



# **Yocto Project® Demystifying the OVERRIDES mechanism and Bitbake operators**

Quentin Schulz, StreamUnlimited  
Slides under CC BY-SA-4.0

**Yocto Project Summit, May 2021**



# About

# About



- **BSP engineer @ StreamUnlimited GmbH**
- **Smart speakers**
- **SoM for anything related to audio or voice assistance**
- **StreamSDK SW stack:**
  - Google Cast, Amazon Alexa, Spotify, Airplay, multiroom audio, Dolby Atmos, Multi-UI...**
- **Open positions:**
  - <https://www.streamunlimited.com/vacancies/>**
- **quentin.schulz [at] streamunlimited.com**
- **foss [at] 0leil.net (zero not o)**



# No-surprise operators

# No-surprise operators

- =, +=, =+

```
GUESTS = "jasmine"  
GUESTS += "christian"
```

```
GUESTS = "aaron"  
GUESTS =+ "marie"
```

```
GUESTS += "muhammad"  
GUESTS = "ilana"
```

# No-surprise operators

- =, +=, =+

```
GUESTS = "jasmine"  
GUESTS += "christian"  
# Result: GUESTS => "jasmine christian"
```

```
GUESTS = "aaron"  
GUESTS =+ "marie"  
# Result: GUESTS => "marie aaron"
```

```
GUESTS += "muhammad"  
GUESTS = "ilana"  
# Result: GUESTS => "ilana"
```

# No-surprise operators

- `.=`, `=.`

```
MARK = "B"
```

```
MARK .= "+"
```

```
MONEY = "524,00$"
```

```
MONEY =. "1000000"
```

```
MARK .= "+"
```

```
MARK = "B"
```

# No-surprise operators

- `.=`, `=.`

```
MARK = "B"
```

```
MARK .= "+"
```

```
# Result: MARK => "B+"
```

```
MONEY = "524,00$"
```

```
MONEY =. "1000000"
```

```
# Result: MONEY => "1000000524,00$"
```

```
MARK .= "+"
```

```
MARK = "B"
```

```
# Result: MARK => "B"
```





# Weak assignment operators

# Weak assignment operators

- `?=`, `??=`

```
BUFFET ?= "tacos croissant baklava sushi"  
BUFFET += "curry"
```

```
BUFFET ??= "lokum gyoza hummus"  
BUFFET ?= "pizza"  
BUFFET += "paella"
```

```
BUFFET += "ice-cream"  
BUFFET ?= "schnitzel blini"
```

# Weak assignment operators

- `?=`, `??=`

```
BUFFET ?= "tacos croissant baklava sushi"
```

```
BUFFET += "curry"
```

```
# Result: BUFFET => "tacos croissant baklava sushi curry"
```

```
BUFFET ??= "lokum gyoza hummus"
```

```
BUFFET ?= "pizza"
```

```
BUFFET += "paella"
```

```
# Result: BUFFET => "pizza paella"
```

```
BUFFET += "ice-cream"
```

```
BUFFET ?= "schnitzel blini"
```

```
# Result: BUFFET => "ice-cream"
```



# Non-obvious order

# Non-obvious order

```
# cinnamon-buns recipe  
INGREDIENTS ?= "cinnamon flour yeast butter eggs milk  
sugar"
```

```
# local.conf  
INGREDIENTS += "glaze vanilla-extract"
```

# Non-obvious order

```
# cinnamon-buns recipe  
INGREDIENTS ?= "cinnamon flour yeast butter eggs milk  
sugar"
```

```
# local.conf  
INGREDIENTS += "glaze vanilla-extract"
```

```
# Result: INGREDIENTS => "glaze vanilla-extract"
```



# Debugging technique

# Debugging technique

```
qschulz> bitbake -e cinnamon-buns | \
    awk '/^# \$INGREDIENTS \[/,/^INGREDIENTS/'
# $INGREDIENTS [2 operations]
#   append /yocto/build/conf/local.conf:268
#   "glaze vanilla-extract"
#   set? /yocto/meta-yp-summit/recipes-example/cinnamon-
buns/cinnamon-buns.bb:39
#   "cinnamon flour yeast butter eggs milk sugar"
# pre-expansion value:
#   " glaze vanilla-extract"
INGREDIENTS=" glaze vanilla-extract"
```

Source: [https://theyoctojester.info/session\\_14/main.html](https://theyoctojester.info/session_14/main.html) Josef Holzmayr a.k.a. LetoThe2nd a.k.a. TheYoctoJester



# Debugging technique n°1

- **Configuration files (.conf) parsed before recipes, bbappends or include files**
- **New operator needed!**



# `_append` and `_prepend`

# **\_append and \_prepend**

- **Equivalent of a "postponed" `.=` and `=.`**
- **~ post-its/pile/procrastinator**
- **Tell Bitbake "resolve `??=` `?=` `= +=` `=+` `.=` `=.` then and only then `_append/_prepend`"**

```
# cinnamon-buns recipe  
INGREDIENTS ?= "cinnamon flour yeast butter eggs milk  
sugar"
```

```
# local.conf  
INGREDIENTS += "glaze vanilla-extract"
```

# **`_append` and `_prepend`**

```
# cinnamon-buns recipe  
INGREDIENTS ?= "cinnamon flour yeast butter eggs milk  
sugar"
```

```
# local.conf  
INGREDIENTS_append = " glaze vanilla-extract"
```

```
# Result: INGREDIENTS => "cinnamon flour yeast butter eggs  
milk sugar glaze vanilla-extract"
```

## append and \_prepend

```
qschulz> bitbake -e cinnamon-buns | \
    awk '/^# \$INGREDIENTS \[/,/^INGREDIENTS/'
# $INGREDIENTS2 [2 operations]
# _append /yocto/build/conf/local.conf:269
# " glaze vanilla-extract"
# set? /yocto/meta-yp-summit/recipes-example/cinnamon-
buns/cinnamon-buns.bb:42
# "cinnamon flour yeast butter eggs milk sugar"
# pre-expansion value:
# "cinnamon flour yeast butter eggs milk sugar glaze
vanilla-extract"
INGREDIENTS="cinnamon flour yeast butter eggs milk sugar
glaze vanilla-extract"
```



**Surprise, surprise bis**

# Surprise, surprise bis

- `_append += => _append + leading space`
- **One `_append` (or `_prepend`) = one post-it/entry in the pile**

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
# OOPSIES, my nephew wants a doll and not a puzzle  
GIFTS_append = " doll"
```

# Surprise, surprise bis

- `_append += => _append + leading space`
- **One `_append` (or `_prepend`) = one post-it/entry in the pile**

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
# OOPSIES, my nephew wants a doll and not a puzzle  
GIFTS_append = " doll"  
# Result: GIFTS => "train puzzle doll"
```

- **New operator!**





# Last word

## Last word

- **`_remove`, another "postponed" operator**
- **Tell Bitbake "resolve ??= ?= = += =+ .= =.  
then `_append/_prepend` then `_remove`"**
- **Final and removes all occurrences!**

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
GIFTS_append = " doll"  
GIFTS_remove = "puzzle"
```

## Last word

- `_remove`, another "postponed" operator
- Tell Bitbake "resolve ??= ?= = += =+ .= =.  
then `_append/_prepend` then `_remove`"
- Final and removes all occurrences!

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
GIFTS_append = " doll"  
GIFTS_remove = "puzzle"  
# Result: GIFTS => "train doll"
```

## Last word

- **\_remove**, another "postponed" operator
- **Tell Bitbake "resolve ??= ?= = += =+ .= =. then \_append/\_prepend then \_remove"**
- **Final and removes all occurrences!**

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
GIFTS_append = " doll"  
GIFTS_remove = "puzzle"  
# OOPSIES, my nephew actually wants a puzzle too  
GIFTS_append = " puzzle"
```

## Last word

- `_remove`, another "postponed" operator
- Tell Bitbake "resolve ??= ?= = += =+ .= =.  
then `_append/_prepend` then `_remove`"
- Final and removes all occurrences!

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
GIFTS_append = " doll"  
GIFTS_remove = "puzzle"  
# OOPSIES, my nephew actually wants a puzzle too  
GIFTS_append = " puzzle"  
# Result: GIFTS => "train doll"
```

## Last word

- `_remove`, to be avoided (3<sup>rd</sup> party layer)
- `_append/_prepend` to be carefully chosen
- Trick to undo `_remove`

```
GIFTS = "train"  
GIFTS_append = " puzzle"  
GIFTS_append = " doll"  
UNWANTED_GIFTS = "puzzle"  
GIFTS_remove = "${UNWANTED_GIFTS}"  
UNWANTED_GIFTS = ""  
# Result: GIFTS => "train puzzle doll"
```

Source: [https://theyoctojester.info/session\\_14/main.html](https://theyoctojester.info/session_14/main.html) Chris Larson



**Now!**

# Now!

- **:= immediate expansion operator**

```
FILESEXTRAPATHS_prepend := "${THISDIR}/files:"
```

- **FILESEXTRAPATHS\_prepend**
- **THISDIR is directory at expansion time**
- **Variables expanded after all parsing**
- **Called "flattening"**
- **If =, THISDIR is directory of recipe**
- **If :=, THISDIR is dir of currently parsed file**





# Usage

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS += "cmake-native"  
# cmake-based-sw_0.bb  
inherit cmake  
DEPENDS = "bash"  
# cmake-based-sw_1.bb  
DEPENDS = "bash"  
inherit cmake
```

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS += "cmake-native"  
# cmake-based-sw_0.bb  
inherit cmake  
DEPENDS = "bash"  
# Result: DEPENDS => "bash"  
# cmake-based-sw_1.bb  
DEPENDS = "bash"  
inherit cmake  
# Result: DEPENDS => "bash cmake-native"
```

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS += "cmake-native"  
# boost.bbclass  
DEPENDS = "boost"  
# cmake-based-sw_2.bb  
inherit cmake boost  
# cmake-based-sw_3.bb  
inherit boost cmake
```

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS += "cmake-native"  
# boost.bbclass  
DEPENDS = "boost"  
# cmake-based-sw_2.bb  
inherit cmake boost  
# Result: DEPENDS => "boost"  
# cmake-based-sw_3.bb  
inherit boost cmake  
# Result: DEPENDS => "boost cmake-native"
```

# Usage

- **`_append/_prepend`** in **`.bbclass`**, **`.inc`** and **`.conf`**

```
# cmake.bbclass  
DEPENDS_append = " cmake-native"  
# cmake-based-sw_0.bb  
inherit cmake  
DEPENDS = "bash"  
# cmake-based-sw_1.bb  
DEPENDS = "bash"  
inherit cmake
```

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS_append = " cmake-native"
```

```
# cmake-based-sw_0.bb
```

```
inherit cmake
```

```
DEPENDS = "bash"
```

```
# cmake-based-sw_1.bb
```

```
DEPENDS = "bash"
```

```
inherit cmake
```

```
# Result for both: DEPENDS => "bash cmake-native"
```

# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS_append = " cmake-native"  
# boost.bbclass  
DEPENDS_append = " boost"  
# cmake-based-sw_2.bb  
inherit cmake boost  
# cmake-based-sw_3.bb  
inherit boost cmake
```



# Usage

- **`_append/_prepend` in `.bbclass`, `.inc` and `.conf`**

```
# cmake.bbclass  
DEPENDS_append = " cmake-native"  
# boost.bbclass  
DEPENDS_append = " boost"  
# cmake-based-sw_2.bb  
inherit cmake boost  
# Result: DEPENDS => " cmake-native boost"  
# cmake-based-sw_3.bb  
inherit boost cmake  
# Result: DEPENDS => " boost cmake-native"
```

# Usage

- **\_append/\_prepend** in **.bbclass**, **.inc** and **.conf**
- **For "context-specific" additions**
- **Should be avoided (hard to override)**
- **Best practice in recipes:**
  - 1) inherit some-class**
  - 2) Set/modify/append variables**



# **"Context-specific" operators**

# "Context-specific" operators

```
# For Omar
```

```
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"
```

# "Context-specific" operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For Elena  
TODAY_TODO = "breakfast school nap play dinner sleep"
```

# "Context-specific" operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For Elena  
TODAY_TODO_elena = "breakfast school nap play dinner sleep"
```

- **\_elena operator = new pile for TODAY\_TODO**
- **Overrides TODAY\_TODO only for Elena after all ??= ?= = += =+ .= =., before \_append**
- **TODAY\_TODO\_elena resolved (but unused!) even for Omar**

# "Context-specific" operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For Elena  
TODAY_TODO_elena = "breakfast school nap play dinner sleep"  
# For Esteban  
TODAY_TODO_esteban = "breakfast school nap play dinner  
sleep"
```

- **`_elena` and `_esteban` = MACHINE in Yocto**



# Avoiding copy-pasting



# Avoiding copy-pasting

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child = "breakfast school nap play dinner sleep"
```

- **\_child = MACHINEOVERRIDES in Yocto**
- **Elena and Esteban are children, TODAY\_TODO overridden for them**

# Avoiding copy-pasting

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child = "breakfast school nap play dinner sleep"  
# For Elena  
TODAY_TODO_elena = "breakfast school nap play dinner sleep  
hindi-language"
```

- **Individual > "kind"**

**MACHINE > MACHINEOVERRIDES in Yocto**

# Avoiding copy-pasting

```
MACHINEOVERRIDES = "kind1:kind2:kind3:kind4:${MACHINE}"  
MACHINEOVERRIDES = "child:elena"  
MACHINEOVERRIDES = "child:esteban"  
MACHINEOVERRIDES = "omar"
```

- **Rightmost > leftmost**
- **\_elena > \_child**
- **\_esteban > \_child**



# Combining operators

# Combining operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child = "breakfast school nap play dinner sleep"  
TODAY_TODO += "play-with-dog"
```

# Combining operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child = "breakfast school nap play dinner sleep"  
TODAY_TODO += "play-with-dog"  
# Result for children: TODAY_TODO => "breakfast school nap  
play dinner sleep"  
# Result for Omar: TODAY_TODO => "make-breakfast breakfast  
commute work sport make-dinner dinner sleep play-with-dog"
```

# Combining operators

```
# For Omar  
TODAY_TODO = "make-breakfast breakfast commute work sport  
make-dinner dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child = "breakfast school nap play dinner sleep"  
TODAY_TODO_append = " play-with-dog"
```

- **\_append after "normal" resolution**
- **For children, base TODAY\_TODO = TODAY\_TODO\_child, then \_append**

# Combining operators

```
# For all  
TODAY_TODO = "breakfast dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child_append = " school nap play"  
# For Omar  
TODAY_TODO_omar_append = " make-breakfast make-dinner  
commute work sport"
```



# Combining operators

```
# For all  
TODAY_TODO = "breakfast dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_child_append = " school nap play"  
# For Omar  
TODAY_TODO_omar_append = " make-breakfast make-dinner  
commute work sport"  
# Result for children: TODAY_TODO => " school nap play"  
# Result for Omar: TODAY_TODO => " make-breakfast make-  
dinner commute work sport"  
# Result for others: TODAY_TODO => "breakfast dinner sleep"
```

# Combining operators

```
# For all  
TODAY_TODO = "breakfast dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_append_child = " school nap play"  
# For Omar  
TODAY_TODO_append_omar = " make-breakfast make-dinner  
commute work sport"
```

# Combining operators

```
# For all  
TODAY_TODO = "breakfast dinner sleep"  
# For children (Elena, Esteban, ...)  
TODAY_TODO_append_child = " school nap play"  
# For Omar  
TODAY_TODO_append_omar = " make-breakfast make-dinner  
commute work sport"  
# Result for children: TODAY_TODO => "breakfast dinner  
sleep school nap play"  
# Result for Omar: TODAY_TODO => "breakfast dinner sleep  
make-breakfast make-dinner commute work sport"  
# Result for others: TODAY_TODO => "breakfast dinner sleep"
```



# Reading combined operators

# Reading combined operators

*TODAY\_TODO\_omar\_append*

- **\_omar = pile for Omar**
- **\_append = pile for later (post-it)**
- **\_omar\_append creates a post-it for Omar-reserved pile**
- **Omar: TODAY\_TODO = TODAY\_TODO\_omar, resolved (+ = .= ...), then \_append**

# Reading combined operators

*TODAY\_TODO\_append\_omar*

- **\_append = pile for later (post-it)**
- **\_omar = pile for Omar**
- **\_append\_omar creates for Omar only a post-it on top of default pile**
- **TODAY\_TODO will be resolved, then \_append\_omar will be added if Omar**

# Reading combined operators

```
TODAY_TODO = "breakfast dinner sleep"
TODAY_TODO_append = " play-with-dog"
# For children (Elena, Esteban, ...)
TODAY_TODO_append_child = " school nap play"
# For Omar
TODAY_TODO_append_omar = " make-breakfast make-dinner
commute work sport"
# Result for children: TODAY_TODO => "breakfast dinner
sleep play-with-dog school nap play"
# Result for Omar: TODAY_TODO => "breakfast dinner sleep
play-with-dog make-breakfast make-dinner commute work
sport"
# Result for others: TODAY_TODO => "breakfast dinner sleep
play-with-dog"
```

# Reading combined operators

- **When in doubt, bitbake -e!**





# **OVERRIDES for files**

# OVERRIDES for files

```
# my-kernel.bb  
SRC_URI += "file://defconfig"
```

```
meta-yp-summit  
├─ recipes-kernel  
  └─ linux  
    └─ my-kernel  
      └─ aarch64  
        └─ defconfig <--- aarch64 machines  
      └─ defconfig <--- all other machines  
    └─ my-kernel_5.10.bb
```



# **OVERRIDES for files - gotchas**

# OVERRIDES for files - gotchas

*meta-yp-summit*

├─ *recipes-append*

├─ *linux*

├─ *my-kernel*

├─ *defconfig* <--- want for *my-aarch64-machine*

├─ *my-kernel\_5.10.bbappend*

├─ *recipes-kernel*

├─ *linux*

├─ *my-kernel*

├─ *aarch64*

├─ *defconfig*

├─ *my-kernel\_5.10.bb*

# OVERRIDES for files - gotchas

*meta-yp-summit*

├─ *recipes-append*

├─ *linux*

├─ *my-kernel*

├─ *defconfig*

├─ *my-kernel\_5.10.bbappend*

├─ *recipes-kernel*

├─ *linux*

├─ *my-kernel*

├─ *aarch64*

├─ *defconfig* <--- *aarch64 machines*

├─ *my-kernel\_5.10.bb*

# OVERRIDES for files - gotchas

- **Fetch order from FILESPATH**
- **Check value with**
  - **bitbake my-kernel -e | awk '/^# \\${FILESPATH} \[/,/^FILESPATH/'**
  - **Content of `\${WORKDIR}/temp/log.do_fetch`**

# OVERRIDES for files - gotchas

*meta-yp-summit*

├─ *recipes-append*

├─ *linux*

├─ *my-kernel*

├─ *aarch64*

├─ *defconfig* <--- *aarch64 machines*

├─ *my-kernel\_5.10.bbappend*

├─ *recipes-kernel*

├─ *linux*

├─ *my-kernel*

├─ *aarch64*

├─ *defconfig*

├─ *my-kernel\_5.10.bb*

# OVERRIDES for files - gotchas

*meta-yp-summit*

├── *recipes-append*

├── *linux*

├── *my-kernel*

├── *my-aarch64-machine*

├── *defconfig* <--- *my-aarch64-machine*

├── *my-kernel\_5.10.bbappend*

├── *recipes-kernel*

├── *linux*

├── *my-kernel*

├── *aarch64*

├── *defconfig* <--- *aarch64 machines*

├── *my-kernel\_5.10.bb*



# OVERRIDES for files - gotchas

```
meta-yp-summit
├── recipes-append
│   └── linux
│       ├── my-kernel
│       │   └── src
│       │       └── fileA
│       └── my-kernel_5.10.bbappend
├── recipes-kernel
│   └── linux
│       ├── my-kernel
│       │   └── src
│       │       ├── fileA
│       │       └── fileB
│       └── my-kernel_5.10.bb
```

```
# my-kernel_5.10.bb
SRC_URI = "file://src/"

# my-kernel_5.10.bbappend
FILESEXTRAPATHS_prepend :=
"${THISDIR}/my-kernel:"
```

# OVERRIDES for files - gotchas

```
meta-yp-summit
├── recipes-append
│   └── linux
│       ├── my-kernel
│       │   └── src
│       │       └── fileA
│       └── my-kernel_5.10.bbappend
├── recipes-kernel
│   └── linux
│       ├── my-kernel
│       │   └── src
│       │       ├── fileA
│       │       └── fileB
│       └── my-kernel_5.10.bb
```

```
# my-kernel_5.10.bb
SRC_URI = "file://src/"

# my-kernel_5.10.bbappend
FILESEXTRAPATHS_prepend :=
"${THISDIR}/my-kernel:"

# Result: only fileA from
the bbappend will be
fetched
```



# Other OVERRIDES

# Other OVERRIDES

```
OVERRIDES = "${TARGET_OS}:${TRANSLATED_TARGET_ARCH}:pn-${PN}:  
${MACHINEOVERRIDES}:${DISTROOVERRIDES}:${CLASSOVERRIDE}  
${LIBCOVERRIDE}:forcevariable"
```

- **\_pn-<recipename>**
  - only makes sense from .conf files (c.f. Yocto chant #1)
- **\_class-native, class-target, ...**
- **\_virtclass-multilib-lib32 (MULTILIBS = "multilib:lib32")**
- **\_libc-musl (LIBCOVERRIDE tclibc-musl.inc)**
- **\_poky (DISTROOVERRIDES)**
- **\_forcevariable (takes precedence over everything)**

## Other OVERRIDES

- **\_2.0, \_3.2+ (BBVERSIONS; rarely used)**
- **\_task-compile (task-specific changes, e.g. PARALLEL\_MAKE; note missing do\_)**
- **Probably others :)**

# Sources

- <https://docs.yoctoproject.org/bitbake/bitbake-user-manual/bitbake-user-manual-metadata.html#syntax-and-operators>
- [https://theyoctojester.info/session\\_14/main.html](https://theyoctojester.info/session_14/main.html)
- Me with years of trial and error :)
  
- **OVERRIDES history:**
  - <https://lists.openembedded.org/g/openembedded-architecture/message/1204>
  - <https://lists.openembedded.org/g/openembedded-architecture/message/1208>
- **What's next for OVERRIDES? Not much :)**



**This is the end**

```
LAST_WORDS_conference_2021_yocto-summit_speaker_quentin = "Thank you  
for attending! Any questions?"
```



# Bonus slides





**Surprise, surprise**

# Surprise, surprise

- **\_append and \_prepend for shell functions and tasks!**

```
prepare_prepend() {  
    take shopping-list  
}  
prepare() {  
    take wallet  
}  
prepare_append() {  
    take tote-bag  
}
```

```
do_groceries() {  
    buy bread  
}  
do_groceries_append() {  
    buy butter  
}  
do_groceries_prepend() {  
    prepare  
}
```

# Surprise, surprise

- **\_append and \_prepend for Bitbake-style python functions and tasks!**

```
python do_walk_dog_prepend() {  
    leash_dog()  
}
```

```
python do_walk_dog() {  
    go_outside()  
    take_dog()  
}
```

```
python end_walk_dog() {  
    clean_paws()  
}
```

```
python end_walk_dog_append() {  
    feed_dog()  
}
```



# Importance of naming

# Importance of naming

- **TODAY\_TODO\_omar\_append**
- **Variables:**
  - All uppercase,
  - Underscore possible,
- **Overrides:**
  - Prefixed with an underscore,
- **String from variable or override?**
  - **Override:**
    - No underscore,
    - All lowercase,

# Importance of naming

- **Package managers don't like uppercase letters**
  - **Debian package naming convention:**  
<https://www.debian.org/doc/debian-policy/ch-controlfields.html#source>
- **Naming convention:**
  - **No underscore, or capital letters for:**
    - **Recipe and package names,**
    - **Machine names, (MACHINEOVERRIDES),**
    - **Distro names, (DISTROOVERRIDES)**



# Files **OVERRIDES** algorithm

# Files OVERRIDES algorithm

```
FILESETRAPATHS_prepend := "${THISDIR}/${BPN}:"
```

- **FILESOVERRIDES = "...:\${MACHINEOVERRIDES}..."**
- Iterate from rightmost to leftmost FILESOVERRIDES
- For each iteration, iterate from leftmost to rightmost SRCDIRS (illustrative variable)

```
SRCDIRS = "${FILESETRAPATHS}:/dir/${BP}:/dir/${BPN}:/dir/files"  
SRCDIRS =  
"/dir2/my-kernel:/dir/my-kernel-5.10:/dir/my-kernel:/dir/files"  
# /dir = /path/to/meta-yp-summit/recipes-kernel/linux  
# /dir2 = /path/to/meta-yp-summit/recipes-append/linux
```

- **FILESPATH** contains the list of paths to traverse



# Files OVERRIDES algorithm

```
# pseudo-code
file # file listed in SRC_URI
paths = ${FILESEXTRAPATHS} + /dir/${BP} + /dir/${BPN} + /dir/files

for path in $paths
  for override in reverse(${FILESOVERRIDES})
    if exists($path/$override/$file)
      return found

  if exists($path/$file)
    return found

error "file not found"
```

# Files OVERRIDES algorithm

*meta-yp-summit*

├── *recipes-append*

├── *linux*

├── *my-kernel*

├── *aarch64*

├── *defconfig* <--- ***aarch64 machines***

├── *my-kernel\_5.10.bbappend*

├── *recipes-kernel*

├── *linux*

├── *my-kernel*

├── *aarch64*

├── *defconfig*

├── *my-kernel\_5.10.bb*

# Files OVERRIDES algorithm

*meta-yp-summit*

├── *recipes-append*

├── *linux*

├── *my-kernel*

├── *my-aarch64-machine*

├── *defconfig* <--- *my-aarch64-machine*

├── *my-kernel\_5.10.bbappend*

├── *recipes-kernel*

├── *linux*

├── *my-kernel*

├── *aarch64*

├── *defconfig* <--- *aarch64 machines*

├── *my-kernel\_5.10.bb*



# Python-equivalent operators

# Python-equivalent operators

- **Meaningful within anonymous python functions => task scope!**

```
X = "A" => d.setVar("X", "A")  
${X} => d.getVar("X")  
Y := ${X} => d.getVar("X", True)  
X_append = "B" => d.appendVar("X", "B")  
X_prepend = "C" => d.prependVar("X", "C")
```

- **Source:**  
<https://docs.yoctoproject.org/bitbake/bitbake-user-manual/bitbake-user-manual-metadata.html#functions-for-accessing-datastore-variables>

# This is the (real) end

```
LAST_WORDS_conference_2021_yocto-summit_speaker_quentin = "Thank you  
for attending! Any questions?"
```



yocto  
PROJECT

THE  
LINUX  
FOUNDATION