

# .NET Core RID Catalog

📅 09/07/2017 ⌚ 3 minutes to read Contributors      [all](#)

## In this article

[RID graph](#)

[Using RIDs](#)

[Windows RIDs](#)

[Linux RIDs](#)

[macOS RIDs](#)

[Android RIDs \(.NET Core 2.0 or later versions\)](#)

[See also](#)

RID is short for *Runtime Identifier*. RID values are used to identify target platforms where the application runs. They're used by .NET packages to represent platform-specific assets in NuGet packages. The following values are examples of RIDs: `linux-x64`, `ubuntu.14.04-x64`, `win7-x64`, or `osx.10.12-x64`. For the packages with native dependencies, the RID designates on which platforms the package can be restored.

A single RID can be set in the `<RuntimeIdentifier>` element of your project file. Multiple RIDs can be defined as a semicolon-delimited list in the project file's `<RuntimeIdentifiers>` element. They're also used via the `--runtime` option with the following [.NET Core CLI commands](#):

- [dotnet build](#)
- [dotnet clean](#)
- [dotnet pack](#)
- [dotnet publish](#)
- [dotnet restore](#)
- [dotnet run](#)
- [dotnet store](#)

RIDs that represent concrete operating systems usually follow this pattern:

`[os].[version]-[architecture]-[additional qualifiers]` where:

- `[os]` is the operating/platform system moniker. For example, `ubuntu`.
- `[version]` is the operating system version in the form of a dot-separated ( `.` ) version number. For example, `15.10`.
  - The version **shouldn't** be marketing versions, as they often represent multiple discrete versions of the operating system with varying platform API surface area.
- `[architecture]` is the processor architecture. For example: `x86`, `x64`, `arm`, or `arm64`.


- [additional qualifiers] further differentiate different platforms. For example: aot or coreclr.

## RID graph

The RID graph or runtime fallback graph is a list of RIDs that are compatible with each other. The RIDs are defined in the [Microsoft.NETCore.Platforms](#) package. You can see the list of supported RIDs and the RID graph in the [runtime.json](#) file, which is located at the CoreFX repo. In this file, you can see that all RIDs, except for the base one, contain an `"#import"` statement. These statements indicate compatible RIDs.

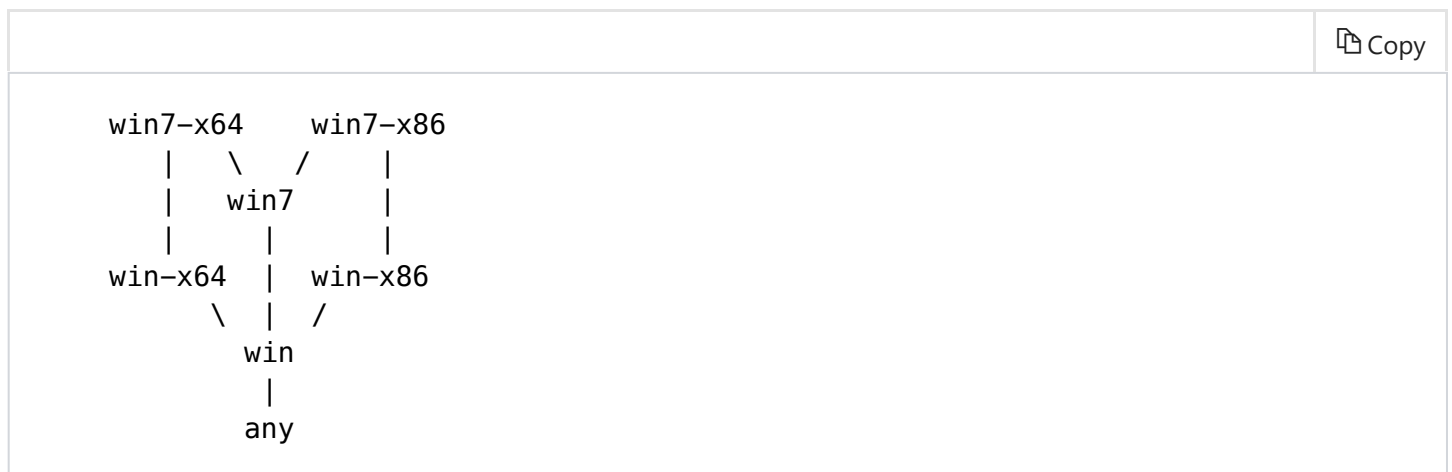
When NuGet restores packages, it tries to find an exact match for the specified runtime. If an exact match is not found, NuGet walks back the graph until it finds the closest compatible system according to the RID graph.

The following example is the actual entry for the `osx.10.12-x64` RID:

JSON	
<pre>"osx.10.12-x64": {   "#import": [ "osx.10.12", "osx.10.11-x64" ] }</pre>	

The above RID specifies that `osx.10.12-x64` imports `osx.10.11-x64`. So, when NuGet restores packages, it tries to find an exact match for `osx.10.12-x64` in the package. If NuGet cannot find the specific runtime, it can restore packages that specify `osx.10.11-x64` runtimes, for example.

The following example shows a slightly bigger RID graph also defined in the `runtime.json` file:



All RIDs eventually map back to the root `any` RID.

There are some considerations about RIDs that you have to keep in mind when working with them:

- RIDs are **opaque strings** and should be treated as black boxes.
- Don't build RIDs programmatically.
- Use RIDs that are already defined for the platform.
- The RIDs need to be specific, so don't assume anything from the actual RID value.

## Using RIDs

To be able to use RIDs, you have to know which RIDs exist. New values are added regularly to the platform. For the latest and complete version, see the [runtime.json](#) file on CoreFX repo.

.NET Core 2.0 SDK introduces the concept of portable RIDs. They are new values added to the RID graph that aren't tied to a specific version or OS distribution. They're particularly useful when dealing with multiple Linux distros.

The following list shows the most common RIDs used for each OS. It doesn't cover `arm` or `coreclr` values.

## Windows RIDs

- Portable
  - `win-x86`
  - `win-x64`
- Windows 7 / Windows Server 2008 R2
  - `win7-x64`
  - `win7-x86`
- Windows 8 / Windows Server 2012
  - `win8-x64`
  - `win8-x86`
  - `win8-arm`
- Windows 8.1 / Windows Server 2012 R2
  - `win81-x64`
  - `win81-x86`
  - `win81-arm`
- Windows 10 / Windows Server 2016
  - `win10-x64`
  - `win10-x86`
  - `win10-arm`
  - `win10-arm64`

See [Prerequisites for .NET Core on Windows](#) for more information.

# Linux RIDs

- Portable
  - `linux-x64`
- CentOS
  - `centos-x64`
  - `centos.7-x64`
- Debian
  - `debian-x64`
  - `debian.8-x64`
- Fedora
  - `fedora-x64`
  - `fedora.24-x64`
  - `fedora.25-x64` (.NET Core 2.0 or later versions)
  - `fedora.26-x64` (.NET Core 2.0 or later versions)
- Gentoo (.NET Core 2.0 or later versions)
  - `gentoo-x64`
- openSUSE
  - `opensuse-x64`
  - `opensuse.42.1-x64`
- Oracle Linux
  - `ol-x64`
  - `ol.7-x64`
  - `ol.7.0-x64`
  - `ol.7.1-x64`
  - `ol.7.2-x64`
- Red Hat Enterprise Linux
  - `rhel-x64`
  - `rhel.6-x64` (.NET Core 2.0 or later versions)
  - `rhel.7-x64`
  - `rhel.7.1-x64`
  - `rhel.7.2-x64`
  - `rhel.7.3-x64` (.NET Core 2.0 or later versions)
  - `rhel.7.4-x64` (.NET Core 2.0 or later versions)
- Tizen (.NET Core 2.0 or later versions)
  - `tizen`
- Ubuntu
  - `ubuntu-x64`

- `ubuntu.14.04-x64`
- `ubuntu.14.10-x64`
- `ubuntu.15.04-x64`
- `ubuntu.15.10-x64`
- `ubuntu.16.04-x64`
- `ubuntu.16.10-x64`
- Ubuntu derivatives
  - `linuxmint.17-x64`
  - `linuxmint.17.1-x64`
  - `linuxmint.17.2-x64`
  - `linuxmint.17.3-x64`
  - `linuxmint.18-x64`
  - `linuxmint.18.1-x64` (.NET Core 2.0 or later versions)

See [Prerequisites for .NET Core on Linux](#) for more information.

## macOS RIDs

macOS RIDs use the older "OSX" branding.

- `osx-x64` (.NET Core 2.0 or later versions, minimum version is `osx.10.12-x64` )
- `osx.10.10-x64`
- `osx.10.11-x64`
- `osx.10.12-x64` (.NET Core 1.1 or later versions)
- `osx.10.13-x64`

See [Prerequisites for .NET Core on macOS](#) for more information.

## Android RIDs (.NET Core 2.0 or later versions)

- `android`
- `android.21`

## See also

[Runtime IDs](#)