Server Packages Summary Report Generator

For

For Application Stability, Security Audits and Ensuring Uniform Deployments across Environment

Version 1.10.1

Prepared by

Santosh Kulkarni

System Admin / Automation Enthusiast
WZ Nanded DO Cell 9960708564

Revision History

Date	Version	Description	Author
October 7, 2021	1.10.1	Server Packages Summary Report Generator	Santosh Kulkarni

Disclaimer

This is just fact gathering utility it doesn't modify any setting on any target server, or install any packages/scripts or create any files on target server. Using this utility is as good as using command **#rpm -qa** command on Red Hat base target servers and processing that output on localhost with high accuracy.

Synonyms

Target Server: The server where we want generate package summary report.
Host or Localhost: The Linux Machin which is running this utility.
Division Name or Site Name: A collection of target nodes identified by string.
Zone: A collection of sites identified by georgical location.
Docode or Division Code: A Number associated with site name identified by digit.

Server Packages Summary Report Generator

Table of Contents

Contents

1. Introduction

To deliver stable and consistent performance through application services. Any application depends on the number of application packages, and sometimes those application packages are dependent on the system packages also. Hence, it's part of periodic audits to ensure stable and secure services by ensuring packages are installed and are with correct versions on the server. Although, this is quite a difficult process. This utility is meant to perform the same with ease in a presentable way. Let's explore following points

- Purpose
- Objectives

a. Purpose

Whenever we have a scenario of unstable performance, breakdown of application services. First step towards restores and ensure stable services is verifying all application and dependent packages are installed and also at desired version. This utility is designed for the same purpose: it is super lite and robust. We only need to supply inventory configuration; the rest will do this itself. We will see this in detail in **How It Works**.

b. Objectives

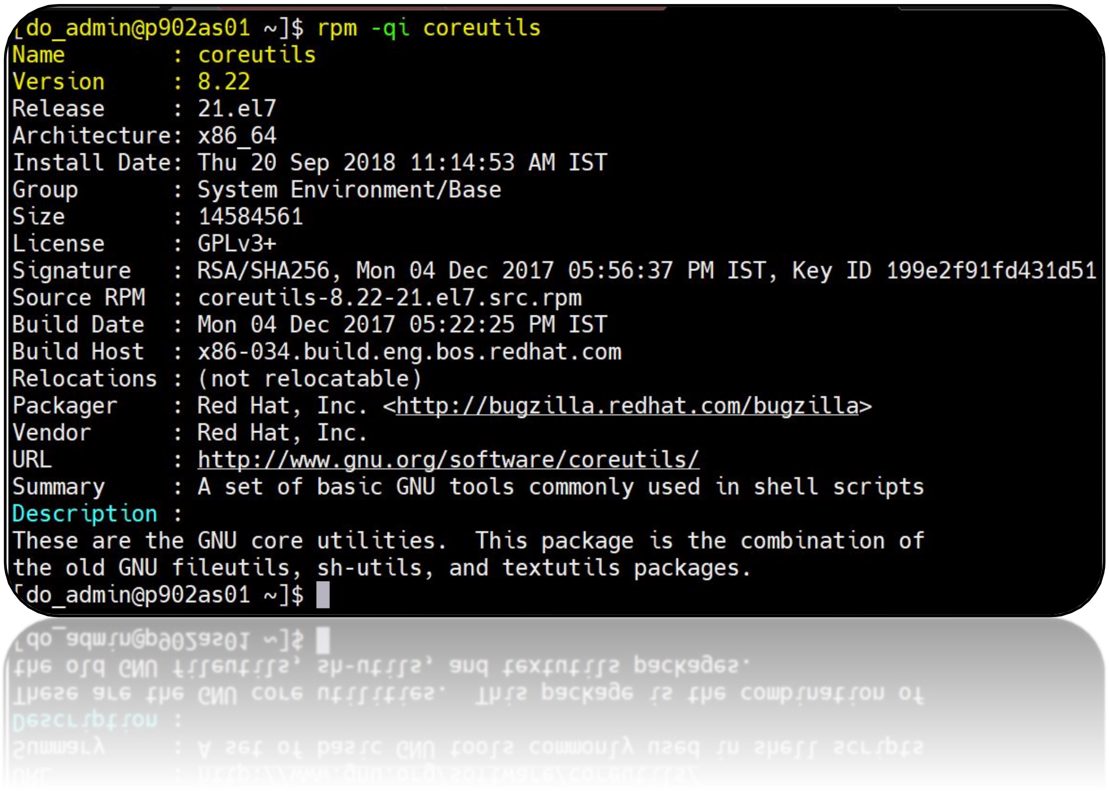
Traditional way of verifying installation status along with the version packages is quite difficult. So, the objective is to make it simple enough so that it can be part of regular stability and security audits for the application.

2. Traditional ways of package verification.

a. *Verification of packages with default tools.*

Every server has different tools for managing/verifying packages. Like Ubuntu, the server has an apt tool to manage/verify packages. We have Red Hat/CentOS based target Servers which use **RPM (Red Hat Package Manager)** for package management. And **Yellowdog Updater, Modified (YUM)** for managing packages with dependency resolution. We will see how we check packages installation status. And in case of installed we will check at which version they are installed with. We will check the installation status with the RPM tool and then with the YUM tool. For a sample, we will check coreutils packages.

Syntax: rpm <command flags> <package_name>

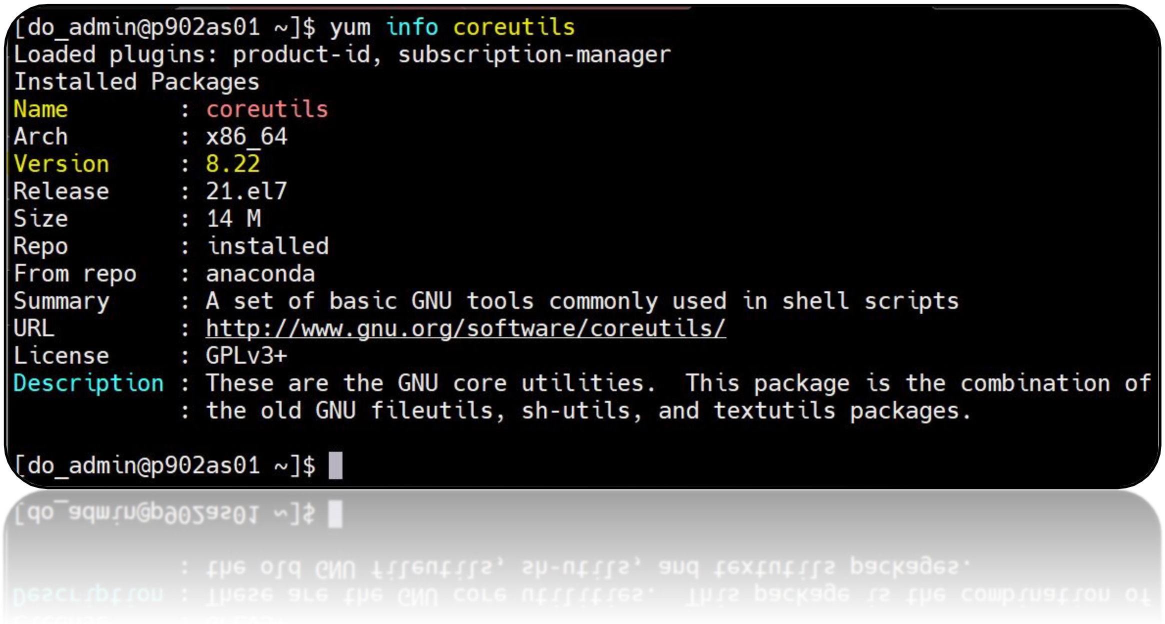


```
[do_admin@p902as01 ~]$ rpm -qi coreutils
Name        : coreutils
Version     : 8.22
Release     : 21.el7
Architecture: x86_64
Install Date: Thu 20 Sep 2018 11:14:53 AM IST
Group       : System Environment/Base
Size        : 14584561
License     : GPLv3+
Signature   : RSA/SHA256, Mon 04 Dec 2017 05:56:37 PM IST, Key ID 199e2f91fd431d51
Source RPM  : coreutils-8.22-21.el7.src.rpm
Build Date  : Mon 04 Dec 2017 05:22:25 PM IST
Build Host  : x86-034.build.eng.bos.redhat.com
Relocations : (not relocatable)
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Vendor      : Red Hat, Inc.
URL         : http://www.gnu.org/software/coreutils/
Summary     : A set of basic GNU tools commonly used in shell scripts
Description :
These are the GNU core utilities.  This package is the combination of
the old GNU fileutils, sh-utils, and textutils packages.
[do_admin@p902as01 ~]$ 
```

As we see on our rpm command, the query returned with coreutils package information. Which confirms coreutils package is installed on the server and having version 8.22

Let's see the same with YUM command

Syntax: yum info <package name>



```
[do_admin@p902as01 ~]$ yum info coreutils
Loaded plugins: product-id, subscription-manager
Installed Packages
Name        : coreutils
Arch        : x86_64
Version     : 8.22
Release     : 21.el7
Size        : 14 M
Repo        : installed
From repo   : anaconda
Summary     : A set of basic GNU tools commonly used in shell scripts
URL         : http://www.gnu.org/software/coreutils/
License     : GPLv3+
Description : These are the GNU core utilities.  This package is the combination of
            : the old GNU fileutils, sh-utils, and textutils packages.

[do_admin@p902as01 ~]$ 
```

As we can see, the yum info command returns with the package version, which confirms that the package is installed and has version 8.22.

We have checked the coreutils package with RPM and YUM tools. This is good if we need to check with a few packages at once. But what if we have to check thousands of packages to ensure the desired state of application as a regular practice. We will check this further.

3. Utility information.

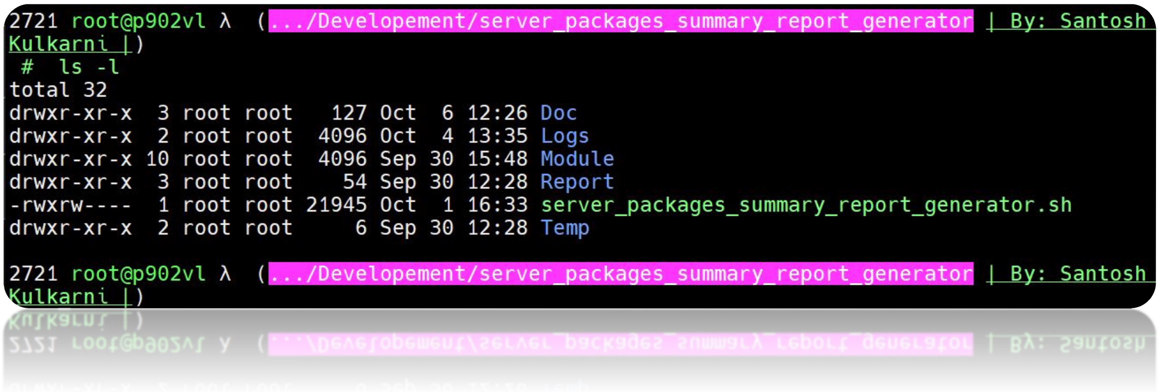
This section will give a brief understanding of the utility

a. Where it will work

This utility can be used on any Linux host. In this case I am using this on Upload **VM Machine (RHEL7)** which is used for day-to-day batch upload. It just gathers the Package's information from the target server and rest on the magic is done on the host it is running from.

b. Utility folder information and view

Users need to copy this utility folder in any preferred path. The utility will work on only Linux hosts. The Utility Folder will look like below.



```
2721 root@p902vl λ  (.../Developement/server_packages_summary_report_generator | By: Santosh
Kulkarni |)
 #  ls -l
total 32
drwxr-xr-x  3 root root   127 Oct  6 12:26 Doc
drwxr-xr-x  2 root root  4096 Oct  4 13:35 Logs
drwxr-xr-x 10 root root  4096 Sep 30 15:48 Module
drwxr-xr-x  3 root root    54 Sep 30 12:28 Report
-rwxrw----  1 root root 21945 Oct  1 16:33 server_packages_summary_report_generator.sh
drwxr-xr-x  2 root root     6 Sep 30 12:28 Temp

2721 root@p902vl λ  (.../Developement/server_packages_summary_report_generator | By: Santosh
Kulkarni |)
```

Utility folder structure and file information

Doc: Utility documents path
Logs: This is where utility can store its logs files

Module: This is where the utility file resides. Package fact gathering and post report generation is done here.
Report: This is where the package status report XLSX file is generated and stored.

server_packages_summary_report_generator.sh: The shell which starts the utility
Temp: A temporary path for temp files.

4. How Utility will work

This is perhaps the most important section of the document. This utility will do packages report generation in five stages as below.

a. **Checking target server connectivity:**

If connectivity is found between host and target server, only then the target server is considered for package facts gathering.

b. **Packages information gathering**:

This is the only stage where it will be in contact with the Target server. This is just like using #rpm -qa command on target server. Which is used to see package information and this is just a fact gathering. It doesn't change anything on the server nor create any residual files on the target server.

c. **Generation of packages fact file:**

This is the second stage where the utility will create packages fact files for the target server on the Local host.

d. **Checking packages status:**

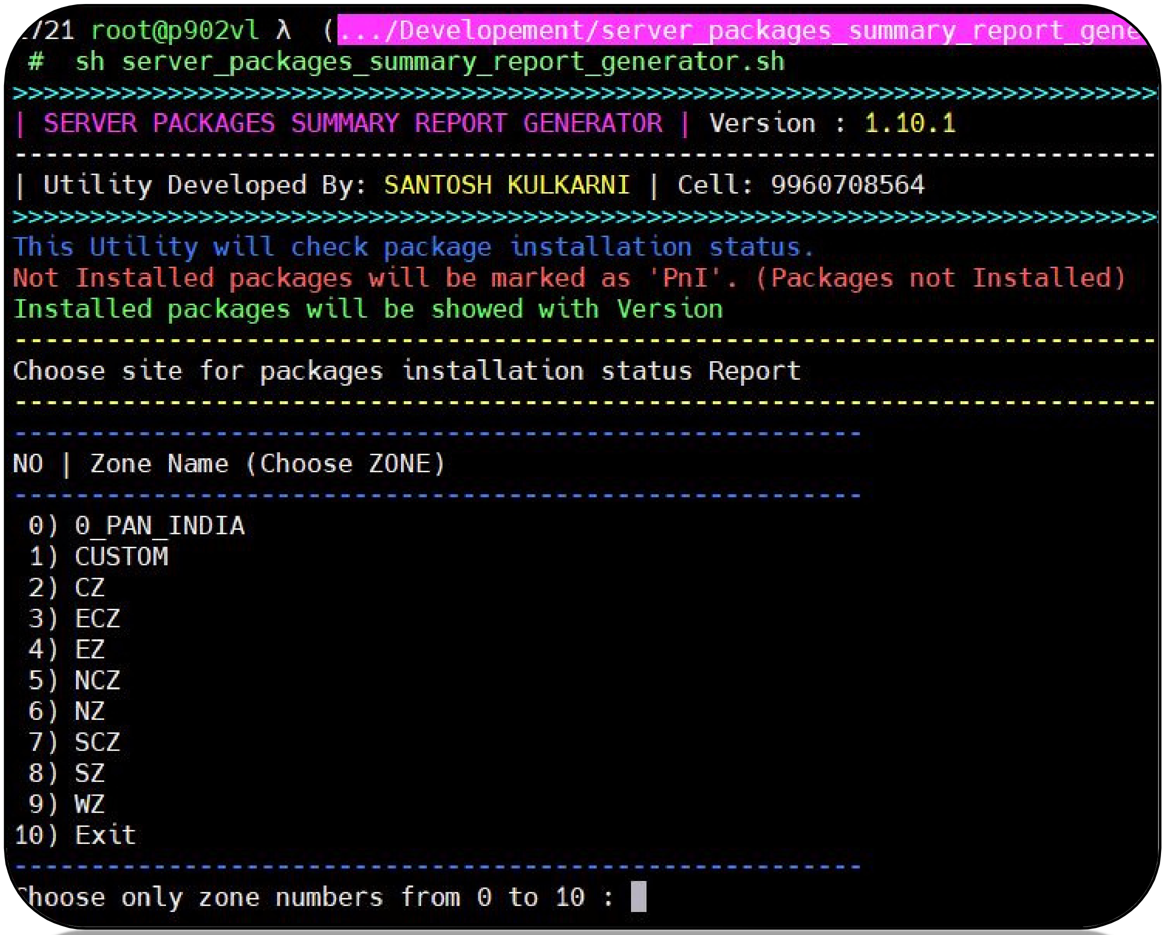
This utility has an inventory of packages to check in the target server fact file. In this stage it will only check whether the package is installed or not with the target packages fact file which is generated on the second stage. Everything is done on localhost in case of package installed then the package version information is gathered. And if package is not installed it will be marked as **PnI(Package not installed).**

e. **Final XLSX Report Generation**:

This is the final stage where the utility will generate XLSX Report. The report will be compiled as per the site name, which is Division Name. Whenever we query on the site, it will query on all 7 servers on the site. And package information status of all 7 nodes of the site is shown in the output report as a site name Tab XLSX output report. If a query is done with 2 or more sites at once, then the output XLSX report will have a separate XLSX Tab with Division Name for each site. Let's combine a query for site NANDED and AURANGABAD at once, then output XLSX report will have 2 Tabs with the names NANDED and AURANGABAD. Which will contain information of all 7 nodes of their respective location**.**

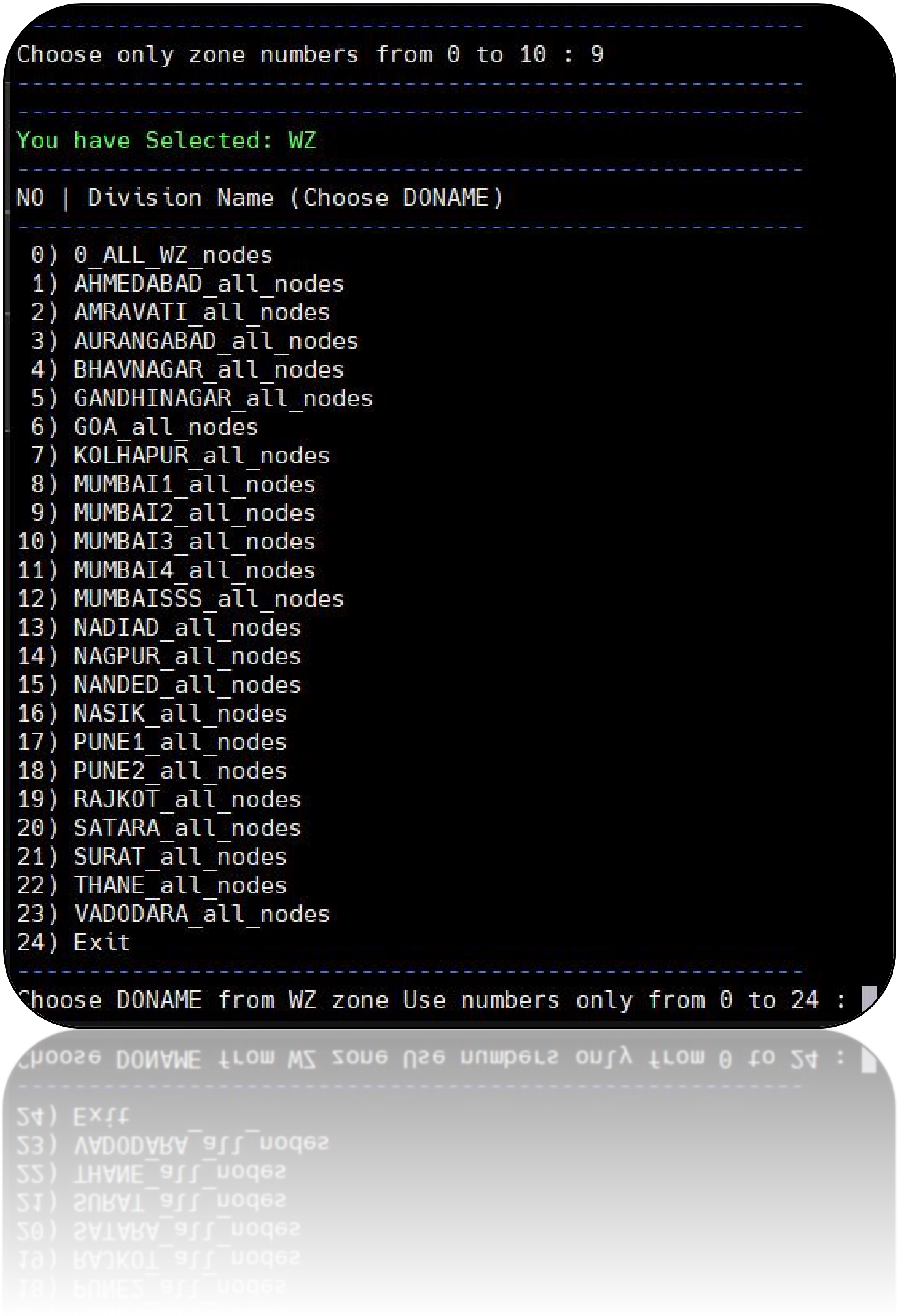
5. Utility usage / How to use.

Let's use the utility now. Navigate to the path where we copied the utility through the terminal. And run the utility by invoking a shell file.



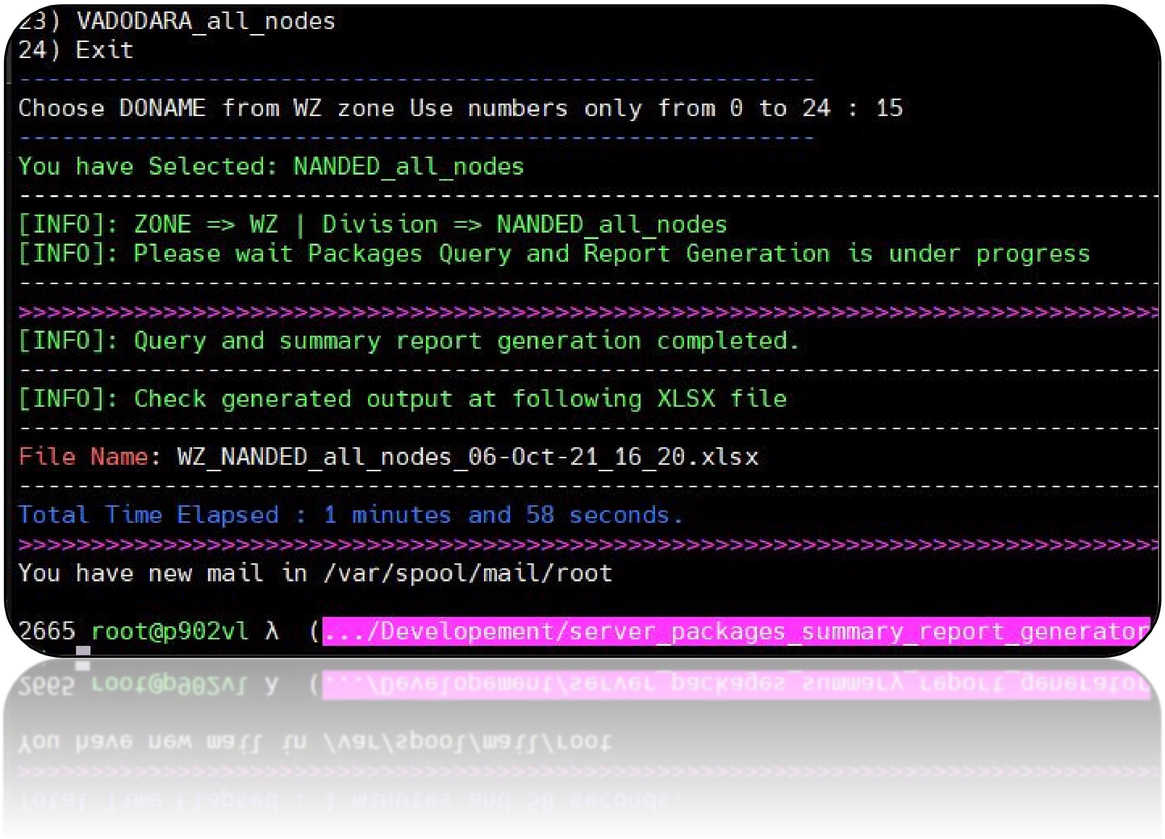
```
/21 root@p902vl λ  (.../Developement/server_packages_summary_report_gene
 #  sh server_packages_summary_report_generator.sh
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
| SERVER PACKAGES SUMMARY REPORT GENERATOR | Version : 1.10.1
---------------------------------------------------------------------------
| Utility Developed By: SANTOSH KULKARNI | Cell: 9960708564
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
This Utility will check package installation status.
Not Installed packages will be marked as 'PnI'. (Packages not Installed)
Installed packages will be showed with Version
---------------------------------------------------------------------------
Choose site for packages installation status Report
---------------------------------------------------------------------------
-------------------------------------------------------
NO | Zone Name (Choose ZONE)
-------------------------------------------------------
 0) 0_PAN_INDIA
 1) CUSTOM
 2) CZ
 3) ECZ
 4) EZ
 5) NCZ
 6) NZ
 7) SCZ
 8) SZ
 9) WZ
10) Exit
-------------------------------------------------------
hoose only zone numbers from 0 to 10 :
```

As we can see in the above picture, we need to choose the zone of site (Division) where we want to generate packages summary reports. We will be choosing Western Zone (WZ) with option no 9.



```
-------------------------------------------------------
Choose only zone numbers from 0 to 10 : 9
-------------------------------------------------------
-------------------------------------------------------
You have Selected: WZ
-------------------------------------------------------
NO | Division Name (Choose DONAME)
-------------------------------------------------------
 0) 0_ALL_WZ_nodes
 1) AHMEDABAD_all_nodes
 2) AMRAVATI_all_nodes
 3) AURANGABAD_all_nodes
 4) BHAVNAGAR_all_nodes
 5) GANDHINAGAR_all_nodes
 6) GOA_all_nodes
 7) KOLHAPUR_all_nodes
 8) MUMBAI1_all_nodes
 9) MUMBAI2_all_nodes
10) MUMBAI3_all_nodes
11) MUMBAI4_all_nodes
12) MUMBAISSS_all_nodes
13) NADIAD_all_nodes
14) NAGPUR_all_nodes
15) NANDED_all_nodes
16) NASIK_all_nodes
17) PUNE1_all_nodes
18) PUNE2_all_nodes
19) RAJKOT_all_nodes
20) SATARA_all_nodes
21) SURAT_all_nodes
22) THANE_all_nodes
23) VADODARA_all_nodes
24) Exit
-------------------------------------------------------
Choose DONAME from WZ zone Use numbers only from 0 to 24 :
```

Now the utility is asking to choose the site name (Division Name) under the selected zone, which WZ in this case. We will be inputting no 15 which will select site NANDED_all_nodes. For package summary report.



```
23) VADODARA_all_nodes
24) Exit
-------------------------------------------------------
Choose DONAME from WZ zone Use numbers only from 0 to 24 : 15
-------------------------------------------------------
You have Selected: NANDED_all_nodes
-------------------------------------------------------------------------------
[INFO]: ZONE => WZ | Division => NANDED_all_nodes
[INFO]: Please wait Packages Query and Report Generation is under progress
-------------------------------------------------------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[INFO]: Query and summary report generation completed.
-------------------------------------------------------------------------------
[INFO]: Check generated output at following XLSX file
-------------------------------------------------------------------------------
File Name: WZ_NANDED_all_nodes_06-Oct-21_16_20.xlsx
-------------------------------------------------------------------------------
Total Time Elapsed : 1 minutes and 58 seconds.
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
You have new mail in /var/spool/mail/root

2665 root@p902vl λ  (.../Developement/server_packages_summary_report_generator
```

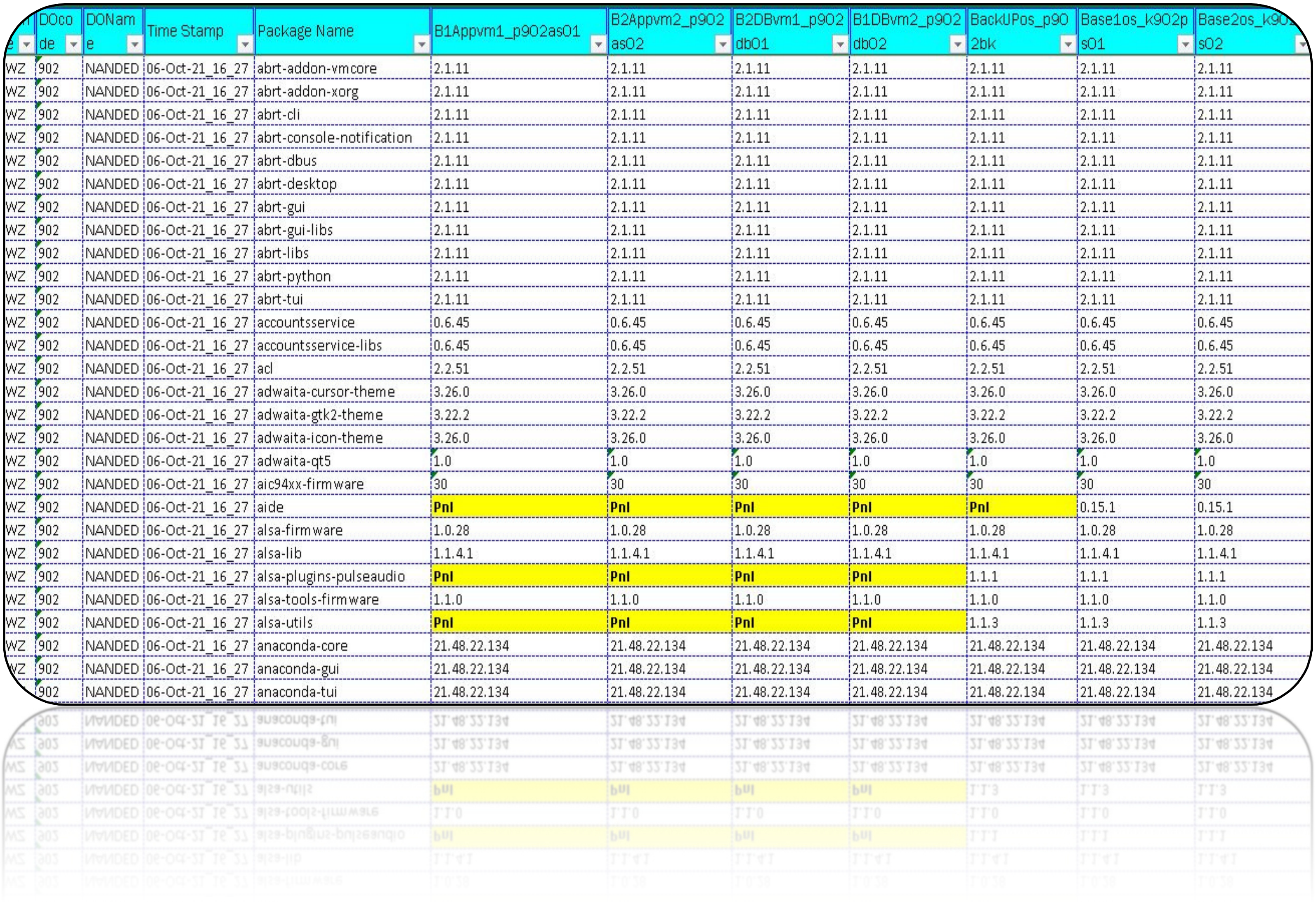
As we can see in the above picture. We had selected a NANDED site and compiled XLSX generation completed which only took 1 Minute and 58 Seconds. Imagining a manual task for doing the same. Connecting Servers, Getting Package Information, compiling reports based on individual nodes And Creating Final XLSX Report. Would have taken lots of time and involved the possibility of inaccuracy and manual errors. This utility will do this again and again with same level of accuracy and efficiency.

Now we will see the generated XLSX report. As shown in above picture output XLSX file with name **WZ_NANDED_all_nodes_06-Oct-21_16_20.xlsx** created and has a stamp for selected zone and site name with date and query timestamp.

Just for the better readability, I have pasted the XLSX report picture on last page . Packages Marked in PnI in yellow background are the packages not installed. Columns from F to L represent all NODES of the site. Like column "F" header name.

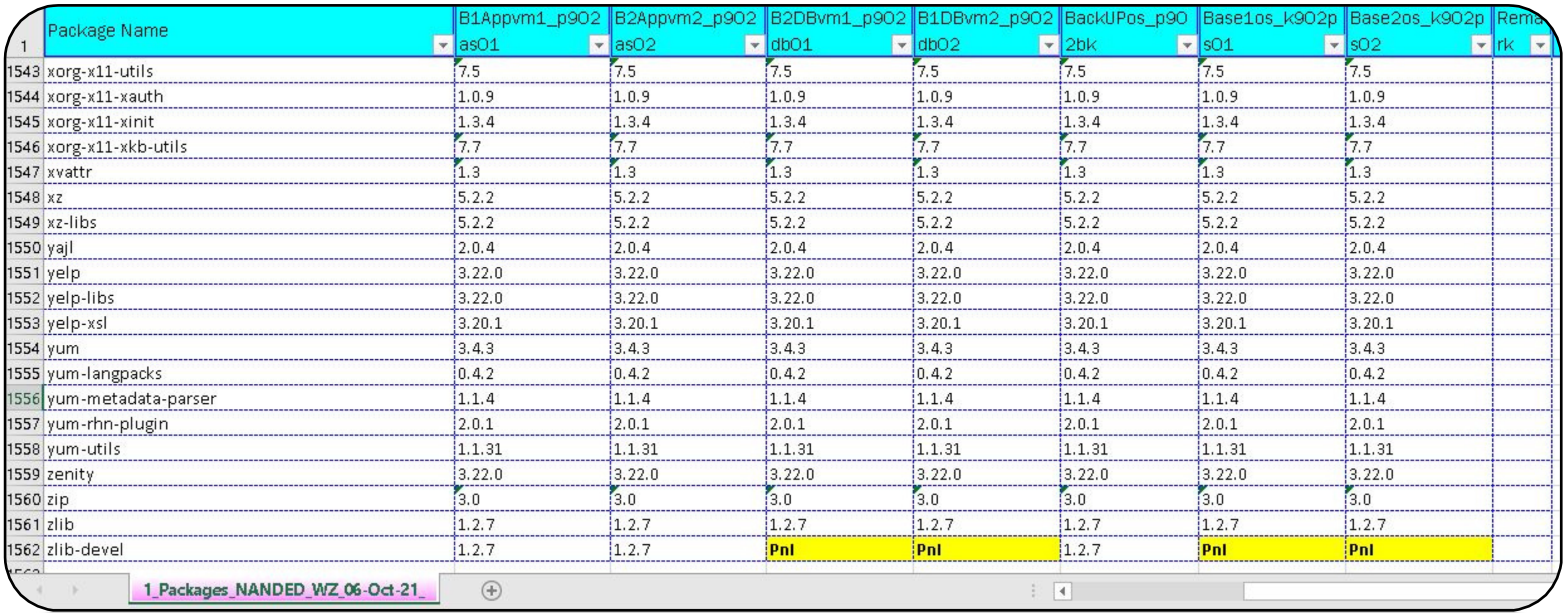
B1Appvm1_p902as01 "B1Appvm1" in column name denotes server role and p902as01 is server hostname, which is **p902as01** in this case. Same with **column "G"** B2Appvm2_p902as02 **Server role B2Appvm2** and hostname is **p902as02,** this follows till column "L" till completion of all nodes of the site.

Report generation done for 1561 packages on each of 7 nodes of the sites. So, this way we have checked 10927 packages across 7 nodes. Now we have confirmed with this report whether x package from 1561 packages is installed or not across all nodes of site(Division) . And in case of installed, we have its version number also. And all this is done in less than a common coffee break time. Please check XLSX report below.



ne	DOcode	DOName	Time Stamp	Package Name	B1Appvm1_p902as01	B2Appvm2_p902as02	B2DBvm1_p902db01	B1DBvm2_p902db02	BackUPos_p902bk	Base1os_k902ps01	Base2os_k90 s02
WZ	902	NANDED	06-Oct-21_16_27	abrt-addon-vmcore	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-addon-xorg	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-cli	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-console-notification	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-dbus	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-desktop	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-gui	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-gui-libs	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-libs	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-python	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	abrt-tui	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11	2.1.11
WZ	902	NANDED	06-Oct-21_16_27	accountsservice	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45
WZ	902	NANDED	06-Oct-21_16_27	accountsservice-libs	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45	0.6.45
WZ	902	NANDED	06-Oct-21_16_27	acl	2.2.51	2.2.51	2.2.51	2.2.51	2.2.51	2.2.51	2.2.51
WZ	902	NANDED	06-Oct-21_16_27	adwaita-cursor-theme	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0
WZ	902	NANDED	06-Oct-21_16_27	adwaita-gtk2-theme	3.22.2	3.22.2	3.22.2	3.22.2	3.22.2	3.22.2	3.22.2
WZ	902	NANDED	06-Oct-21_16_27	adwaita-icon-theme	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0	3.26.0
WZ	902	NANDED	06-Oct-21_16_27	adwaita-qt5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
WZ	902	NANDED	06-Oct-21_16_27	aic94xx-firmware	30	30	30	30	30	30	30
WZ	902	NANDED	06-Oct-21_16_27	aide	**PnI**	**PnI**	**PnI**	**PnI**	**PnI**	0.15.1	0.15.1
WZ	902	NANDED	06-Oct-21_16_27	alsa-firmware	1.0.28	1.0.28	1.0.28	1.0.28	1.0.28	1.0.28	1.0.28
WZ	902	NANDED	06-Oct-21_16_27	alsa-lib	1.1.4.1	1.1.4.1	1.1.4.1	1.1.4.1	1.1.4.1	1.1.4.1	1.1.4.1
WZ	902	NANDED	06-Oct-21_16_27	alsa-plugins-pulseaudio	**PnI**	**PnI**	**PnI**	**PnI**	1.1.1	1.1.1	1.1.1
WZ	902	NANDED	06-Oct-21_16_27	alsa-tools-firmware	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0
WZ	902	NANDED	06-Oct-21_16_27	alsa-utils	**PnI**	**PnI**	**PnI**	**PnI**	1.1.3	1.1.3	1.1.3
WZ	902	NANDED	06-Oct-21_16_27	anaconda-core	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134
WZ	902	NANDED	06-Oct-21_16_27	anaconda-gui	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134
	902	NANDED	06-Oct-21_16_27	anaconda-tui	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134	21.48.22.134

The below picture will show the bottom of the xlsx report looks like. Note that separate tab for site name which have information of all nodes in that site.



1	Package Name	B1Appvm1_p902as01	B2Appvm2_p902as02	B2DBvm1_p902db01	B1DBvm2_p902db02	BackUPos_p902bk	Base1os_k902ps01	Base2os_k902ps02	Remark
1543	xorg-x11-utils	7.5	7.5	7.5	7.5	7.5	7.5	7.5	
1544	xorg-x11-xauth	1.0.9	1.0.9	1.0.9	1.0.9	1.0.9	1.0.9	1.0.9	
1545	xorg-x11-xinit	1.3.4	1.3.4	1.3.4	1.3.4	1.3.4	1.3.4	1.3.4	
1546	xorg-x11-xkb-utils	7.7	7.7	7.7	7.7	7.7	7.7	7.7	
1547	xvattr	1.3	1.3	1.3	1.3	1.3	1.3	1.3	
1548	xz	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	
1549	xz-libs	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	5.2.2	
1550	yajl	2.0.4	2.0.4	2.0.4	2.0.4	2.0.4	2.0.4	2.0.4	
1551	yelp	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	
1552	yelp-libs	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	
1553	yelp-xsl	3.20.1	3.20.1	3.20.1	3.20.1	3.20.1	3.20.1	3.20.1	
1554	yum	3.4.3	3.4.3	3.4.3	3.4.3	3.4.3	3.4.3	3.4.3	
1555	yum-langpacks	0.4.2	0.4.2	0.4.2	0.4.2	0.4.2	0.4.2	0.4.2	
1556	yum-metadata-parser	1.1.4	1.1.4	1.1.4	1.1.4	1.1.4	1.1.4	1.1.4	
1557	yum-rhn-plugin	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1	
1558	yum-utils	1.1.31	1.1.31	1.1.31	1.1.31	1.1.31	1.1.31	1.1.31	
1559	zenity	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	3.22.0	
1560	zip	3.0	3.0	3.0	3.0	3.0	3.0	3.0	
1561	zlib	1.2.7	1.2.7	1.2.7	1.2.7	1.2.7	1.2.7	1.2.7	
1562	zlib-devel	1.2.7	1.2.7	**Pnl**	**Pnl**	1.2.7	**Pnl**	**Pnl**	