



# USING DEVTOOL TO STREAMLINE YOUR YOCTO\* PROJECT WORKFLOW

Tim Orling  
Open Source Technology Center

ELC NA, Portland, OR  
February 21, 2017



# OUTLINE

- Why `devtool`?
- Overview of how `devtool` works
- Types of Projects Currently Supported
- Most Common `devtool` Commands
- `devtool` is Evolving and Improving

# WHY DEVTOOL?

## Workflow before:

Fire up your trusty editor

Figure out where the recipe is or should be

Copy/Paste errors.

What is the minimum for a valid recipe?

What was that `VARIABLE_NAME`?

How do I do a md5sum in my editor?

What should I inherit? require?  
DEPENDS? RDEPENDS?

Oops, I forgot to commit that.

Darn, I should have created a new layer

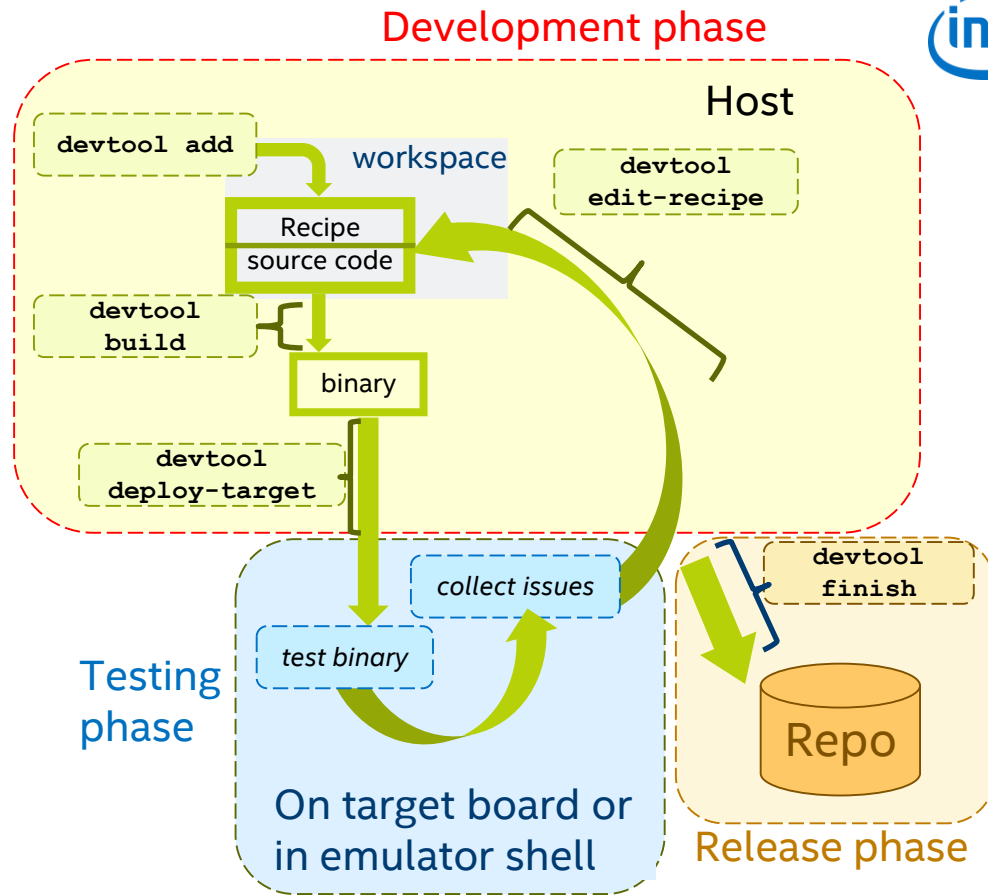
I just want to build it!

I just want to deploy it!

# WHY DEVTOOL?

## Workflow after:

- ✓ Fire up your trusty editor
- ✓ Figure out where the recipe is or should be
- ✓ Copy/Paste errors.
- ✓ What is the minimum for a valid recipe?
- ✓ What was that VARIABLE\_NAME?
- ✓ How do I do a md5sum in my editor?
- ✓ What should I inherit? require? DEPENDS? RDEPENDS?
- ✓ Oops, I forgot to commit that.
- ✓ Darn, I should have created a new layer
- ✓ I just want to build it!
- ✓ I just want to deploy it!





# TYPES OF PROJECTS CURRENTLY SUPPORTED

- Autotools (`autoconf` and `automake`)
- Cmake
- `qmake`
- Plain `Makefile`
- Out-of-tree kernel module
- Binary package (i.e. “-b” option)
- Node.js\* module
- Python modules that use `setuptools` or `distutils`

# MOST COMMON DEVTOOL COMMANDS

## `devtool add`

- Create a new recipe

## `devtool modify`

- Modify the source built by a recipe

## `devtool upgrade`

- Upgrade a recipe to a newer version



# DEVTOOL HAS GREAT SELF-DOCUMENTATION

```
$ devtool add --help
```

```
usage: devtool add [-h] [--same-dir | --no-same-dir] [--fetch URI]
[--version VERSION] [--no-git] [--autorev] [--binary][--also-native]
[--src-subdir SUBDIR] [recipeName] [srctree] [fetchuri]
```

Adds a new recipe to the workspace to build a specified source tree. Can optionally fetch a remote URI and unpack it to create the source tree.

arguments:

- recipeName** Name for new recipe to add (just name - no version, path or extension). If not specified, will attempt to auto-detect it.
- srctree** Path to external source tree. If not specified, a subdirectory of `/<workdir>/workspace/sources` will be used.
- fetchuri** Fetch the specified URI and extract it to create the source tree!







# WHY CREATE RECIPES FROM SCRATCH?

Do you have extra time to spare?

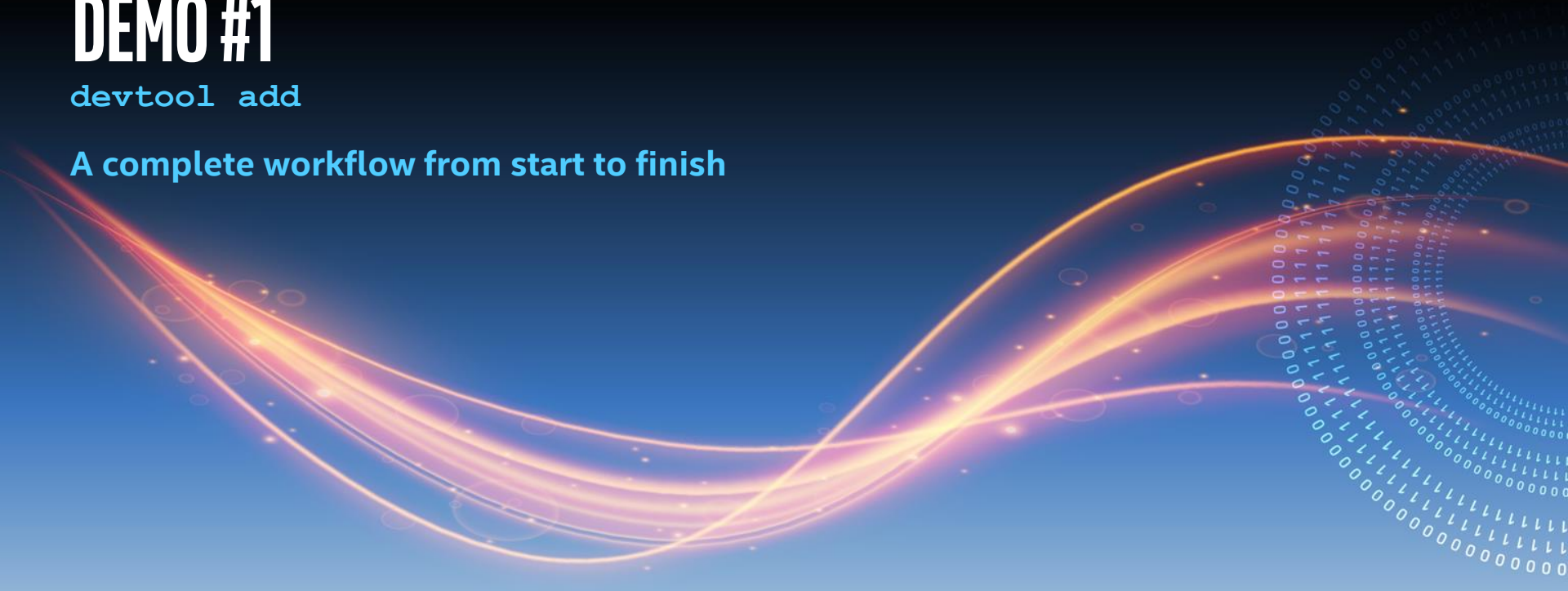




# DEMO #1

`devtool add`

A complete workflow from start to finish





# WHY MODIFY SOURCE CODE WITH QUILT?

Do you have extra time to spare?





# DEMO #2

`devtool modify`

A simple example





# WHY UPGRADE RECIPES BY HAND?

Do you have extra time to spare?





# DEMO #3

devtool upgrade

A real layer maintenance workflow example





# DEVTOOL IS EVOLVING AND IMPROVING

<b>Fido (1.8)</b> <ul style="list-style-type: none"><li>• Introduced</li></ul>	<b>Jethro (2.0)</b> <ul style="list-style-type: none"><li>• Improved</li></ul>	<b>Krogoth (2.1)</b> <ul style="list-style-type: none"><li>• True Timesaver</li></ul>	<b>Morty (2.2)</b> <ul style="list-style-type: none"><li>• Refined</li></ul>	<b>Pyro (2.3)</b> <ul style="list-style-type: none"><li>• Polished</li></ul>	<b>2.4+</b> <ul style="list-style-type: none"><li>• Possibilities!</li><li>• Your idea</li></ul>
--	--	---	--	--	--



# GRATUITOUS PLUG

Wednesday, February 22 • 11:40am - 12:30pm

## Cross-Platform Enablement for the Yocto\* Project with Containers

- Randy Witt, Intel OTC
- <http://sched.co/9ltu>

Wednesday, February 22 • 3:00pm - 3:50pm

## Yocto\* Project Extensible SDK: Simplifying the Workflow for Application Developers

- Henry Bruce, Intel OTC
- <http://sched.co/9ltz>



# CALL TO ACTION

- Use the tool whenever practical
- Contribute!
  - `devtool` is part of OE-Core
  - [openembedded-core@lists.openembedded.org](mailto:openembedded-core@lists.openembedded.org)
  - <http://lists.openembedded.org/mailman/listinfo/openembedded-core>



# READ THE DOCS

<http://www.yoctoproject.org/docs/current/sdk-manual/sdk-manual.html#using-devtool-in-your-sdk-workflow>



## 2.4. Using `devtool` in Your SDK Workflow

The cornerstone of the extensible SDK is a command-line tool called `devtool`. This tool provides a number of features that help you build, test and package software within the extensible SDK, and optionally integrate it into an image built by the OpenEmbedded build system.

The `devtool` command line is organized similarly to `Git` in that it has a number of sub-commands for each function. You can run `devtool --help` to see all the commands.

Three `devtool` subcommands that provide entry-points into development are:

- **`devtool add`**: Assists in adding new software to be built.
- **`devtool modify`**: Sets up an environment to enable you to modify the source of an existing component.
- **`devtool upgrade`**: Updates an existing recipe so that you can build it for an updated set of source files.

# GET MORE INFO

[https://wiki.yoctoproject.org/wiki/Developer\\_Workflow\\_Improvements](https://wiki.yoctoproject.org/wiki/Developer_Workflow_Improvements)

A screenshot of the Yocto Project Wiki page titled "Developer Workflow Improvements". The page features a header with the Yocto Project logo and navigation tabs for "page", "discussion", "view source", and "history". Below the header, there is a link to "Back to Yocto Project Main Page" and a "Contents" section with a "[hide]" link. The contents list includes: 1 Introduction (with sub-items 1.1 Layer Maintenance and 1.2 Application Development), 2 Workflow Vision (with sub-items 2.1 Build OS Image, 2.2 Creating Extensible SDK, and 2.3 Application Development). On the left side of the screenshot, there is a "navigation" box with links for "Main page", "Recent changes", and "Random page", and a "search" box below it.

yocto  
PROJECT

page discussion view source history

## Developer Workflow Improvements

Back to Yocto Project [Main Page](#)

**Contents** [hide]

- 1 Introduction
  - 1.1 Layer Maintenance
  - 1.2 Application Development
- 2 Workflow Vision
  - 2.1 Build OS Image
  - 2.2 Creating Extensible SDK
  - 2.3 Application Development

navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)

search



# THANK YOU

Paul Eggleton, Chris Larson, Leo Sandoval and others

Henry Bruce

Todor Minchev, Randy Witt and Brian Avery



# QUESTIONS?



# DISCLAIMER

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

© Intel Corporation

