Yocto Providers, Preferences, and Compatible in Kernel Recipes

Introduction

In Yocto, `PROVIDERS`, `PREFERRED_PROVIDER`, `PREFERRED_VERSION`, and `COMPATIBLE_MACHINE` are key mechanisms used to control **which package versions** and **which kernel** (or other recipes) should be selected during a build. These mechanisms are especially important in **kernel recipes**, where multiple kernel sources or configurations might be available for different hardware architectures.

---

1. Understanding Yocto Providers

What is PROVIDERS?

- `PROVIDERS` defines which recipe **provides** a specific package or virtual target.
- It allows multiple recipes to provide the same functionality, and the build system can choose one.

Syntax

```
PROVIDERS += "virtual/kernel"
```

This means that a recipe **provides** the functionality of `virtual/kernel`, which is required for kernel builds.

Example

If there are multiple kernel recipes in a Yocto project:

```
meta/recipes-kernel/linux/linux-yocto_6.6.bb
meta/recipes-kernel/linux/linux-yocto_5.15.bb
meta-custom/recipes-kernel/linux/linux-custom.bb
```

Each of these may provide `virtual/kernel`:

```
PROVIDES += "virtual/kernel"
```

The build system then decides which one to use based on `PREFERRED_PROVIDER` and `PREFERRED_VERSION`.

---

2. Using PREFERRED_PROVIDER

What is PREFERRED_PROVIDER?

- It allows selecting a **specific provider** when multiple recipes provide the same package.
- It is set in `local.conf` or a **machine configuration file**.

Syntax

```
PREFERRED_PROVIDER_virtual/kernel = "linux-yocto"
```

This tells Yocto to use `linux-yocto.bb` as the provider for `virtual/kernel`.

Example

If there are multiple kernel providers:

```
linux-yocto_6.6.bb  # Provides virtual/kernel
linux-custom.bb      # Provides virtual/kernel
```

And we want to use `linux-custom`, we set:

```
PREFERRED_PROVIDER_virtual/kernel = "linux-custom"
```

This ensures the build uses **linux-custom.bb** instead of the default `linux-yocto`.

---

3. Controlling Versions with PREFERRED_VERSION

What is PREFERRED_VERSION?

- It allows selecting a **specific version** of a package.
- Useful when multiple versions of a kernel are available.

Syntax

```
PREFERRED_VERSION_linux-yocto = "6.6"
```

This tells BitBake to use the `linux-yocto_6.6.bb` recipe.

Example

If there are multiple kernel versions:

```
linux-yocto_6.6.bb
linux-yocto_5.15.bb
```

To use version **6.6**, set in `local.conf`:

```
PREFERRED_VERSION_linux-yocto = "6.6"
```

Or in `machine.conf`:

```
PREFERRED_VERSION_virtual/kernel = "6.6"
```

This ensures that **linux-yocto_6.6** is selected during the build.

4. Filtering Kernel for a Specific Machine with COMPATIBLE_MACHINE

What is COMPATIBLE_MACHINE?

- Restricts a recipe to **only be used for specific machines**.
- Prevents a kernel recipe from being selected on unsupported hardware.

Syntax

```
COMPATIBLE_MACHINE = "qemuarm64"
```

This means the recipe is only **valid for qemuarm64** machines.

Example

If there are kernel recipes for different architectures:

```
linux-yocto_6.6.bb
linux-yocto-arm.bb
```

And `linux-yocto-arm.bb` should only be used for ARM machines:

```
COMPATIBLE_MACHINE = "qemuarm"
```

Trying to use this kernel on x86 will result in an error:

```
ERROR: No recipes available for virtual/kernel
```

This ensures that the correct kernel is chosen **based on hardware**.

5. Practical Use in Kernel Recipes

Example Kernel Recipe (`linux-custom.bb`)

```
DESCRIPTION = "Custom Linux Kernel"
PROVIDES += "virtual/kernel"
PREFERRED_VERSION_linux-custom = "6.6"

SRC_URI =
"git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git;branch=v6.6
"
COMPATIBLE_MACHINE = "qemuarm64"

do_compile() {
    echo "Building Kernel..."
}
```

Machine Configuration (`qemuarm64.conf`)

```
PREFERRED_PROVIDER_virtual/kernel = "linux-custom"
PREFERRED_VERSION_virtual/kernel = "6.6"
```

Expected Outcome

1. The build system selects **linux-custom.bb** for `virtual/kernel`.
2. The build system chooses **version 6.6**.
3. The kernel is **only used for `qemuarm64`**, preventing compatibility issues.

---

6. Summary

Feature	Purpose	Example
`PROVIDERS`	Defines what a recipe provides	`PROVIDERS += "virtual/kernel"`
`PREFERRED_PROVIDER`	Selects a specific provider	`PREFERRED_PROVIDER_virtual/kernel = "linux-yocto"`
`PREFERRED_VERSION`	Selects a specific version	`PREFERRED_VERSION_linux-yocto = "6.6"`
`COMPATIBLE_MACHINE`	Restricts a recipe to certain machines	`COMPATIBLE_MACHINE = "qemuarm64"`

---

7. Conclusion

- `PROVIDERS` allows multiple recipes to provide the same functionality.
- `PREFERRED_PROVIDER` selects the preferred recipe when multiple providers exist.
- `PREFERRED_VERSION` ensures the correct version of a package is used.
- `COMPATIBLE_MACHINE` restricts a recipe to specific hardware.

By combining these features, Yocto provides a **flexible** and **controlled** way to manage kernel selection for different architectures and machine types.