

Web Hob Stage I

User Test Findings

Main Themes

1. Feedback was **very** positive overall, even from skeptics.

2. No Metrics were missing.



3. Different users have different understandings of some basic concepts:

- Builds
- Images
- Projects

4. Users vary in goals, team size, and experience

builders  optimisers

soloists  big bands

beginners  greybeards

5. Web Hob must accommodate individual workflows.

- Customisation
- Filtering
- Multiple views
- Overridable defaults

6. Users want Web Hob to support their existing workflows.

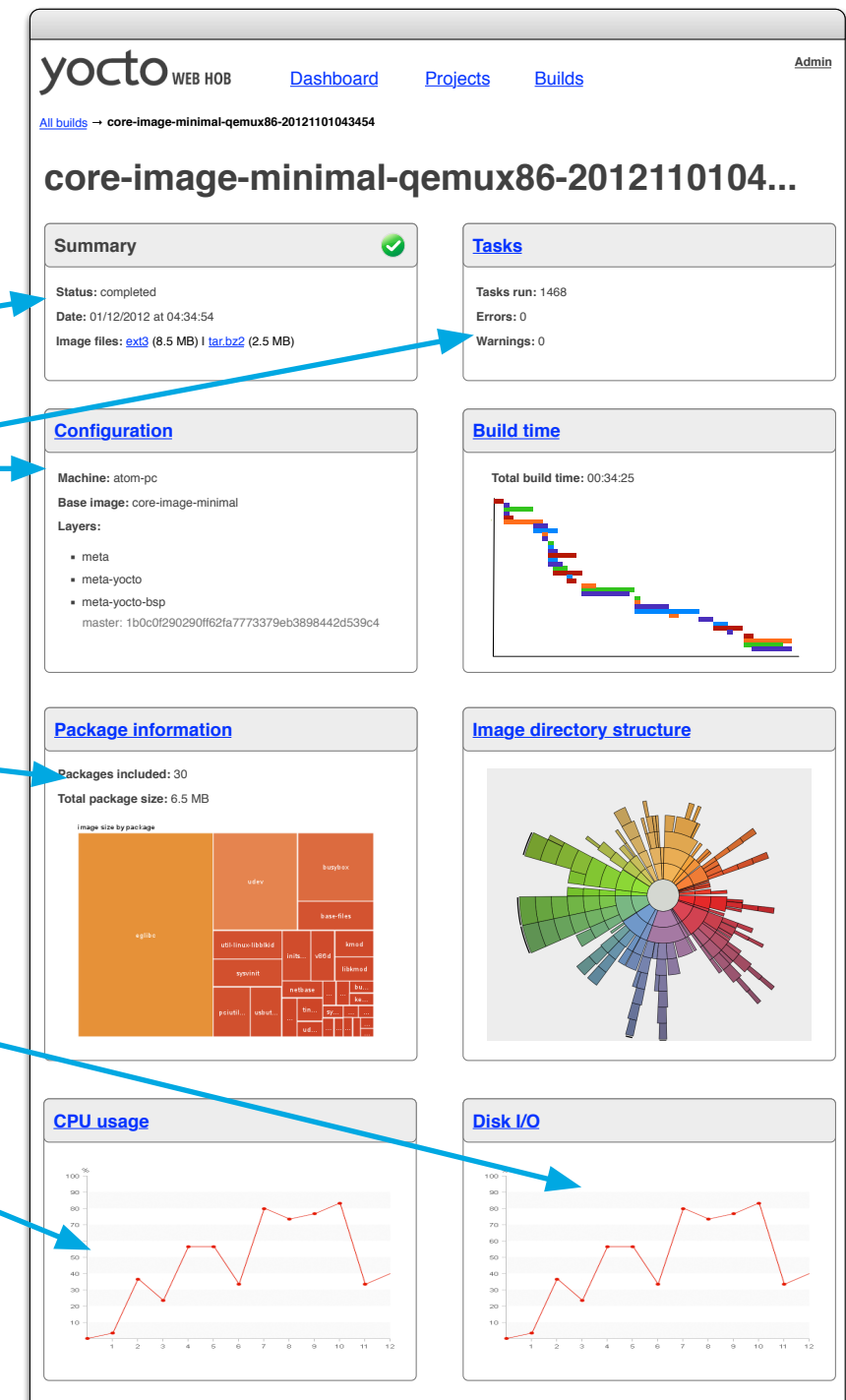
- User management
- Sharing
- Integration with other tools

Build Dashboard



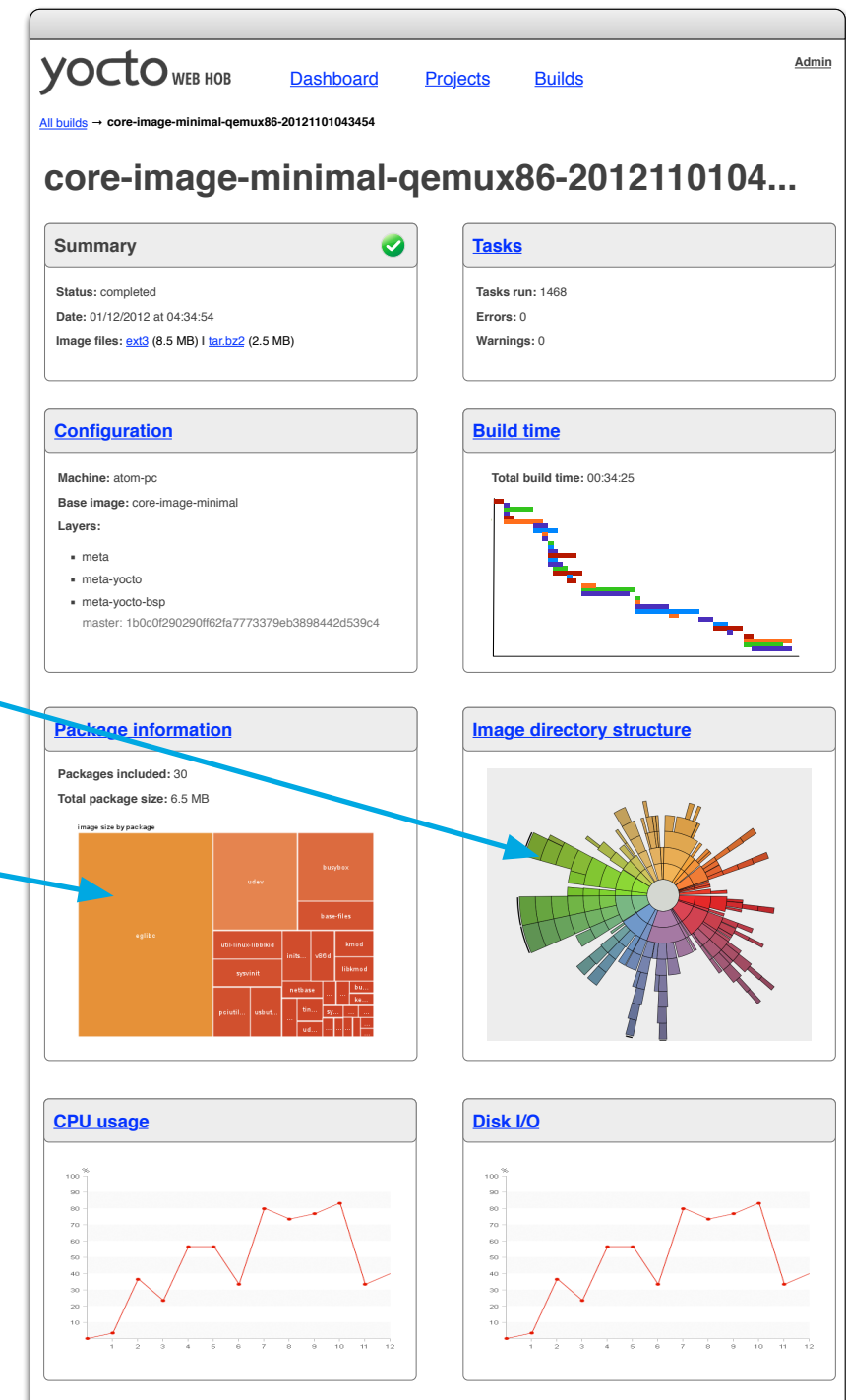
Users grouped information into three clusters:

1. Configuration
2. Diagnostic
3. Optimisation



Some visualisations didn't work for everyone.

- Sunburst diagram
- Treemap



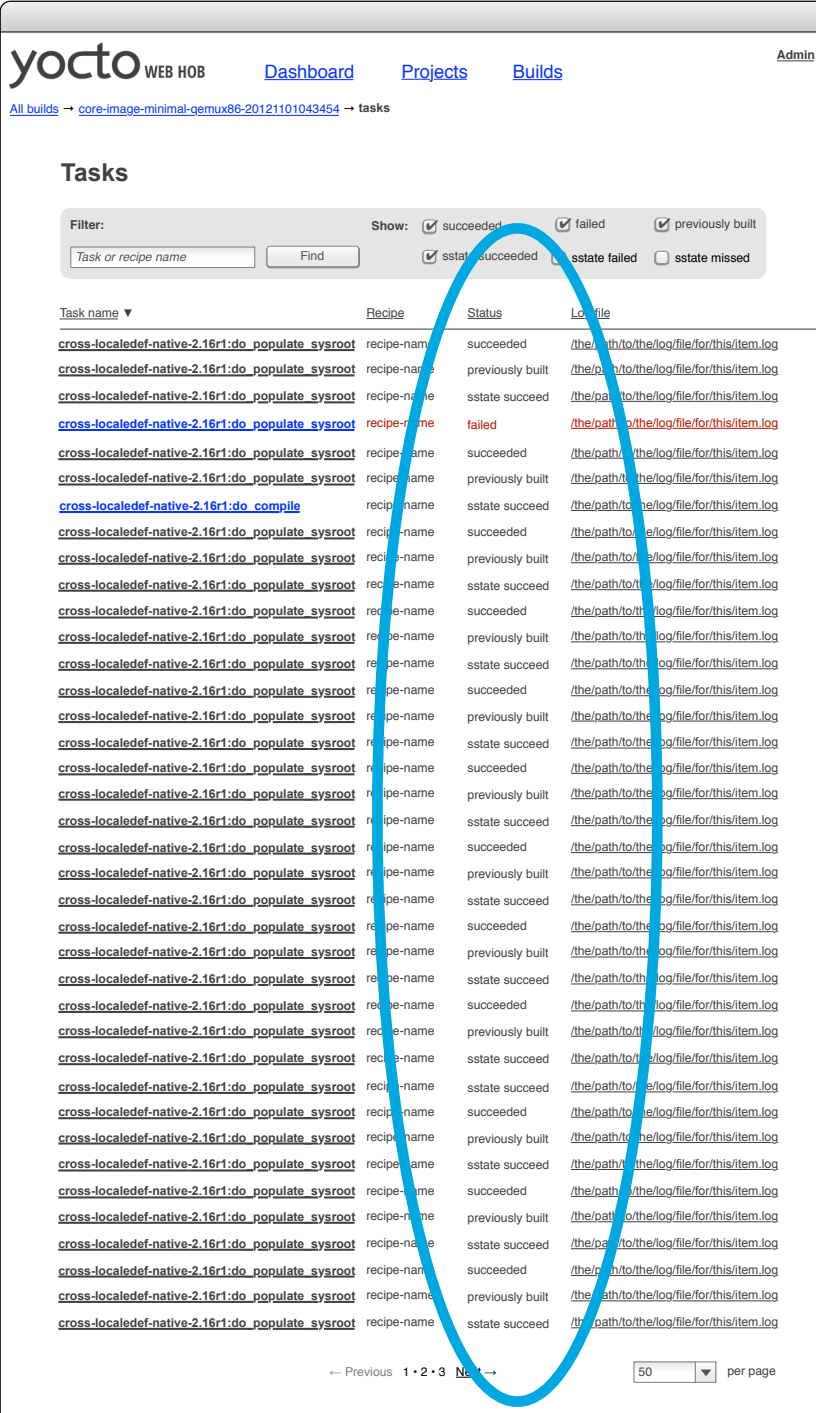
Tasks

[illegible]



But users were unclear about what task types were.

- Succeeded
- Failed
- Previously built
- Sstate succeeded
- Sstate failed
- Sstate missed



The screenshot shows the Yocto Project web interface. At the top, there are navigation links for 'Dashboard', 'Projects', and 'Builds'. Below these, there's a 'Tasks' section with a filter and 'Show' options. The 'Show' options include 'succeeded', 'failed', 'previously built', 'sstate succeeded', 'sstate failed', and 'sstate missed'. A table lists tasks with columns for 'Task name', 'Recipe', 'Status', and 'Log file'. The 'Status' column is circled in blue, highlighting the various task states. The table contains 30 rows of task data.

Task name	Recipe	Status	Log file
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	failed	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_compile	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	succeeded	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	previously built	/the/path/to/the/log/file/for/this/item.log
cross-localedef-native-2.16r1:do_populate_sysroot	recipe-name	sstate succeeded	/the/path/to/the/log/file/for/this/item.log

And we need to rework how ancestor tasks work

- Rename?
- Table view?
- Highlight changes from previous build?



Configuration

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

[All builds](#) → [core-image-minimal-qemux86-20121101043454](#) → Configuration

Configuration

▼ Summary

Target system: i586-poky-linux

Machine: qemux86

Distro: poky

Distro version: 1.3+snapshot-20130114

Layers:

meta

meta-yocto

meta-yocto-bsp (master: 1b0c0f290290ff62fa7773379eb3898442d539c4)

▼ Key / Value pairs

Search key / value pairs

Key	Value	Description	Show in Summary
BB_NUMBER_THREADS	4	Parallelism option to control how much parallelism BitBake should use. This option determines ho many tasks BitBake should run in parallel. For a quad-core machine, BB_NUMBER_THREADS = "4" would be appropriate.	☐
PARALLEL_MAKE	-j 4	Parallelism option to control how much parallelism BitBake should use. This option controls how many processes make should run in parallel when running compile tasks. For a quad-core machine, PARALLEL_MAKE = "-j 4" would be appropriate.	☐
MACHINE	qemux86	This sets the default machine to be qemux86 if no other machine is selected.	☒
DL_DIR	\$(TOPDIR)/downloads	Where to place downloads. During a first build the system will download many different source code tarballs from various upstream projects. This can take a while, particularly if your network connection is slow. These are all stored in DL_DIR. When wiping and rebuilding you can preserve this directory to speed up this part of subsequent builds. This directory is safe to share between multiple builds on the same machine too. The default is a downloads directory under TOPDIR which is the build directory.	☐
SSTATE_DIR	\$(TOPDIR)/sstate-cache	Where to place shared-state files. BitBake has the capability to accelerate builds based on previously built output. This is done using "shared state" files which can be thought of as cache objects and this option determines where those files are placed. You can wipe out TMPDIR leaving this directory intact and the build would regenerate from these files if no changes were made to the configuration. If changes were made to the configuration, only shared state files where the state was still valid would be used (done using checksums). The default is a sstate-cache directory under TOPDIR.	☐
TMPDIR	\$(TOPDIR)/tmp	Where to place the build output. This option specifies where the bulk of the building work should be done and where BitBake should place its temporary files and output. Keep in mind that this includes the extraction and compilation of many applications and the toolchain which can use Gigabytes of hard disk space. The default is a tmp directory under TOPDIR.	☐
DISTRO	poky	Default policy config. The distribution setting controls which policy settings are used as defaults. The default value is fine for general Yocto project use, at least initially. Ultimately when creating custom policy, people will likely end up subclassing these defaults.	☒
PACKAGE_CLASSES	package_rpm	Package Management configuration. This variable lists which packaging formats to enable. Multiple package backends can be enabled at once and the first item listed in the variable will be used to generate the root filesystems. Options are: - 'package_deb' for debian style deb files - 'package_ipk' for ipk files are used by opkg (a debian style embedded package manager) - 'package_rpm' for rpm style packages E.g.: PACKAGE_CLASSES ?= "package_rpm package_deb package_ipk" We default to rpm.	☐

Users need to see which .conf a value comes from, and which have been overridden.

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

[All builds](#) → [core-image-minimal-qemux86-20121101043454](#) → Configuration

Configuration

▼ Summary

Target system: i586-poky-linux

Machine: qemux86

Distro: poky

Distro version: 1.3+snapshot-20130114

Layers:

meta

meta-yocto

meta-yocto-bsp (master: 1b0c0f290290ff62fa7773379eb3898442d539c4)

▼ Key / Value pairs

Search key / value pairs

Key	Value	Description	Show in Summary
BB_NUMBER_THREADS	4	Parallelism option to control how much parallelism BitBake should use. This option determines ho many tasks BitBake should run in parallel. For a quad-core machine, BB_NUMBER_THREADS = "4" would be appropriate.	☐
PARALLEL_MAKE	-j 4	Parallelism option to control how much parallelism BitBake should use. This option controls how many processes make should run in parallel when running compile tasks. For a quad-core machine, PARALLEL_MAKE = "-j 4" would be appropriate.	☐
MACHINE	qemux86	This sets the default machine to be qemux86 if no other machine is selected.	☒
DL_DIR	\$(TOPDIR)/downloads	Where to place downloads. During a first build the system will download many different source code tarballs from various upstream projects. This can take a while, particularly if your network connection is slow. These are all stored in DL_DIR. When wiping and rebuilding you can preserve this directory to speed up this part of subsequent builds. This directory is safe to share between multiple builds on the same machine too. The default is a downloads directory under TOPDIR which is the build directory.	☐
SSTATE_DIR	\$(TOPDIR)/sstate-cache	Where to place shared-state files. BitBake has the capability to accelerate builds based on previously built output. This is done using "shared state" files which can be thought of as cache objects and this option determines where those files are placed. You can wipe out TMPDIR leaving this directory intact and the build would regenerate from these files if no changes were made to the configuration. If changes were made to the configuration, only shared state files where the state was still valid would be used (done using checksums). The default is a sstate-cache directory under TOPDIR.	☐
TMPDIR	\$(TOPDIR)/tmp	Where to place the build output. This option specifies where the bulk of the building work should be done and where BitBake should place its temporary files and output. Keep in mind that this includes the extraction and compilation of many applications and the toolchain which can use Gigabytes of hard disk space. The default is a tmp directory under TOPDIR.	☐
DISTRO	poky	Default policy config. The distribution setting controls which policy settings are used as defaults. The default value is fine for general Yocto project use, at least initially. Ultimately when creating custom policy, people will likely end up subclassing these defaults.	☒
PACKAGE_CLASSES	package_rpm	Package Management configuration. This variable lists which packaging formats to enable. Multiple package backends can be enabled at once and the first item listed in the variable will be used to generate the root filesystems. Options are: - 'package_deb' for debian style deb files - 'package_ipk' for ipk files are used by opkg (a debian style embedded package manager) - 'package_rpm' for rpm style packages E.g.: PACKAGE_CLASSES ?= "package_rpm package_deb package_ipk" We default to rpm.	☐

Users want to customise the information display.

- Select summary values per-build
- Show only changed values
- Show only values from local .conf

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

[All builds](#) → [core-image-minimal-qemux86-20121101043454](#) → Configuration

Configuration

▼ Summary

Target system: i586-poky-linux

Machine: qemux86

Distro: poky

Distro version: 1.3+snapshot-20130114

Layers:

meta

meta-yocto

meta-yocto-bsp (master: 1b0c0f290290ff62fa7773379eb3898442d539c4)

▼ Key / Value pairs

Search key / value pairs

Key	Value	Description	Show in Summary
BB_NUMBER_THREADS	4	Parallelism option to control how much parallelism BitBake should use. This option determines how many tasks BitBake should run in parallel. For a quad-core machine, BB_NUMBER_THREADS = "4" would be appropriate.	☐
PARALLEL_MAKE	-j 4	Parallelism option to control how much parallelism BitBake should use. This option controls how many processes make should run in parallel when running compile tasks. For a quad-core machine, PARALLEL_MAKE = "-j 4" would be appropriate.	☐
MACHINE	qemux86	This sets the default machine to be qemux86 if no other machine is selected.	☒
DL_DIR	\$(TOPDIR)/downloads	Where to place downloads. During a first build the system will download many different source code tarballs from various upstream projects. This can take a while, particularly if your network connection is slow. These are all stored in DL_DIR. When wiping and rebuilding you can preserve this directory to speed up this part of subsequent builds. This directory is safe to share between multiple builds on the same machine too. The default is a downloads directory under TOPDIR which is the build directory.	☐
SSTATE_DIR	\$(TOPDIR)/sstate-cache	Where to place shared-state files. BitBake has the capability to accelerate builds based on previously built output. This is done using "shared state" files which can be thought of as cache objects and this option determines where those files are placed. You can wipe out TMPDIR leaving this directory intact and the build would regenerate from these files if no changes were made to the configuration. If changes were made to the configuration, only shared state files where the state was still valid would be used (done using checksums). The default is a sstate-cache directory under TOPDIR.	☐
TMPDIR	\$(TOPDIR)/tmp	Where to place the build output. This option specifies where the bulk of the building work should be done and where BitBake should place its temporary files and output. Keep in mind that this includes the extraction and compilation of many applications and the toolchain which can use Gigabytes of hard disk space. The default is a tmp directory under TOPDIR.	☐
DISTRO	poky	Default policy config. The distribution setting controls which policy settings are used as defaults. The default value is fine for general Yocto project use, at least initially. Ultimately when creating custom policy, people will likely end up subclassing these defaults.	☒
PACKAGE_CLASSES	package_rpm	Package Management configuration. This variable lists which packaging formats to enable. Multiple package backends can be enabled at once and the first item listed in the variable will be used to generate the root filesystems. Options are: - 'package_deb' for debian style deb files - 'package_ipk' for ipk files are used by opkg (a debian style embedded package manager) - 'package_rpm' for rpm style packages E.g.: PACKAGE_CLASSES ?= "package_rpm package_deb package_ipk" We default to rpm.	☐

The background information is useful, but may be too prominent.

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

[All builds](#) → [core-image-minimal-qemux86-20121101043454](#) → Configuration

Configuration

▼ Summary

Target system: i586-poky-linux

Machine: qemux86

Distro: poky

Distro version: 1.3+snapshot-20130114

Layers:

meta

meta-yocto

meta-yocto-bsp (master: 1b0c0f290290ff62fa7773379eb3898442d539c4)

▼ Key / Value pairs

Search key / value pairs

Key	Value	Description	Show in Summary
BB_NUMBER_THREADS	4	Parallelism option to control how much parallelism BitBake should use. This option determines how many tasks BitBake should run in parallel. For a quad-core machine, BB_NUMBER_THREADS = "4" would be appropriate.	☐
PARALLEL_MAKE	-j 4	Parallelism option to control how much parallelism BitBake should use. This option controls how many processes make should run in parallel when running compilation tasks. For a quad-core machine, PARALLEL_MAKE = "-j 4" would be appropriate.	☐
MACHINE	qemux86	This sets the default machine to be qemux86 if no other machine is selected.	☒
DL_DIR	\$(TOPDIR)/download	Where to place downloads. During a first build the system will download many different source code tarballs from various upstream projects. This can take a while, particularly if your network connection is slow. These are all stored in DL_DIR. When wiping and rebuilding you can preserve this directory to speed up this part of subsequent builds. This directory is safe to share between multiple builds on the same machine too. The default is a downloads directory under TOPDIR which is the build directory.	☐
SSTATE_DIR	\$(TOPDIR)/ssstate-cache	Where to place shared-state files. BitBake has the capability to accelerate builds based on previously built output. This is done using "shared state" files which can be thought of as cache objects and this option determines where those files are placed. You can wipe out TMPDIR leaving this directory intact and the build would regenerate from these files if no changes were made to the configuration. If changes were made to the configuration, only shared state files where the state was still valid would be used (done using checksums). The default is a sstate-cache directory under TOPDIR.	☐
TMPDIR	\$(TOPDIR)/tmp	Where to place the build output. This option specifies where the bulk of the building work should be done and where BitBake should place its temporary files and output. Keep in mind that this includes the extraction and compilation of many applications and the toolchain which can use Gigabytes of hard disk space. The default is a tmp directory under TOPDIR.	☐
DISTRO	poky	Default policy config. The distribution setting controls which policy settings are used as defaults. The default value is fine for general Yocto project use, at least initially. Ultimately when creating custom policy, people will likely end up subclassing these defaults.	☒
PACKAGE_CLASSES	package_rpm	Package Management configuration. This variable lists which packaging formats to enable. Multiple package backends can be enabled at once and the first item listed in the variable will be used to generate the root filesystems. Options are: - 'package_deb' for debian style deb files - 'package_ipk' for ipk files are used by opkg (a debian style embedded package manager) - 'package_rpm' for rpm style packages E.g. PACKAGE_CLASSES ?= "package_rpm package_deb package_ipk" We default to rpm.	☐

Package Information & Dependencies

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

All builds

→

core-image-minimal-qemu86-20121101043454

→

Package information

Packages Included: 30

Total package size: 6.5 MB

Package information

All packages

Size

Dependencies

Search packages

Package name

Recipe

Size

image size by package

eglibc

udev

busybox

base-files

util-linux-libkld

initrd

libmod

sysvinit

netbase

bc

privat...

udev...

fs...

Sc...

ud...

yocto

WEB HOB

Dashboard

Projects

Builds

Admin

All builds

→

core-image-minimal-qemu86-20121101043454

→

Package information

Packages Included: 30

Total package size: 6.5 MB

Package information

All packages

Size

Dependencies

Search packages

Package name	Depends on	Recommends
base-files		
base-passwd		
busybox	- update-rc.d - eglibc	- busybox-syslog - busybox-udhcp
busybox-hwclock		
busybox-syslog	update-alternatives-cworth	
busybox-udhcp		
eglibc		
initscripts	update-alternatives-cworth	
kernel-base		
kernel-module-cfbcopyarea	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbfillrect	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbimgblt	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-uvesafb	- kernel-3.4.24-yocto-standard - kernel-module-cfbcopyarea - kernel-module-cfbfillrect - kernel-module-cfbimgblt - update-modules	
kmod	- eglibc - libkmod - update-alternatives-cworth	
libkmod	eglibc	
module-initscripts	update-rc.d	
netbase	update-rc.d	
packagegroup-core-boot	- base-files - base-passwd - busybox - busybox-hwclock - initscripts - module-initscripts - netbase - sysvinit - tinylogin - udev - update-alternatives-cworth - v86d	
pclutils-ids		
sysvinit	- eglibc - sysvinit-inittab - sysvinit-pidof - update-alternatives-cworth	
sysvinit-inittab		
sysvinit-pidof	- eglibc - update-alternatives-cworth	
tinylogin	- eglibc - update-alternatives-cworth	

busybox-syslog

update-rc.d

eglibc

busybox-udhcp

busybox

Users looooooove the network diagram!

yoctoWEB HOB

[Dashboard](#)

[Projects](#)

[Builds](#)

Admin

All builds → core-image-minimal-qemu86-20121101043454 → Package information

Packages included: 30
Total package size: 6.5 MB

Package information

All packages

Search packages

Package name	Depends on	Recommends
base-files		
base-passwd		
busybox	- update-rc.d - eglibc	- busybox-syslog - busybox-udhcp
busybox-hwclock		
busybox-syslog	update-alternatives-cworth	
busybox-udhcp		
eglibc		
initscripts	update-alternatives-cworth	
kernel-base		
kernel-module-cfbcopyarea	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbfillrect	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbimgblt	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-uvesafb	- kernel-3.4.24-yocto-standard - kernel-module-cfbcopyarea - kernel-module-cfbfillrect - kernel-module-cfbimgblt - update-modules	
kmod	- eglibc - libkmod - update-alternatives-cworth	
libkmod	eglibc	
modutils-initscripts	update-rc.d	
netbase	update-rc.d	
packagegroup-core-boot	- base-files - base-passwd - busybox - busybox-hwclock - initscripts - modutils-initscripts - netbase - sysvinit - tinylogin - udev - update-alternatives-cworth - v86d	
pciutils-ids		
sysvinit	- eglibc - sysvinit-inittab - sysvinit-pidof - update-alternatives-cworth	
sysvinit-inittab		
sysvinit-pidof	- eglibc - update-alternatives-cworth	
tinylogin	- eglibc - update-alternatives-cworth	

But recommended packages should probably be hidden by default.

yoctoWEB HOB

DashboardProjectsBuildsAdmin

All builds → core-image-minimal-gemux86-20121101043454 → Package information

Packages included: 30
Total package size: 6.5 MB

Package information

All packages

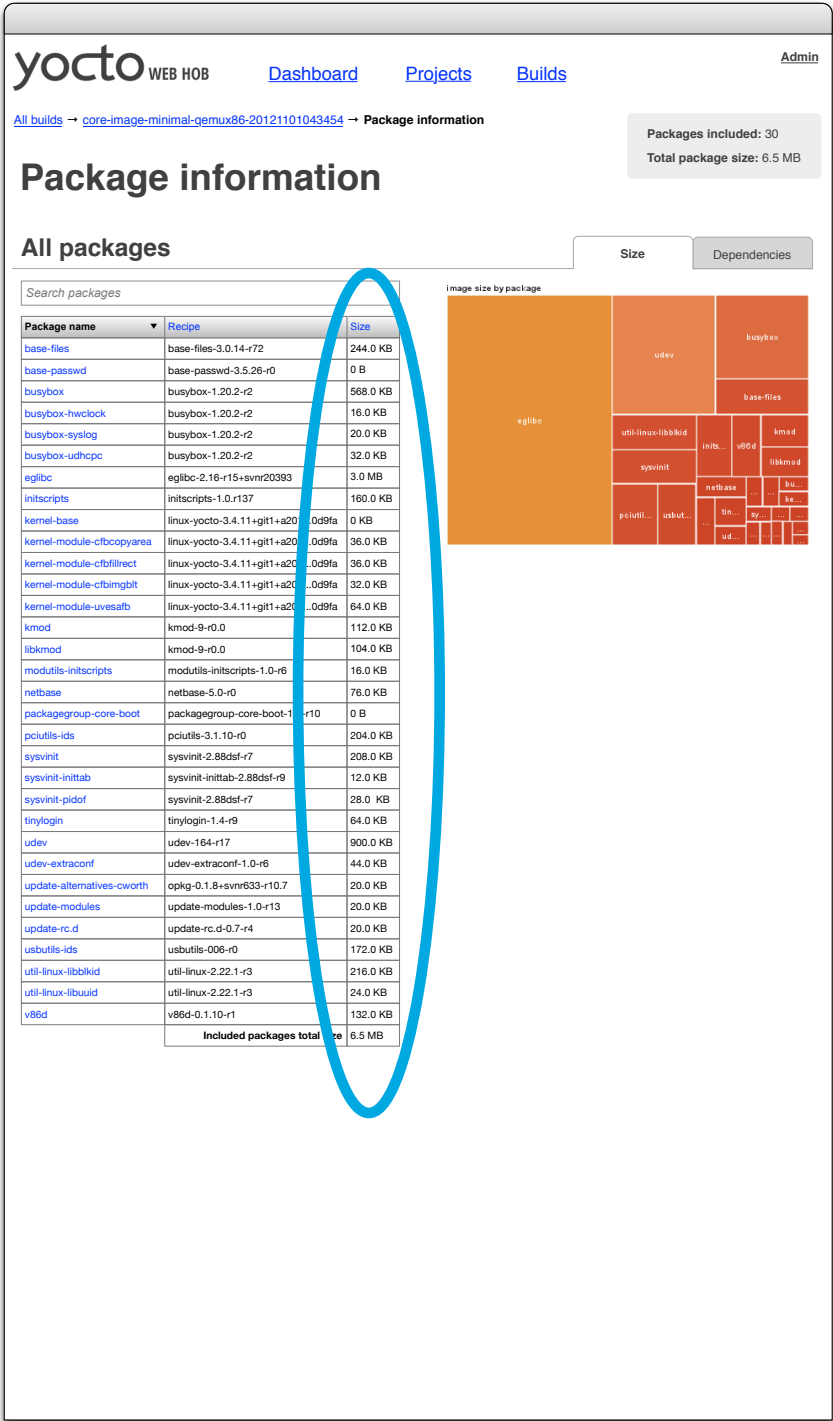
SizeDependencies

Search packages

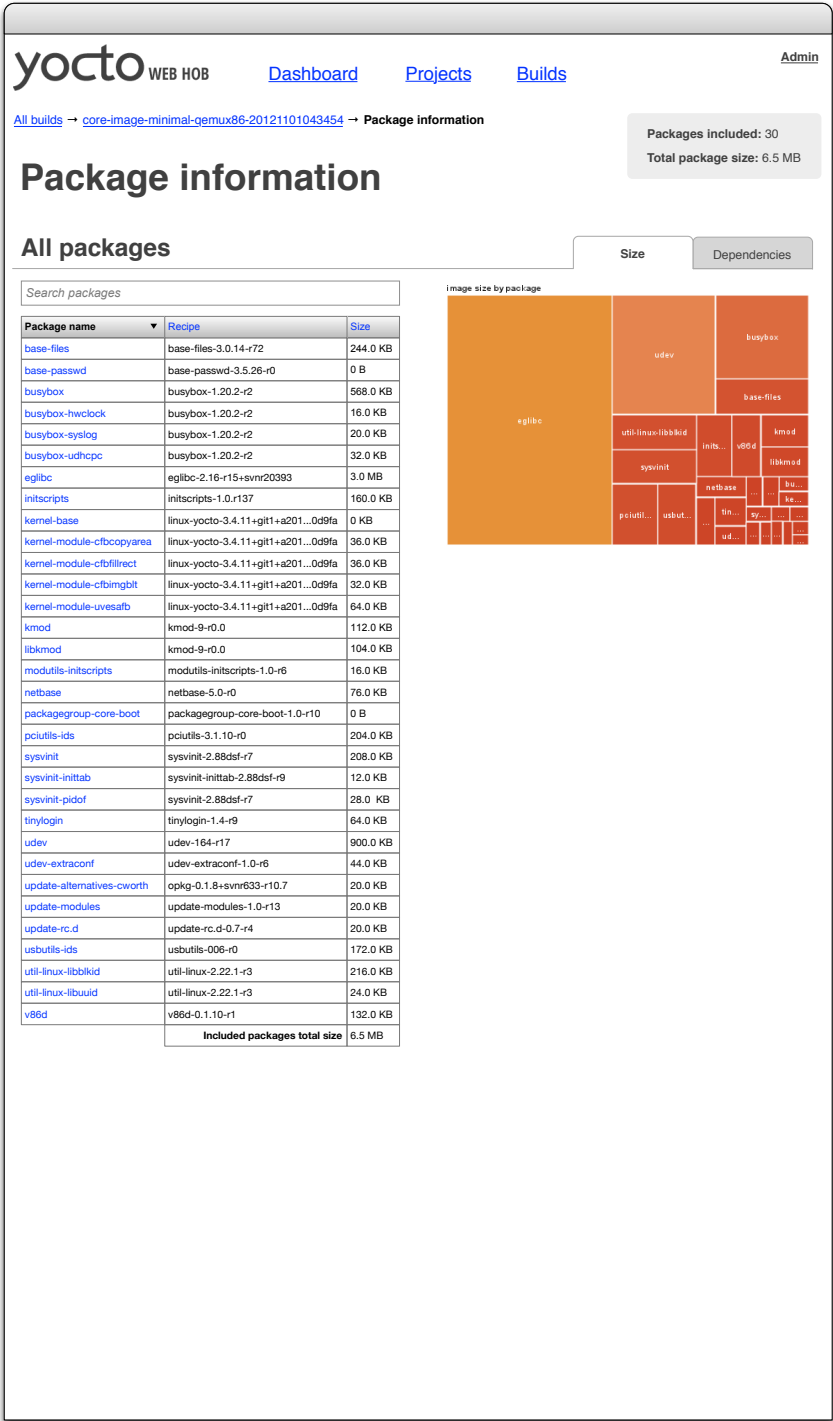
Package name	Depends on	Recommends
base-files		
base-passwd		
busybox	- update-rc.d - eglibc	- busybox-syslog - busybox-udhcp
busybox-hwclock		
busybox-syslog	update-alternatives-cworth	
busybox-udhcp		
eglibc		
initscripts	update-alternatives-cworth	
kernel-base		
kernel-module-cfbcopyarea	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbfillrect	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-cfbimgblt	- kernel-3.4.24-yocto-standard - update-modules	
kernel-module-uvesafb	- kernel-3.4.24-yocto-standard - kernel-module-cfbcopyarea - kernel-module-cfbfillrect - kernel-module-cfbimgblt - update-modules	
kmod	- eglibc - libkmod - update-alternatives-cworth	
libkmod	eglibc	
modutils-initscripts	update-rc.d	
netbase	update-rc.d	
packagegroup-core-boot	- base-files - base-passwd - busybox - busybox-hwclock - initscripts - modutils-initscripts - netbase - sysvinit - tinylogin - udev - update-alternatives-cworth - v86d	
pciutils-ids		
sysvinit	- eglibc - sysvinit-inittab - sysvinit-pidof - update-alternatives-cworth	
sysvinit-inittab		
sysvinit-pidof	- eglibc - update-alternatives-cworth	
tinylogin	- eglibc - update-alternatives-cworth	

```
graph TD; busybox((busybox)) --- update_rc_d((update-rc.d)); busybox --- eglibc((eglibc)); busybox --- busybox_syslog((busybox-syslog)); busybox --- busybox_udhcp((busybox-udhcp)); update_rc_d --- eglibc; update_rc_d --- busybox_syslog; update_rc_d --- busybox_udhcp; eglibc --- busybox_syslog; eglibc --- busybox_udhcp; busybox_syslog --- update_alts_cworth((update-alternatives-cworth)); update_alts_cworth --- busybox_udhcp; update_alts_cworth --- kmod((kmod)); update_alts_cworth --- libkmod((libkmod)); update_alts_cworth --- modutils_initscripts((modutils-initscripts)); update_alts_cworth --- netbase((netbase)); update_alts_cworth --- packagegroup_core_boot((packagegroup-core-boot)); update_alts_cworth --- pciutils_ids((pciutils-ids)); update_alts_cworth --- sysvinit((sysvinit)); update_alts_cworth --- sysvinit_inittab((sysvinit-inittab)); update_alts_cworth --- sysvinit_pidof((sysvinit-pidof)); update_alts_cworth --- tinylogin((tinylogin)); kmod --- libkmod; kmod --- modutils_initscripts; kmod --- netbase; kmod --- packagegroup_core_boot; kmod --- pciutils_ids; kmod --- sysvinit; kmod --- sysvinit_inittab; kmod --- sysvinit_pidof; kmod --- tinylogin; libkmod --- modutils_initscripts; libkmod --- netbase; libkmod --- packagegroup_core_boot; libkmod --- pciutils_ids; libkmod --- sysvinit; libkmod --- sysvinit_inittab; libkmod --- sysvinit_pidof; libkmod --- tinylogin; modutils_initscripts --- netbase; modutils_initscripts --- packagegroup_core_boot; modutils_initscripts --- pciutils_ids; modutils_initscripts --- sysvinit; modutils_initscripts --- sysvinit_inittab; modutils_initscripts --- sysvinit_pidof; modutils_initscripts --- tinylogin; netbase --- packagegroup_core_boot; netbase --- pciutils_ids; netbase --- sysvinit; netbase --- sysvinit_inittab; netbase --- sysvinit_pidof; netbase --- tinylogin; packagegroup_core_boot --- pciutils_ids; packagegroup_core_boot --- sysvinit; packagegroup_core_boot --- sysvinit_inittab; packagegroup_core_boot --- sysvinit_pidof; packagegroup_core_boot --- tinylogin; pciutils_ids --- sysvinit; pciutils_ids --- sysvinit_inittab; pciutils_ids --- sysvinit_pidof; pciutils_ids --- tinylogin; sysvinit --- sysvinit_inittab; sysvinit --- sysvinit_pidof; sysvinit --- tinylogin; sysvinit_inittab --- sysvinit_pidof; sysvinit_inittab --- tinylogin; sysvinit_pidof --- tinylogin;
```

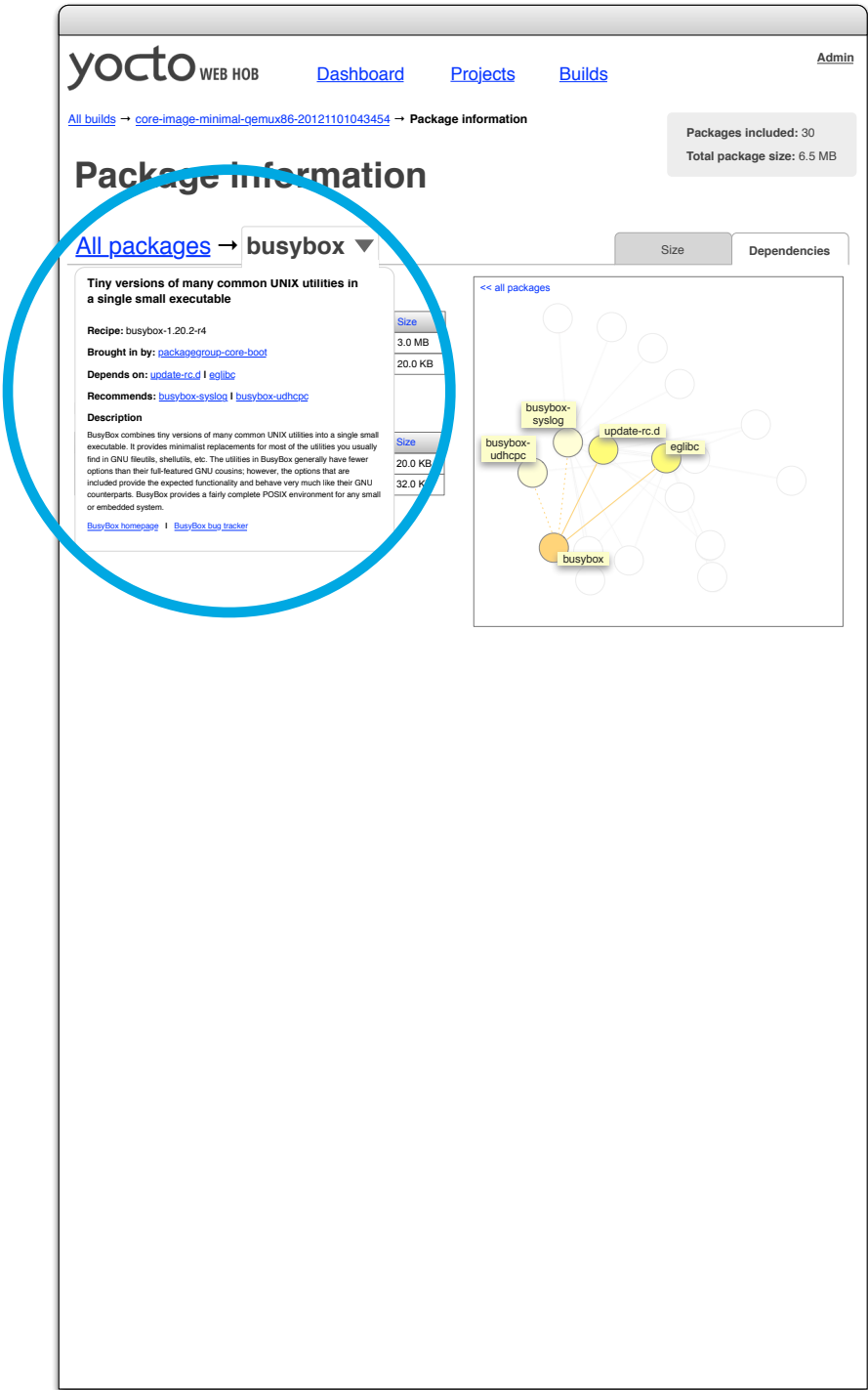
In addition to absolute size, users want to see what percentage of the total image a package takes up.



Some users wanted to see native & support packages as well as deployed packages.



Package context information should always be visible.



Package file view was useful to find where packages put files, especially for relative beginners.

