (target=quimby01) step=runFlrtvc force=yes
Debug: Into mine_this_step  removing /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_runFlrtvc.yml
Debug: Doing mine_this_step (target=quimby01) step=runFlrtvc
Debug:  lslpp_file=/etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_lslpp.txt
Debug:  url_file=/etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_URL.txt
Debug:  emgr_file=/etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_emgr.txt
Debug:  flrtvc_file=/etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_flrtvc.csv
Debug: target=quimby01, remote command=/usr/bin/lslpp -Lcq
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/usr/bin/lslpp -Lcq; echo C_RSH_CMD_RC=\$?"
Debug:  lslpp_file /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_lslpp.txt written
Debug: target=quimby01, remote command=/usr/sbin/emgr -lv3
Debug: c_rsh command=/usr/lpp/bos.sysmgt/nim/methods/c_rsh quimby01 "/usr/sbin/emgr -lv3; echo C_RSH_CMD_RC=\$?"
Error: c_rsh stderr=There is no efix data on this system.


Debug:  emgr_file /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_emgr.txt written
Debug: Utils.execute2 command : /usr/bin/flrtvc.ksh -l /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_lslp
p.txt -e /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_emgr.txt
Debug:     Fileset|Current Version|Type|EFix Installed|Abstract|Unsafe Versions|APARs|Bulletin URL|Download URL|CVSS Base Score|Reboot Required|Last
Update|Fixed In
Debug:     bos.mp64|7.1.3.48|sec||NOT FIXED - There is a vulnerability in sysproc that impacts AIX.|7.1.3.0-7.1.3.48|IV91488|http://aix.software.ibm.
com/aix/efixes/security/sysproc_advisory.asc|ftp://aix.software.ibm.com/aix/efixes/security/sysproc_fix.tar|CVE-2016-8944:6.2|NO|01/30/2017|7100-03-0
9
Debug:     bos.mp64|7.1.3.48|hiper||NOT FIXED - SYSTEM CRASH WHEN USING PROCFS FOR PROCESSES CLOSING MANY FILES|7.1.3.15-7.1.3.50|IV95111|http://www-
01.ibm.com/support/docview.wss?uid=isg1IV95111|http://aix.software.ibm.com/aix/ifixes/iv95111/||YES|06/02/2017|NONE
Debug:     bos.net.tcp.client|7.1.3.47|sec||NOT FIXED - Vulnerabilities in BIND affects AIX|7.1.3.0-7.1.3.47|IV85296|https://aix.software.ibm.com/aix
/efixes/security/bind_advisory12.asc|https://aix.software.ibm.com/aix/efixes/security/bind_fix12.tar|CVE-2016-1285:7.5 CVE-2016-1286:7.5|NO|08/17/201
6|7100-03-08
Debug:     bos.net.tcp.client|7.1.3.47|sec||NOT FIXED - Vulnerabilities in NTP affect AIX|7.1.3.0-7.1.3.47|IV83993|https://aix.software.ibm.com/aix/e
```

- Each recommended eFix has a download URL, building the list of what needs to be downloaded

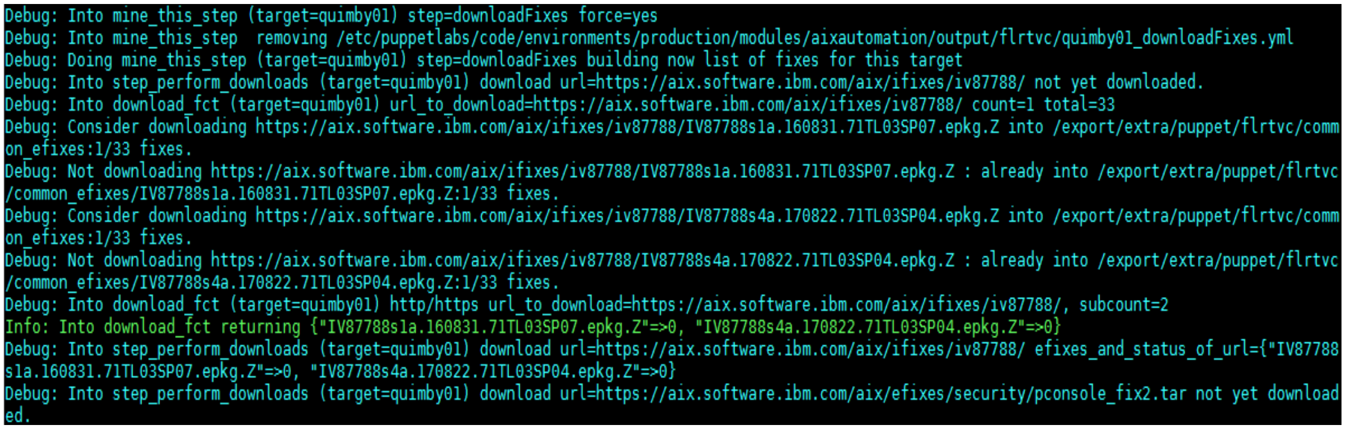


```
Info: Flrtvc step : parseFlrtvc (target=quimby01)
Debug: Into mine_this_step (target=quimby01) step=parseFlrtvc force=yes
Debug: Into mine_this_step  removing /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_parseFlrtvc.yml
Debug: Doing mine_this_step (target=quimby01) step=parseFlrtvc
Debug:  download_url=ftp://aix.software.ibm.com/aix/efixes/security/sysproc_fix.tar
Debug:  download_url=http://aix.software.ibm.com/aix/ifixes/iv95111/
Debug:  download_url=https://aix.software.ibm.com/aix/efixes/security/bind_fix12.tar
Debug:  download_url=https://aix.software.ibm.com/aix/efixes/security/ntp_fix6.tar
Debug:  download_url=https://aix.software.ibm.com/aix/efixes/security/ntp_fix7.tar
Debug:  download_url=ftp://aix.software.ibm.com/aix/efixes/security/bellmail_fix.tar
Debug:  download_url=ftp://aix.software.ibm.com/aix/efixes/security/ntp_fix8.tar
```

- Downloading recommended eFixes into ‘root’ directory defined in ‘init.pp’: ‘/export/extra/puppet/flrtvc’

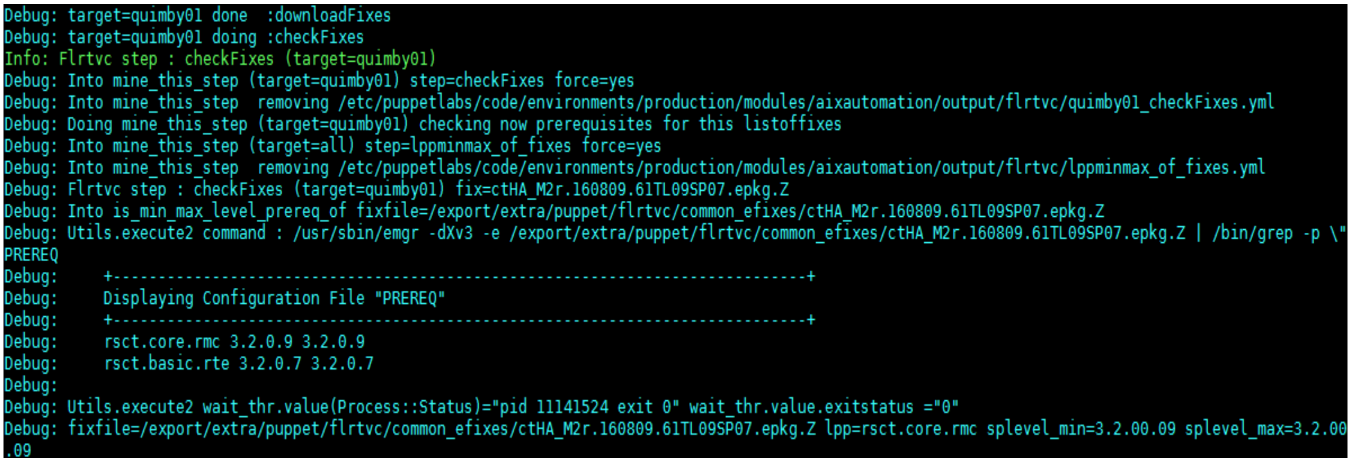


```
Info:  For quimby01, we found 33 different download links over 75 vulnerabilities and 22 filesets.
Info:  See list of advisories mentionned by flrtvc into /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_Adv
isoryFlrtvc.yml
Info:  See list of advisory URLs mentionned by flrtvc into /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_
AdvisoryURLs.yml
Debug: target=quimby01 done  :parseFlrtvc
Debug: target=quimby01 doing :downloadFixes
Info: Flrtvc step : 33 downloadFixes (target=quimby01)
Debug:  url=https://aix.software.ibm.com/aix/ifixes/iv87788/
Debug:  url=https://aix.software.ibm.com/aix/efixes/security/pconsole_fix2.tar
Debug:  url=https://aix.software.ibm.com/aix/efixes/security/pconsole_fix.tar
Debug:  url=https://aix.software.ibm.com/aix/efixes/security/openssl_fix18.tar
Debug:  url=https://aix.software.ibm.com/aix/efixes/security/openssl_fix17.tar
```

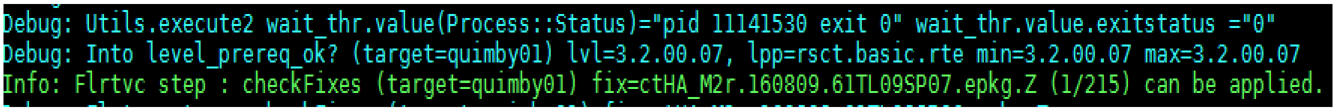


```
Debug: Into mine_this_step (target=quimby01) step=downloadFixes force=yes
Debug: Into mine_this_step  removing /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_downloadFixes.yml
Debug: Doing mine_this_step (target=quimby01) step=downloadFixes building now list of fixes for this target
Debug: Into step_perform_downloads (target=quimby01) download url=https://aix.software.ibm.com/aix/ifixes/iv87788/ not yet downloaded.
Debug: Into download_fct (target=quimby01) url_to_download=https://aix.software.ibm.com/aix/ifixes/iv87788/ count=1 total=33
Debug: Consider downloading https://aix.software.ibm.com/aix/ifixes/iv87788/IV87788s1a.160831.71TL03SP07.epkg.Z into /export/extra/puppet/flrtvc/comm
on_efixes:1/33 fixes.
Debug: Not downloading https://aix.software.ibm.com/aix/ifixes/iv87788/IV87788s1a.160831.71TL03SP07.epkg.Z : already into /export/extra/puppet/flrtvc
/common_efixes/IV87788s1a.160831.71TL03SP07.epkg.Z:1/33 fixes.
Debug: Consider downloading https://aix.software.ibm.com/aix/ifixes/iv87788/IV87788s4a.170822.71TL03SP04.epkg.Z into /export/extra/puppet/flrtvc/comm
on_efixes:1/33 fixes.
Debug: Not downloading https://aix.software.ibm.com/aix/ifixes/iv87788/IV87788s4a.170822.71TL03SP04.epkg.Z : already into /export/extra/puppet/flrtvc
/common_efixes/IV87788s4a.170822.71TL03SP04.epkg.Z:1/33 fixes.
Debug: Into download_fct (target=quimby01) http/https url_to_download=https://aix.software.ibm.com/aix/ifixes/iv87788/, subcount=2
Info: Into download_fct returning {"IV87788s1a.160831.71TL03SP07.epkg.Z"=>0, "IV87788s4a.170822.71TL03SP04.epkg.Z"=>0}
Debug: Into step_perform_downloads (target=quimby01) download url=https://aix.software.ibm.com/aix/ifixes/iv87788/ efixes_and_status_of_url={"IV87788
s1a.160831.71TL03SP07.epkg.Z"=>0, "IV87788s4a.170822.71TL03SP04.epkg.Z"=>0}
Debug: Into step_perform_downloads (target=quimby01) download url=https://aix.software.ibm.com/aix/efixes/security/pconsole_fix2.tar not yet download
ed.
```

- Checking eFixes downloaded to check their installability. Only installable eFixes are kept.

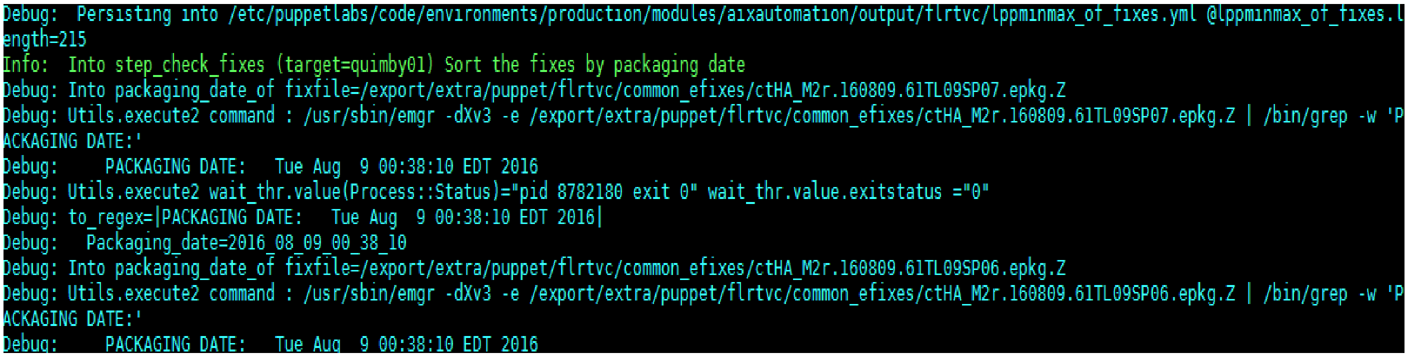


```
Debug: target=quimby01 done  :downloadFixes
Debug: target=quimby01 doing :checkFixes
Info: Flrtvc step : checkFixes (target=quimby01)
Debug: Into mine_this_step (target=quimby01) step=checkFixes force=yes
Debug: Into mine_this_step  removing /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/quimby01_checkFixes.yml
Debug: Doing mine_this_step (target=quimby01) checking now prerequisites for this listoffixes
Debug: Into mine_this_step (target=all) step=lppminmax_of_fixes force=yes
Debug: Into mine_this_step  removing /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/lppminmax_of_fixes.yml
Debug: Flrtvc step : checkFixes (target=quimby01) fix=ctHA_M2r.160809.61TL09SP07.epkg.Z
Debug: Into is_min_max_level_prereq_of fixfile=/export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP07.epkg.Z
Debug: Utils.execute2 command : /usr/sbin/emgr -dXv3 -e /export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP07.epkg.Z | /bin/grep -p \"
PREREQ
Debug:     +-----------------------------------------------------------------------------+
Debug:     Displaying Configuration File "PREREQ"
Debug:     +-----------------------------------------------------------------------------+
Debug:     rsct.core.rmc 3.2.0.9 3.2.0.9
Debug:     rsct.basic.rte 3.2.0.7 3.2.0.7
Debug:
Debug: Utils.execute2 wait_thr.value(Process::Status)="pid 11141524 exit 0" wait_thr.value.exitstatus ="0"
Debug: fixfile=/export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP07.epkg.Z lpp=rsct.core.rmc splevel_min=3.2.00.09 splevel_max=3.2.00
.09
```



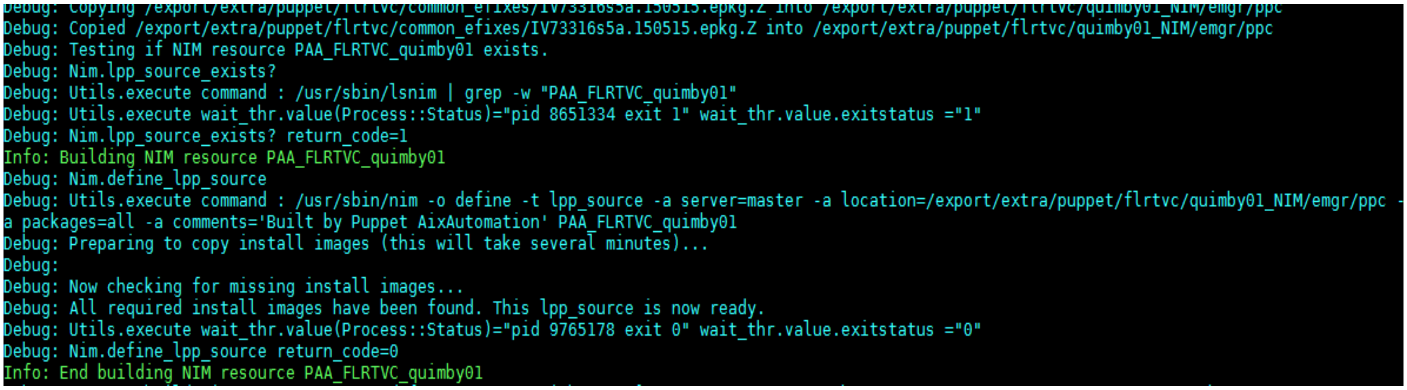
```
Debug: Utils.execute2 wait_thr.value(Process::Status)="pid 11141530 exit 0" wait_thr.value.exitstatus ="0"
Debug: Into level_prereq_ok? (target=quimby01) lvl=3.2.00.07, lpp=rsct.basic.rte min=3.2.00.07 max=3.2.00.07
Info: Flrtvc step : checkFixes (target=quimby01) fix=ctHA_M2r.160809.61TL09SP07.epkg.Z (1/215) can be applied.
```

- Now the list of eFix is build, sort applicable eFixes by packaging date, so that the most recent is applied first.



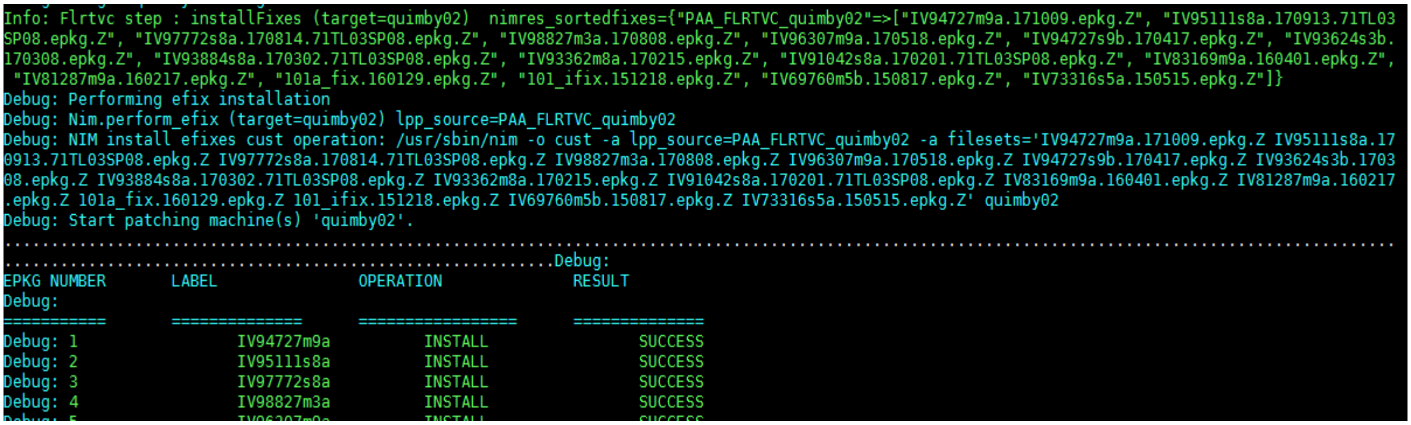
```
Debug:  Persisting into /etc/puppetlabs/code/environments/production/modules/aixautomation/output/flrtvc/lppminmax_of_fixes.yml @lppminmax_of_fixes.l
ength=215
Info:  Into step_check_fixes (target=quimby01) Sort the fixes by packaging date
Debug: Into packaging_date_of fixfile=/export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP07.epkg.Z
Debug: Utils.execute2 command : /usr/sbin/emgr -dXv3 -e /export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP07.epkg.Z | /bin/grep -w 'P
ACKAGING DATE:'
Debug:     PACKAGING DATE:   Tue Aug  9 00:38:10 EDT 2016
Debug: Utils.execute2 wait_thr.value(Process::Status)="pid 8782180 exit 0" wait_thr.value.exitstatus ="0"
Debug: to_regex=|PACKAGING DATE:   Tue Aug  9 00:38:10 EDT 2016|
Debug:   Packaging_date=2016_08_09_00_38_10
Debug: Into packaging_date_of fixfile=/export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP06.epkg.Z
Debug: Utils.execute2 command : /usr/sbin/emgr -dXv3 -e /export/extra/puppet/flrtvc/common_efixes/ctHA_M2r.160809.61TL09SP06.epkg.Z | /bin/grep -w 'P
ACKAGING DATE:'
Debug:     PACKAGING DATE:   Tue Aug  9 00:38:10 EDT 2016
```

- ○ Creating NIM lpp_source resource.



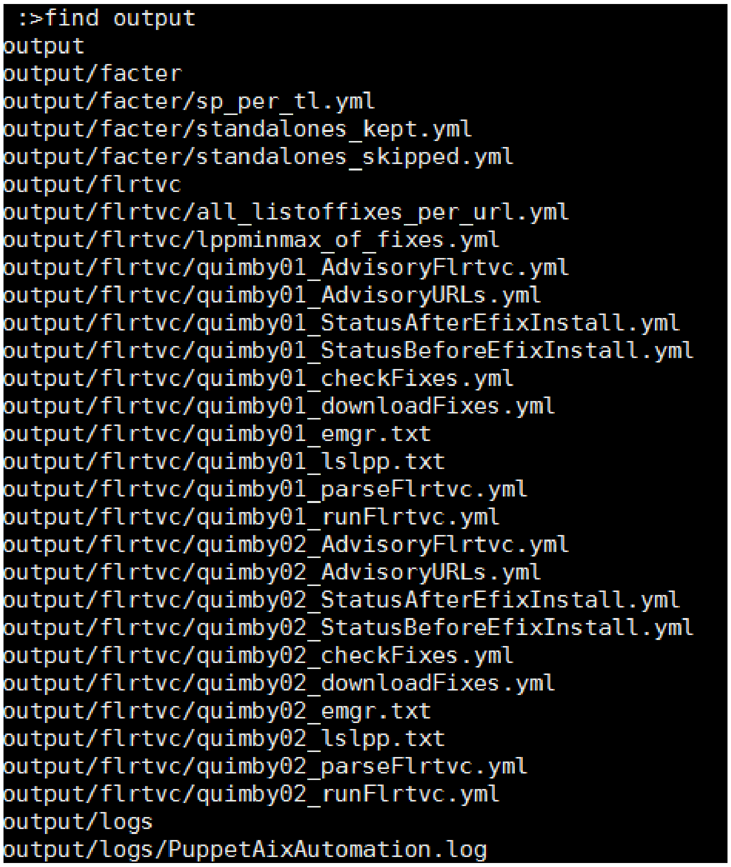
```
Debug: Copying /export/extra/puppet/flrtvc/common_efixes/IV73316s5a.150515.epkg.Z into /export/extra/puppet/flrtvc/quimby01_NIM/emgr/ppc
Debug: Copied /export/extra/puppet/flrtvc/common_efixes/IV73316s5a.150515.epkg.Z into /export/extra/puppet/flrtvc/quimby01_NIM/emgr/ppc
Debug: Testing if NIM resource PAA_FLRTVC_quimby01 exists.
Debug: Nim.lpp_source_exists?
Debug: Utils.execute command : /usr/sbin/lsnim | grep -w "PAA_FLRTVC_quimby01"
Debug: Utils.execute wait_thr.value(Process::Status)="pid 8651334 exit 1" wait_thr.value.exitstatus ="1"
Debug: Nim.lpp_source_exists? return_code=1
Info: Building NIM resource PAA_FLRTVC_quimby01
Debug: Nim.define_lpp_source
Debug: Utils.execute command : /usr/sbin/nim -o define -t lpp_source -a server=master -a location=/export/extra/puppet/flrtvc/quimby01_NIM/emgr/ppc -
a packages=all -a comments='Built by Puppet AixAutomation' PAA_FLRTVC_quimby01
Debug: Preparing to copy install images (this will take several minutes)...
Debug:
Debug: Now checking for missing install images...
Debug: All required install images have been found. This lpp_source is now ready.
Debug: Utils.execute wait_thr.value(Process::Status)="pid 9765178 exit 0" wait_thr.value.exitstatus ="0"
Debug: Nim.define_lpp_source return_code=0
Info: End building NIM resource PAA_FLRTVC_quimby01
```

- ○ Installation of eFixes from NIM lpp_source resource, providing the list of eFixes sorted by packaging date.



```
Info: Flrtvc step : installFixes (target=quimby02)  nimres_sortedfixes={"PAA_FLRTVC_quimby02"=>["IV94727m9a.171009.epkg.Z", "IV95111s8a.170913.71TL03
SP08.epkg.Z", "IV97772s8a.170814.71TL03SP08.epkg.Z", "IV98827m3a.170808.epkg.Z", "IV96307m9a.170518.epkg.Z", "IV94727s9b.170417.epkg.Z", "IV93624s3b.
170308.epkg.Z", "IV93884s8a.170302.71TL03SP08.epkg.Z", "IV93362m8a.170215.epkg.Z", "IV91042s8a.170201.71TL03SP08.epkg.Z", "IV83169m9a.160401.epkg.Z",
 "IV81287m9a.160217.epkg.Z", "101a_fix.160129.epkg.Z", "101_ifix.151218.epkg.Z", "IV69760m5b.150817.epkg.Z", "IV73316s5a.150515.epkg.Z"]}
Debug: Performing efix installation
Debug: Nim.perform_efix (target=quimby02) lpp_source=PAA_FLRTVC_quimby02
Debug: NIM install efixes cust operation: /usr/sbin/nim -o cust -a lpp_source=PAA_FLRTVC_quimby02 -a filesets='IV94727m9a.171009.epkg.Z IV95111s8a.17
0913.71TL03SP08.epkg.Z IV97772s8a.170814.71TL03SP08.epkg.Z IV98827m3a.170808.epkg.Z IV96307m9a.170518.epkg.Z IV94727s9b.170417.epkg.Z IV93624s3b.1703
08.epkg.Z IV93884s8a.170302.71TL03SP08.epkg.Z IV93362m8a.170215.epkg.Z IV91042s8a.170201.71TL03SP08.epkg.Z IV83169m9a.160401.epkg.Z IV81287m9a.160217
.epkg.Z 101a_fix.160129.epkg.Z 101_ifix.151218.epkg.Z IV69760m5b.150817.epkg.Z IV73316s5a.150515.epkg.Z' quimby02
Debug: Start patching machine(s) 'quimby02'.
..............................................................................................................................................
.........................................................Debug:
EPKG NUMBER       LABEL               OPERATION              RESULT
Debug:
===========       ==============      =================      ==============
Debug: 1                IV94727m9a          INSTALL                SUCCESS
Debug: 2                IV95111s8a          INSTALL                SUCCESS
Debug: 3                IV97772s8a          INSTALL                SUCCESS
Debug: 4                IV98827m3a          INSTALL                SUCCESS
Debug: 5                IV96307m9a          INSTALL                SUCCESS
```

Output files from FLRTV update



```
 :>find output
output
output/facter
output/facter/sp_per_tl.yml
output/facter/standalones_kept.yml
output/facter/standalones_skipped.yml
output/flrtvc
output/flrtvc/all_listoffixes_per_url.yml
output/flrtvc/lppminmax_of_fixes.yml
output/flrtvc/quimby01_AdvisoryFlrtvc.yml
output/flrtvc/quimby01_AdvisoryURLs.yml
output/flrtvc/quimby01_StatusAfterEfixInstall.yml
output/flrtvc/quimby01_StatusBeforeEfixInstall.yml
output/flrtvc/quimby01_checkFixes.yml
output/flrtvc/quimby01_downloadFixes.yml
output/flrtvc/quimby01_emgr.txt
output/flrtvc/quimby01_lslpp.txt
output/flrtvc/quimby01_parseFlrtvc.yml
output/flrtvc/quimby01_runFlrtvc.yml
output/flrtvc/quimby02_AdvisoryFlrtvc.yml
output/flrtvc/quimby02_AdvisoryURLs.yml
output/flrtvc/quimby02_StatusAfterEfixInstall.yml
output/flrtvc/quimby02_StatusBeforeEfixInstall.yml
output/flrtvc/quimby02_checkFixes.yml
output/flrtvc/quimby02_downloadFixes.yml
output/flrtvc/quimby02_emgr.txt
output/flrtvc/quimby02_lslpp.txt
output/flrtvc/quimby02_parseFlrtvc.yml
output/flrtvc/quimby02_runFlrtvc.yml
output/logs
output/logs/PuppetAixAutomation.log
```

Data results of the facter execution steps.

Data results for each step of FLRTVC update for standalone client quimby01.

AIX Level and fix status of quimby01 before and after the installation.

Data results for each step of FLRTVC update for standalone client quimby02.

AIX Level and fix status of quimby02 before and after the installation.

Puppet Aix Automation log file

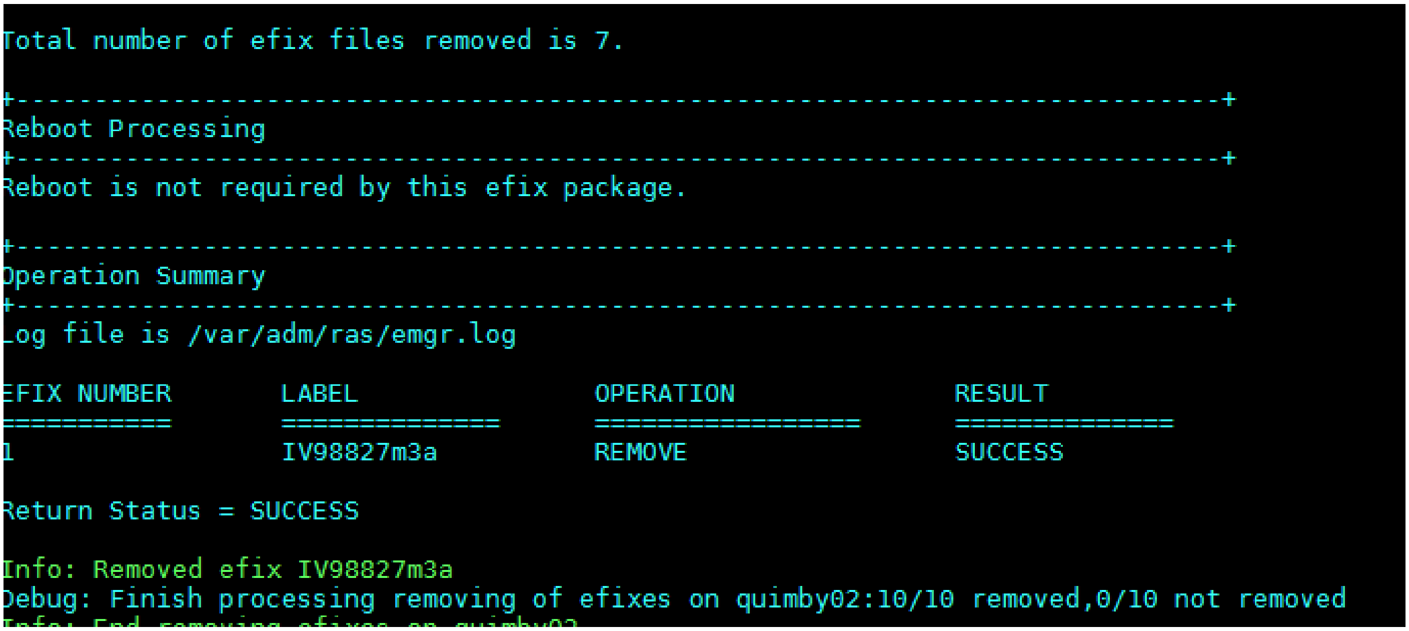
Remove FLRTVC eFixes.

To remove AIX systems eFixes: from the previous install FLRTVC manifest file, just update "**ensure"** tag to "**absent".**

```
class aixautomation {
  # This rule enables to remove eFix through flrtvc provider,
  # on the quimby01 and quimby02 systems so that they are fixed as best as possible.
  # Setting ensure to 'absent'
  fix { "task fix remove":
    provider => flrtvc,
    name     => " task fix remove ",
    ensure   => absent,
    to_step  => "installResource",
    targets  => "quimby01 quimby02",
    force    => "yes",
    root     => "/export/extra/puppet/flrtvc",
  }
 }
```

ensure: Control the task – if ensure = "**present**" : eFixes are installed– if ensure = "**absent**" : eFixes are removed.

targets: List of AIX standalone clients from which eFixes will be removed.



```
Total number of efix files removed is 7.

+-----------------------------------------------------------------------------+
Reboot Processing
+-----------------------------------------------------------------------------+
Reboot is not required by this efix package.

+-----------------------------------------------------------------------------+
Operation Summary
+-----------------------------------------------------------------------------+
Log file is /var/adm/ras/emgr.log

EFIX NUMBER       LABEL               OPERATION              RESULT
===========       ==============      =================      ==============
1                 IV98827m3a          REMOVE                 SUCCESS

Return Status = SUCCESS

Info: Removed efix IV98827m3a
Debug: Finish processing removing of efixes on quimby02:10/10 removed,0/10 not removed
Info: End removing efixes on quimby02
```

For more information…

For more information about AIX and VIOS Automation with Puppet and to know all the attributes available for each custom type, refer to README.md file from Github project repository:

https://github.com/aixoss/aix-puppet/aixautomation/blob/master/README.md

Table of contents

END OF DOCUMENT