# How DreamWorks uses Rez

# Statistics

These are common ranges. Not the extremes.

- New Rez packages per week: 80-300
- Rate of Rez environment requests: 5-20 per second
- Number of packages specified in a request:
  - 6 resolves to 92
  - 34 resolves to 135  (Our job environment likes to specify a lot)
  - 38 resolves to 168
  - 4 resolves to 22 (package build)
- Number of Rez packages: 29,882
- Number of variants: Up to ~40
  - debug/release, EL7/EL9

# What You See in OpenMoonRay

variants = [

['**os-CentOS-7**', 'opt_level-optdebug', 'refplat-vfx2020.3', 'icc-19.0.5.281.x.2', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', '**opt_level-debug**', 'refplat-vfx2020.3', 'icc-19.0.5.281.x.2', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', '**opt_level-optdebug**', 'refplat-vfx2020.3', 'gcc-6.3.x.2', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', 'opt_level-debug', '**refplat-vfx2020.3**', 'gcc-6.3.x.2', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', 'opt_level-optdebug', '**refplat-vfx2021.0**', 'gcc-9.3.x.1', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', 'opt_level-debug', 'refplat-vfx2021.0', 'gcc-9.3.x.1', 'usd_core-0.21.8.x.2'],

['os-CentOS-7', 'opt_level-optdebug', 'refplat-vfx2021.0', 'clang-13', 'gcc-9.3.x.1', 'usd_core-0.21.8.x.2'],

…

# What Goals Influence DreamWorks' Use of Rez

We use Rez to provide a healthy balance between consistency and flexibility.

Teams are free to manage their own packages, so the "healthy balance" varies.

Given that basis, we tend to prioritize:

- Production stability
- Simplifying user interactions
- Provide guardrails for Developers

# Production Stability

- Package: os
  - Tied to a major OS version by overriding the default in a rez config file

```
name = 'os'

version = 'rocky-9.x.0'

requires = [
    'platform-linux',
    'arch-x86_64',
    'dwa_compat_el9'
]
```

# Simplify User Interactions

- The industry software landscape is complex
- Users don’t enjoy knowing the details, so we create some abstractions
  - refplat - Weak package requirements, generally tied to VFX Reference Platform
  - rdcore - Set of DWA packages commonly used across all workflows
  - fxcore - Set of DWA packages commonly used by fx workflows

# Simplify User Interactions

Package: “refplat”

- Avoid mixing up versions of Boost, Python, Qt, etc…
- For example:
  - refplat-vfx2022.0.3.0 (this example)
  - refplat-vfx2021.0.0.1
  - refplat-maya2018.3.1.0

```
requires = [
    'cpp_std-17.0',
    '~alembic-1.8',
    '~boost-1.76',
    '~openexr-3.1',
    '~imath-3.1',
    '~python-3.7|3.9',
    '~qt-5.15',
    '~PySide2-5.15',
    '~PyQt5-5.15',
    '~tbb-2020.3',
    '~mkl-2020',
    '~ptex-2.4.1',
]
```

# Simplify User Interactions

Package: “rdcore”

- Implementation
  - Weak requests in the rdcore packages
  - Loose version specifications in the client packages
- Benefits
  - Avoid mixing up versions of DWA packages
  - Reduces chance of Rez conflicts
  - Reduces need for updates across many packages
- Cons
  - Clients need to wait for the rdcore package to be updated

rdcore-30.18.2.0

```
requires = [
    '~af-10.12.0',
    '~alembic_utilities-10.13.0',
    '~arguments-10.12.0',
    '~display-10.16.0',
```

Client Package

```
requires = [
    'rdcore-30.18',
    'foundation-10',
    'gtypes-10',
    'logs-10',
```

# Provide Developer Guardrails

- Some combinations can be incompatible, like across C++ standards
  - cpp_std
    - Make it hard to get a bad mix of configurations. e.g. This is where we set _GLIBCXX_USE_CXX11_ABI
    - Client package.py's switch on the cpp_std package to determine a compiler and add it to the requirements during builds
    - Allows individual packages to select their preferred compiler, within a range.

```
compilerSelection = {
    'cpp_std-11': ['gcc-4.8.x.2'],
    'cpp_std-14': ['gcc-6.3.x.2'],
    'cpp_std-17': ['gcc-11.x.1']
}
```

# Provide Developer Guardrails

- Package: opt_level
  - Optimized and debug
  - Most packages are built with both variants

# Simple Environment

```
rez env gcc-9
    cpp_std-17 log4cplus
```