

The issue I worked on:

- zopen info -r not showing Build Dependencies

```
09:39:35 RC=(0) [SYS] bash-5.2$ zopen info git
==> git: STABLE 2.48.1
Description:      git on z/OS
==> Package
Version:          2.48.1
Release:          20250128_192801
Buildline:        STABLE
Categories:       development source_control
GitHub:           https://github.com/zopencommunity/gitport
License:          https://github.com/zopencommunity/gitport/blob/main/patches/LICENSE
zopen license:    https://github.com/zopencommunity/gitport/blob/main/LICENSE
==> Installation Details
Installed:        Yes
Installation Path: /hewitt/zopentools/zopen_repo/usr/local/zopen/git/git-heads.v2.48.1.20250128_192801.zos
Installation Size: 210.803 MB
==> Test Status
Passed:           89/92 (96%)
==> Package Details
Download Size:    52.67 MB
Commit SHA:       be3a731b0efd1c47adf8259f09c1e584c493c109
Community SHA:    f93ff170b93a1782659637824b25923245ac9dd1
==> Dependencies
Build:            autoconf, automake, bash, check_clang, check_python, coreutils, curl, curl, diffutils, expat, gawk, gettext, git, git, gzip, help2man, jq,
libpcre2, m4, make, ncurses, openssl, perl, sed, sed, tar, tar, texinfo, xz, zlib, zoslib, zoslib
Runtime:          bash, less, ncurses, perl

09:39:40 RC=(0) [SYS] bash-5.2$ zopen info git -r
==> git (Build 2873) - (STABLE) (Not Installed)
==> Package
Version:          2.48.1.0
Release:          20250128_192801
Buildline:        _gitport_2873
Categories:       development source_control
GitHub:           https://zopencommunity/gitport/releases/download
Installed:        No
==> Test Status
Passed:           89/92 (96%)
==> Package Details
Download Size:    25.80 MB
Install Size:     52.67 MB
Community SHA:    f93ff170b93a1782659637824b25923245ac9dd1
==> Dependencies
Runtime:          bash less ncurses perl
```

- Link: <https://github.com/zopencommunity/meta/issues/954>
- Description of issue: When using the zopen info command, the build dependencies will appear. However, when we add -r at the end, it will not show the build dependencies.

The initial solution I implemented:

- Since we know that -r will force a remote lookup, it means that when we run the “zopen info -r” it will access the remote JSON_CACHE to retrieve the build dependencies data.
- When looking into the zopen-info file with the code for that command, I saw that there was no code allowing build dependencies to be displayed in a remote lookup. To fix this, I first parsed the build and runtime dependencies field from the remote_data JSON object using the jq tool.
 - This is seen here:
 - `build_dependencies=$(echo "${remote_data}" | jq -r '.assets[0].build_dependencies | join(", ")')`

- `runtime_dependencies=$(echo "${remote_data}" | jq -r '.assets[0].runtime_dependencies')`
- I then created a Dependencies section that formatted and displayed both the build and runtime dependencies. I also added a condition that checked if either of these fields was missing or empty which would then display "None".
 - Here is how it looks:
 - `printHeader "==> Dependencies"`
 - `if [-n "${build_dependencies}"]; then`
 - `printf "%-20s %s\n" "Build:" "${build_dependencies}"`
 - `else`
 - `printf "%-20s %s\n" "Build:" "None"`
 - `fi`
 - `if [-n "${runtime_dependencies}"]; then`
 - `printf "%-20s %s\n" "Runtime:" "${runtime_dependencies}"`
 - `else`
 - `printf "%-20s %s\n" "Runtime:" "None"`
 - `fi`
- This should hopefully fix the issue.