# *Yocto AutoBuilder*

# Yocto AutoBuilder

- Basic Setup

```
1 git clone git://git.yoctoproject.org/yocto-autobuilder
2 cd yocto-autobuilder
3 . ./yocto-autobuilder-setup
4 ./yocto-start-autobuilder <master/slave/both>
```

- If you want a more advanced setup, more hacking require(e.g. master.cfg/buildbot.tac)

# Yocto AutoBuilder

## Modify the conf/autobuilder.conf

- PACKAGE_CLASSES
- BB_NUMBER_THREADS
- PARALLEL_MAKE
- MACHINE
- PUBLISH_BUILDS
- PUBLISH_SSTATE
- …

# Yocto AutoBuilder

## yocto-master/master.cfg

```
c= BuildmasterConfig = {}
from buildbot.buildslave import BuildSlave
c['slaves'] = [BuildSlave("builder1", "linux123", max_builds=1)]
c['slavePortnum'] = 9989
c['status'] = []
from buildbot.status.html import WebStatus
from buildbot.status.web.authz import Authz
from buildbot.status.web.auth import BasicAuth
users = [('leoma-HP-Compaq-8200-Elite-SFF-PC', 'linux123')]
authz_cfg=Authz(auth=BasicAuth(users),
    forceBuild = 'auth',
    #forceBuild = True,
    stopBuild = 'auth',
    cancelPendingBuild = True,
    pingBuilder = False)
c['status'].append(WebStatus(http_port=8010, authz=authz_cfg))
c['debugPassword'] = "linux123"
c['buildbotURL'] = "http://localhost:9989/"
```

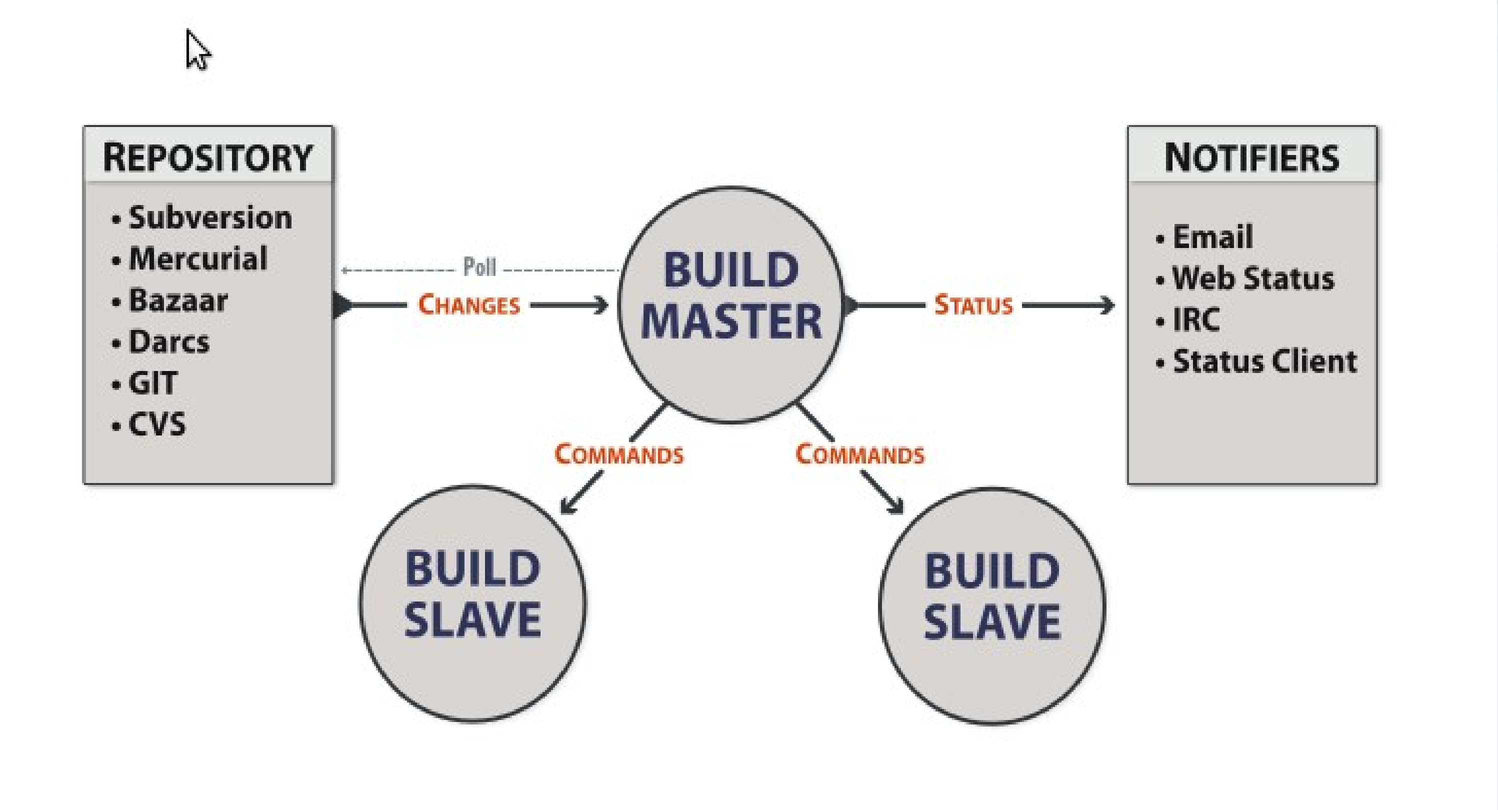
# Yocto AutoBuilder

## yocto-slave/buildbot.tac

```
buildmaster_host = 'localhost'
port = 9989
slavename = 'builder1'
passwd = 'linux123'
keepalive = 600
usepty = 0
umask = 022
maxdelay = 300
```

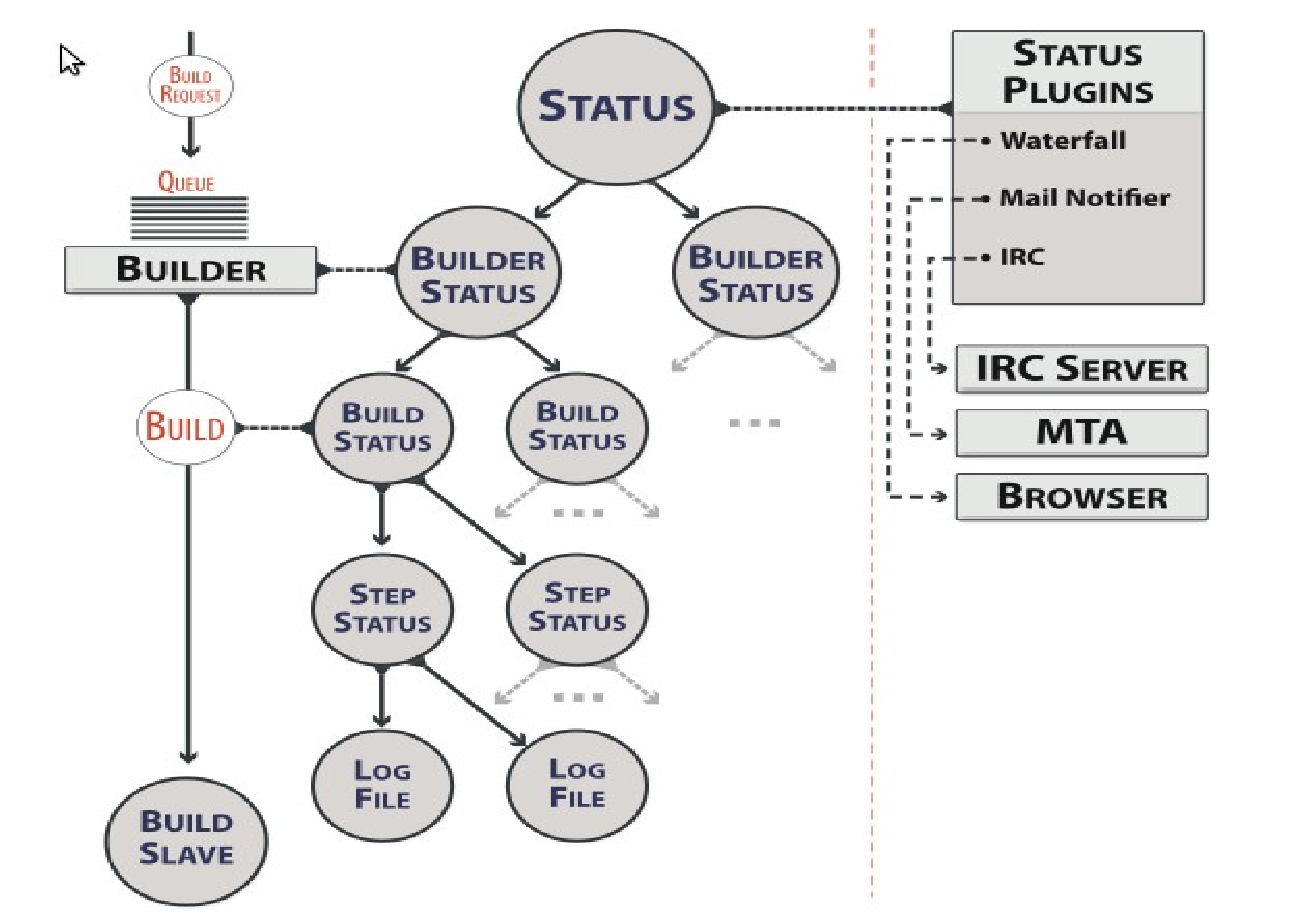
# Yocto AutoBuilder

## Master and Slave



# Yocto AutoBuilder

## Master and Slave



# Yocto AutoBuilder

- Build information

  yocto.sh.intel.com:8010
- Twistd.log

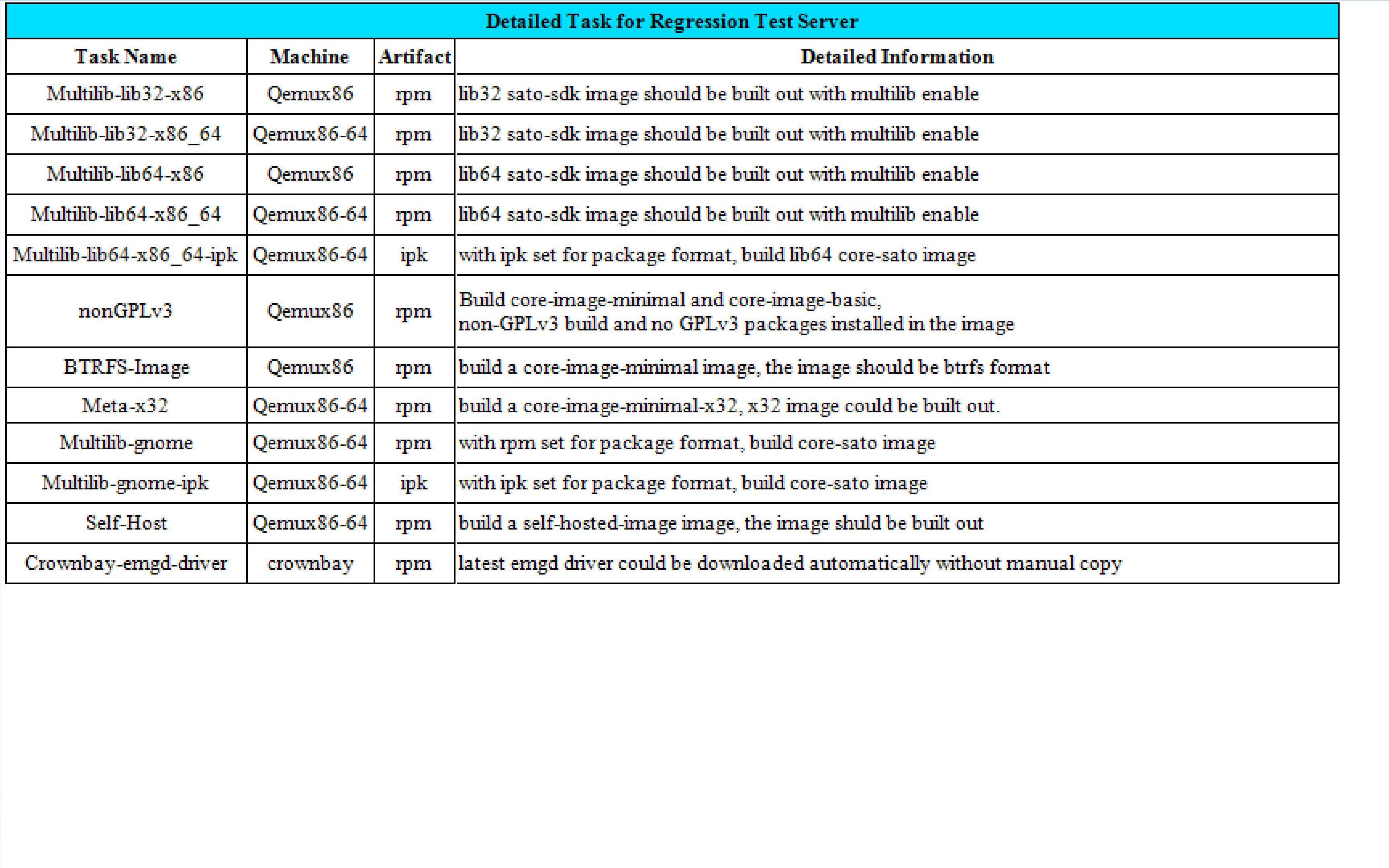
  yocto-master/twistd.log

  yocto-slave/twistd.log
- Pubilsh_build_dir of Regression Test Server

  \\yocto.sh.intel.com\incoming\Autobuilder\yocto-autobuilder

# Yocto AutoBuilder

- Detail Tasks for test Server



**Detailed Task for Regression Test Server**

| Task Name | Machine | Artifact | Detailed Information |
|---|---|---|---|
| Multilib-lib32-x86 | Qemux86 | rpm | lib32 sato-sdk image should be built out with multilib enable |
| Multilib-lib32-x86_64 | Qemux86-64 | rpm | lib32 sato-sdk image should be built out with multilib enable |
| Multilib-lib64-x86 | Qemux86 | rpm | lib64 sato-sdk image should be built out with multilib enable |
| Multilib-lib64-x86_64 | Qemux86-64 | rpm | lib64 sato-sdk image should be built out with multilib enable |
| Multilib-lib64-x86_64-ipk | Qemux86-64 | ipk | with ipk set for package format, build lib64 core-sato image |
| nonGPLv3 | Qemux86 | rpm | Build core-image-minimal and core-image-basic,<br>non-GPLv3 build and no GPLv3 packages installed in the image |
| BTRFS-Image | Qemux86 | rpm | build a core-image-minimal image, the image should be btrfs format |
| Meta-x32 | Qemux86-64 | rpm | build a core-image-minimal-x32, x32 image could be built out. |
| Multilib-gnome | Qemux86-64 | rpm | with rpm set for package format, build core-sato image |
| Multilib-gnome-ipk | Qemux86-64 | ipk | with ipk set for package format, build core-sato image |
| Self-Host | Qemux86-64 | rpm | build a self-hosted-image image, the image shuld be built out |
| Crownbay-emgd-driver | crownbay | rpm | latest emgd driver could be downloaded automatically without manual copy |