Yocto 1.1 M1 Fullpass Test

Test Plan

Project: yocto

Author: tester

Printed by TestLink on 25/05/2011

Table Of Contents

Yocto 1.1 M1 Fullpass Test

System & Core OS

zypper command installed and workable

zypper help search

zypper search package

zypper remove package

zypper install package

zypper install dependency package

zypper verify packages

rpm query package

rpm install package

rpm install dependency package

rpm remove package

boot and install from USB

live boot from USB

boot from runlevel 3

boot from runlevel 5

g++ compile in sdk image

gcc compile in sdk image

run command make in sdk image

cvs project compile in sdk image

iptables project compile in sdk image

sudoku-savant project compile in sdk image

perl program work in image

shutdown system

reboot system

adjust date and time

switch among multi applications and desktop

vncserver for target

file manager

system dmesg log check

usb mount

usb read files

usb umount

usb write files

file copy by scp

connman launch after boot

ethernet enabled in connman

only one connmand in background

remote access by ssh

ethernet static ip set in connman

ethernet get IP in connman via DHCP

connman offline mode in connman-gnome

X server can start up with runlevel 5 boot

qt application quicky

standby

Test if LAN device works well after resume from suspend state

Test if usb hid device works well after resume from suspend state

ADT

gcc from ADT toolchain can build c program

g++ from ADT toolchain can build c program

ADT toolchain could build cvs project

ADT toolchain could build iptables project

ADT toolchain could build sudoku-savant project

unfs support for qemu target

Stress

crashme for stress

helltest for stress

Power/Performance

boot time collection

memory footprint

powertop log

Idle power consumption

Graphics

Graphics ABAT

openarena - 3D

urbanterror - 3D

x11perf - 2D

Mulitimedia

sound on/off

audio play (mp3)

audio play (ogg)

audio stop (ogg)

audio play (wav)

audio stop (wav)

video play (mpeg)

video play (ogg)

video stop (ogg)

Compliance

LTP subset test suite

POSIX subset test suite

LSB subset test suite

Core Build System

kernel interactive targets

KVM enabled with qemu

non-GPLv3 build check

yocto build in Fedora 14

yocto build in OpenSuse 11.4

yocto build in Ubuntu 11.04

yocto build in KVM

sstate work on local host

Regression

disk space check

click terminal icon on X desktop

Add multiple files in music player

system shutdown with UNFS

no connman-gnome icon on desktop

application contact should work

---

1 Test Suite : Yocto 1.1 M1 Fullpass Test

1.1 Test Suite : System & Core OS

Test Case TC-599: zypper command installed and workable	
Summary: check if zypper is installed and can work	
Steps: 1. Run command "zypper", and check the output	
Expected Results: Command "zypper" print the list of available global options and commands	
Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-600: zypper help search	
Summary: check help option with zypper command	
Steps: 1. Run "zypper help search" and check the output	
Expected Results: The command should print help for the search command	
Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-601: zypper search package	
Summary: search package with zypper	
Steps: 1. Run "zypper search package_name" and check the output, for example "zypper search avahi"	
Expected Results: The command should search package "avahi" is installed or not	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Auto
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-602: zypper remove package	
Summary: remove package with zypper	
Steps: 1. Run "zypper rm pakcage_name" and check the output, for example "zypper rm avahi"	
Expected Results: The command should remove package "avahi"	
Test Execution	Weekly

Cycle Type:	
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-603: zypper install package	
Summary: install package with zypper	
Steps: 1. Set up a yum based repository on local server 2. Build out a package, which does not need any run-time dependency package, with local poky tree. For example, package "man" 3. In target system, run "zypper addrepo http://ip_address_of_repository zypper_test_repo" 4. Run "zypper refresh" to refresh the zypper repository cache 5. Run "zypper install package_name" and check the output, for example "zypper install man" to install package, which has no run-time dependency	
Expected Results: The command should install package "man"	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-604: zypper install dependency package
Summary: install dependency package with zypper
Steps: 1. Set up a yum based repository on local server 2. Build out a package, which does not need any run-time dependency package, with local poky tree. For example, package "mc" 3. In target system, run "zypper addrepo http://ip_address_of_repository zypper_test_repo"

4. Run "zypper refresh" to refresh the zypper repository cache 5. Run "zypper install package_name" and check the output, for example "zypper install mc" to install package, which needs run-time dependency packages installed also, like ncurses-terminfo.
Expected Results: The command should install package "mc" and denpendency package ncurses-terminfo.

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-674: zypper verify packages
Summary: Make sure zypper verify could work without problem
Steps: 1. Set up a yum based repository on local server 2. Build out some packages with suffix .all, .${MACHINE}, .${TARGET_ARCH}, taking qemux86 as example, we could build update-rc.d-0.x-rx.all.rpm, task-core-boot-1.0-rx.qemux86.rpm and zip-3.0-rx.i586.rpm and put them into the yum repository. 3. In target system, run "zypper addrepo http://ip_address_of_repository zypper_test_repo" 4. Run "zypper refresh" to refresh the zypper repository cache 5. Run "zypper verify" and check if there is any error message.
Expected Results: zypper verify should work without error

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, netbook, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-625: rpm query package
Summary: make sure rootfs image is built with rpm packages

Steps: 1. launch terminal 2. run command "rpm -qa", which lists all existing packages in system	
Expected Results: "rpm -qa" should print all existing packages in system	
Test Execution Cycle Type:	Sanity
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-624: rpm install package	
Summary: rpm format package can be installed	
Steps: 1. Get a RPM package(for example, avahi or powertop) from zypper repository or build one on local machine 2. Copy the package into image, run command "rpm -ivh package_name" to install the package	
Expected Results: RPM format package can be installed	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-623: rpm install dependency package
Summary: rpm command should report dependency when installing package
Steps: 1. Get a RPM package or build one on local machine, which should have run-time dependency. For example, mc RPM should depends on ncurses-terminfo

2. Run "rpm -ivh package_name" and check the output, for example "rpm -ivh mc.rpm*" should report the dependency on ncurses-terminfo	
Expected Results: rpm command should report message when some RPM installation depends on other packages	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-626: rpm remove package	
Summary: rpm command can remove package in system	
Steps: 1. Launch terminal and run command "rpm -e package_name" to remove some package, for example, avahi	
Expected Results: RPM package can be removed by command rpm	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-606: boot and install from USB	
Summary: boot and install image from usb stick	
Steps: 1. plugin usb which contains live image burned 2. configure device BIOS to firstly boot from USB if necessary 3. boot the device and select some option like "Boot and Install" from boot menu 4. proceed through default install process 5. Remove USB, and reboot into new installed system.	
Expected Results: 1. User can choose install system from usb stick onto harddisk from boot menu or command line option	

2. Imstalled system can boot up	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	installation&boot
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-612: live boot from USB	
Summary: live boot from USB	
Steps: boot live image from usb stick 1. plugin usb which contains live image burned 2. configure device BIOS to firstly boot from USB if necessary 3. boot the device and select some option like "boot from usb" from boot menu	
Expected Results: 1. User can choose boot from live image on usb stick from boot menu or command line option 2. Live image can boot up with usb stick	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	installation&boot
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-607: boot from runlevel 3	
Summary: Verify that system can boot from runlevel 3	
Steps: 1. Boot into system and edit /etc/inittab to make sure system enter init 3 by default ######## id:3:initdefault ######## 2. reboot system, and press Tab to enter "grub" 3. edit "kernel" line and add "psplash=false text" at the end 4. Press "enter" to boot system	
Expected Results: system should boot to runlevel 3.	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual

Case State:	Ready
Feature:	installation&boot
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-608: boot from runlevel 5	
Summary: Verify that system can boot from runlevel 5	
Steps: 1. Boot into system and edit /etc/inittab to make sure system enter init 5 by default ######## id:5:initdefault ######## 2. reboot system, and press Tab to enter "grub" 3. edit "kernel" line and make sure no "psplash=false text" in grub cmdline 4. Press "enter" to boot system Note: The test is only for sato image.	
Expected Results: system should boot to runlevel 5.	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	installation&boot
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-610: g++ compile in sdk image
Summary: check if g++ can compile program in sdk image
Steps: 1. Boot up sdk image 2. check if g++ is built in 3. compile following program test.c "g++ test.c -o test -lm" 4. run "test" and check the output test.c: ########## #include <stdio.h> #include <math.h> double convert(long long l)

```
{
   return (double)l;    // or double(l)
}

int
main(int argc, char * argv[])
{
   long long l = 10;
   double f;

   f = convert(l);
   printf("convert: %lld => %f\n", l, f);

   f = 1234.67;
   printf("floorf(%f) = %f\n", f, floorf(f));
   return 0;
}
##########
```

Expected Results:

executable binary test can run without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-611: gcc compile in sdk image

Summary:

check if gcc can compile program in sdk image

Steps:

1. Boot up sdk image
2. check if gcc is built in
3. compile following program test.c "gcc test.c -o test -lm"
4. run "test" and check the output

test.c:

```
##########
#include <stdio.h>
#include <math.h>

double
convert(long long l)
{
   return (double)l;    // or double(l)
}

int
main(int argc, char * argv[])
{
   long long l = 10;
```

```
   double f;

   f = convert(l);
   printf("convert: %lld => %f\n", l, f);

   f = 1234.67;
   printf("floorf(%f) = %f\n", f, floorf(f));
   return 0;
}
##########
```

Expected Results:

executable binary test can run without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-614: run command make in sdk image

Summary:

check if command make can work in sdk image

Steps:

1. Boot up sdk image
2. check if make is built in
3. run command "make" with following makefile and build the test.c file from case "gcc compile in sdk image"

```
test: test.o
	gcc -o test test.o -lm
test.o: test.c
	gcc -c test.c
```

Expected Results:

make command can work without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-675: cvs project compile in sdk image

Summary: cvs project could be compiled in sdk image	
Steps: 1. Download cvs project from http://ftp.gnu.org/non-gnu/cvs/source/stable/1.11.23/cvs-1.11.23.tar.bz2 2. Copy cvs tarball into sdk image 3. Extract the tarball and do "configure", "make" and "make install"	
Expected Results: cvs project could be compiled successfully	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-676: iptables project compile in sdk image	
Summary: iptables project could be compiled in sdk image	
Steps: 1. Download iptables project from http://netfilter.org/projects/iptables/files/iptables-1.4.9.tar.bz2 2. Copy iptables tarball into sdk image 3. Extract the tarball and do "configure", "make" and "make install"	
Expected Results: iptables could be compiled successfully	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-677: sudoku-savant project compile in sdk image
Summary: sudoku-savant could be compiled in sdk image
Steps: 1. Download sudoku-savant project from http://downloads.sourceforge.net/project/sudoku-

savant/sudoku-savant/sudoku-savant-1.3/sudoku-savant-1.3.tar.bz2 2. Copy sudoku-savant tarball into sdk image 3. Extract the tarball and do "configure", "make"
Expected Results: sudoku-savant could be compiled successfully

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato-sdk, lsb-sdk

Test Case TC-622: perl program work in image
Summary: A perl program could be executed and output correctly in image
Steps: 1. Check if perl is installed in image and could run with "perl -v" 2. Prepare a perl program like followig test.pl 3. Run "perl test.pl" ######## $a = 9.01e+21 + 0.01 - 9.01e+21; print ("the value of a is ", $a, "\n"); $a = 9.01e+21 - 9.01e+21 + 0.01; print ("the value of a is ", $a, "\n"); ########
Expected Results: The test.pl could run without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Auto
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-615: shutdown system
Summary: verify that system can be shutdown by command
Steps:

1. boot system
2. launch terminal and run "shutdown -h now" or "poweroff"

Expected Results:

System can be shutdown successfully

Test Execution Cycle Type:	Sanity
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-613: reboot system

Summary:

verify that system can boot by command

Steps:

1. boot system
2. launch terminal and run "reboot"

Expected Results:

System can reboot successfully

Test Execution Cycle Type:	Sanity
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-605: adjust date and time

Summary:

adjust date and time

Steps:

1.launch terminal and run "date -R" to check current system time
2.adjust Date&Time by these commands:
For date command from coreutils, for example the sdk image use coreutils, you should use following syntax:
$ date -s "10:00:00 20100809"
$ date -R
$ Mon, 09 Aug 2010 10:00:00 +0000
For date command in busybox, for example the sato image use busybox, you should use following syntax:
$ date "080910002010"
$ date -R

$ Mon, 09 Aug 2010 10:00:00 +0000 3. check date with "date -R" and the time shown on matchbox-panel
Expected Results: System time should be adjust to what you specified

Test Execution Cycle Type:	Weekly
Case Automation Type:	Auto
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-616: switch among multi applications and desktop
Summary: switch among multi applications and desktop
Steps: 1. launch browser to read news (yahoo etc.) 2. launch terminal 3. switch among multi applications and desktop 4. close applications Note: The case is for sato image only.
Expected Results: 1. user could switch among multi applications and desktop

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-627: vncserver for target
Summary: Check if vncserver setup work in target and vnc client could connect it
Steps: 1. Check if x11vnc is installed in target 2. Run command "x11vnc -display :0.0", check the ip address of the target 3. On a client, run command "vncviewer $ip_address_of_target:0"
Expected Results: A virtual X desktop of target should be pop-up on the client

Test Execution Cycle Type:	Weekly

Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-609: file manager	
Summary: file manager	
Steps: 1.launch file manager from application panel 2.view folder/file in file manager 3.copy and paste folder/file in file manager Note: The test is only for sato image	
Expected Results: 1.folder and file could be listed in file browser with different display mode	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-617: system dmesg log check	
Summary: check if there is error in dmesg after system boot up	
Steps: 1. boot system and run command "dmesg"	
Expected Results: No error message in dmesg	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-618: usb mount	
Summary: verify that system can mount plugged usb automatically	
Steps: 1. boot system 2. plug usb stick	
Expected Results: 1. system notify that usb stick is accessible	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-619: usb read files	
Summary: verify that system can read files from usb	
Steps: 1. boot system 2. plug usb stick 3. view files in usb by file browser 4.copy some files from usb to local hardware	
Expected Results: 1. view/copy successfully	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-620: usb umount
Summary: verify that system can unmout usb automically
Steps: 1. boot system 2. plug usb stick 3. view files in usb by file browser

4.unplug usb	
Expected Results: 1. usb direcoty in file browser automatically missed	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-621: usb write files	
Summary: verify that system can write files to usb	
Steps: 1. boot system 2. plug usb stick 3. create files in usb 4.copy some files from local hardware to usb	
Expected Results: 1. create/copy successfully	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-645: file copy by scp	
Summary: check if file can be copied from remote machine to device by scp	
Steps: 1. check avahi is install and started 2. get system IP and try "scp file $IP:/home/root" from remote machine (file >= 500M for real HW, file>=5M for QEMU)	
Expected Results: File can be copied from remote machine to device by scp	
Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto

Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-642: connman launch after boot	
Summary: After system booted, the connmand daemon should be launched	
Steps: 1. boot system 2. "ps |grep connmand" 3. check if there is a thread named connmand in background	
Expected Results: There should be one thread named connmand in background	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-644: ethernet enabled in connman	
Summary: After system boot, ethernet can get IP address with connman	
Steps: 1. boot system with network cable plugged in 2. "ps |grep connmand" if connmand is started 3. "ifconfig" check ethernet could get IP address and ping the address from remote machine	
Expected Results: Ethernet interface can get IP via connman	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-646: only one connmand in background	
Summary: there should be no more than one connmand in background	
Steps: 1. boot system 2. "ps |grep connmand" 3. the connmand should be in background 4. run command "connmand" 5. check if the second connmand can be generated	
Expected Results: There will be only one connmand instance in background	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-647: remote access by ssh	
Summary: check if the device can be accessed remotely by ssh	
Steps: 1. check avahi is install and started 2. get system IP and try "ssh $IP" from remote machine	
Expected Results: it is ok to access system by ssh from remote machine	
Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk, lsb-sdk

Test Case TC-655: ethernet static ip set in connman
Summary: we could set static ip for ethernet in connman
Steps:

1. launch connman-properities

2. choose ethernet device and set static ip for it. For example, in our internal network, we can set as following:

ip address: 10.239.48.xxx

Broadcast: 10.239.48.255

Mask: 255.255.255.0

Expected Results:

we can set static ip for ethernet device

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

Test Case TC-656: ethernet get IP in connman via DHCP

Summary:

ethernet device can get IP in connman via DHCP

Steps:

1. Set static IP for ethernet device in connman
2. Check if ethernet device can work with static IP
3. Choose DHCP method for ethernet device
4. Check with ping if ethernet device get IP address via DHCP

Expected Results:

Ethernet device can get dynamic IP address via DHCP in connman

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

Test Case TC-643: connman offline mode in connman-gnome

Summary:

change offline mode in comman-gnome can make all connection off

Steps:

1. Launch connman-properties after system booting

2. choose "offline mode" and check the connection of all network interfaces

Expected Results:

All connection should be off after clicking "offline mode"

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	connectivity
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-631: X server can start up with runlevel 5 boot
Summary: check if X server can work well after system runlevel 5 booting
Steps: 1. boot up system with default runlevel
Expected Results: X server can start up well and desktop display has no problem

Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	graphics
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, beagleboard, crownbay, sugarbay, jasperforest
image profile:	sato, sato-sdk

Test Case TC-632: qt application quicky
Summary: quicky is a simple note-taking application with Wiki-style syntax and behaviour
Steps: launch quicky and write something in quicky
Expected Results: http://qt-apps.org/content/show.php/Quicky?content=80325

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	graphics
target:	e-menlow, blacksand, beagleboard, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

Test Case TC-661: standby	
Summary: system can enter standby and resume from standby	
Steps: 1. boot system and launch terminal; check output of "date" and launch script "continue.sh" 2. echo "mem" > /sys/power/state 3. After system go into S3 mode, move mouse or press any key to make it resume 4. Check "date" and script "continue.sh" 5. Check if application in X can work as normal continue.sh as below: ################## #!/bin/sh i=1 while [0] do echo $i sleep 1 i=$((i+1)) done ##################	
Expected Results: screen should resume back and script can run continuously	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

Test Case TC-662: Test if LAN device works well after resume from suspend state	
Summary: Test if LAN device works well after resume from suspend state.	
Steps: 1. boot system and launch terminal 2. echo "mem" > /sys/power/state 3. After system go into S3 mode, move mouse or press any key to make it resume 4. check ping status	
Expected Results: ping should always work before/after standby	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

Test Case TC-663: Test if usb hid device works well after resume from suspend state	
Summary: Test if usb hid device works well after resume from suspend state.	
Steps: 1. boot system and launch terminal 2. echo "mem" > /sys/power/state 3. After system go into S3 mode, move mouse or press any key to make it resume 4. check usb mouse and keyboard	
Expected Results: usb mouse and keyboard should work	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, crownbay, sugarbay, jasperforest
image profile:	sato-sdk

1.2 Test Suite : ADT

Test Case TC-628: gcc from ADT toolchain can build c program
Summary: gcc from ADT toolchain can build c program and run with qemu-${ARCH} command or in target image

Steps:
 	
1. Install toolchain tarball and setup cross compile environment
2. compile following program test.c "${CC} test.c -o test -cc -lm"
3. run "test" with qemu-${ARCH} or run it into corresponding target image and check the output
 	
Note: Currently, only i586_i586, x86-64_x86-64 and i586_$X(x is mips, arm and ppc) toolchain tarballs are covered in testing.
 	
#########
#include <stdio.h>
#include <math.h>
 	
double
convert(long long l)
{
  return (double)l; // or double(l)
}
 	
int
main(int argc, char * argv[])
{
  long long l = 10;
  double f;
 	
  f = convert(l);

```
  printf("convert: %lld => %f\n", l, f);

  f = 1234.67;
  printf("floorf(%f) = %f\n", f, floorf(f));
  return 0;
}
#########
```

Expected Results:

executable binary test can run without problem

Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	sdk
target:	build_system
image profile:	

Test Case TC-629: g++ from ADT toolchain can build c program

Summary:

g++ from ADT toolchain can build c program and run with qemu-${ARCH} command or in target image

Steps:

1. Install toolchain tarball and setup cross compile environment
2. compile following program test.c "${CXX} test.c -o test -cc++ -lm"
3. run "test" with qemu-${ARCH} or run it in corresponding target image and check the output

Note: Currently, only i586_i586, x86-64_x86-64 and i586_$X(x is mips, arm and ppc) toolchain tarballs are covered in testing.

```
#########
#include <stdio.h>
#include <math.h>

double
convert(long long l)
{
  return (double)l;    // or double(l)
}

int
main(int argc, char * argv[])
{
  long long l = 10;
  double f;

  f = convert(l);
  printf("convert: %lld => %f\n", l, f);

  f = 1234.67;
  printf("floorf(%f) = %f\n", f, floorf(f));
  return 0;
}
#########
```

Expected Results:

executable binary test can run without problem

Test Execution Cycle Type:	Sanity
Case Automation Type:	Auto
Case State:	Ready
Feature:	sdk
target:	build_system
image profile:	

Test Case TC-678: ADT toolchain could build cvs project
Summary: ADT toolchain could build cvs project
Steps: 1. Install toolchain tarball and setup cross compile environment 2. Download cvs project, http://ftp.gnu.org/non-gnu/cvs/source/stable/1.11.23/cvs-1.11.23.tar.bz2 3. With the cross compile environment, run "./configure ${CONFIGURE_FLAGS}", "make", "make install DESTDIR=/opt/tmp" Note: Currently, only i586_i586, x86-64_x86-64 and i586_$X(x is mips, arm and ppc) toolchain tarballs are covered in testing.
Expected Results: cvs project could be compiled successfully with ADT toolchain

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	build_system
image profile:	lsb-sdk

Test Case TC-679: ADT toolchain could build iptables project
Summary: iptables project could be compiled with ADT toolchain
Steps: 1. Install toolchain tarball and setup cross compile environment 2. Download iptables project, http://netfilter.org/projects/iptables/files/iptables-1.4.9.tar.bz2 3. With the cross compile environment, run "./configure ${CONFIGURE_FLAGS}", "make", "make install DESTDIR=/opt/tmp" Note: Currently, only i586_i586, x86-64_x86-64 and i586_$X(x is mips, arm and ppc) toolchain tarballs are covered in testing.
Expected Results: iptables could be compiled successfully

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready

Feature:	sdk
target:	build_system
image profile:	lsb-sdk

Test Case TC-680: ADT toolchain could build sudoku-savant project	
Summary: sudoku-savant could be compiled with ADT toolchain	
Steps: 1. Install toolchain tarball and setup cross compile environment 2. Download sudoku-savant project, http://downloads.sourceforge.net/project/sudoku-savant/sudoku-savant/sudoku-savant-1.3/sudoku-savant-1.3.tar.bz2 3. With the cross compile environment, run "./configure ${CONFIGURE_FLAGS}", "make", "make install DESTDIR=/opt/tmp" Note: Currently, only i586_i586, x86-64_x86-64 and i586_$X(x is mips, arm and ppc) toolchain tarballs are covered in testing.	
Expected Results: sudoku-savant could be compiled successfully	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	build_system
image profile:	lsb-sdk

Test Case TC-630: unfs support for qemu target	
Summary: Check if unfs works for qemu target	
Steps: 1. Prepare a *rootfs.tar.bz2 image 2. Prepare a folder under poky directory as <rootfs-dir>, for example poky/temp 3. Run command "runqemu-extract-sdk *rootfs.tar.bz2 poky/temp" 4. Run command "runqemu nfs <kernel> <rootfs-dir>"	
Expected Results: QEMU target should be started with unfs	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips
image profile:	sato, sato-sdk, lsb-sdk

1.3 Test Suite : Stress

Test Case TC-672: crashme for stress	
Summary: Run crashme in real hardware for stress testing	
Steps: 1. Get crashme from http://people.delphiforums.com/gjc/crashme.html 2. By following the setup steps on above URL, build crashme in target. 3. Run crashme for 24 hours	
Expected Results: target should not crash with the program	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	stress
target:	jasperforest
image profile:	lsb-sdk

Test Case TC-673: helltest for stress	
Summary: Run helltest for stress in target	
Steps: 1. helltest is stress test suite, which does compiler test for hours 2. We download the test suite and run it for 24 hours	
Expected Results: helltest should not make target crash	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	stress
target:	jasperforest
image profile:	lsb-sdk

1.4 Test Suite : Power/Performance

Test Case TC-657: boot time collection
Summary: To collect boot time of clean installation, from grub to full desktop
Steps:

1. Reboot testing device at least 3 times and do not plug anything while collecting boot time by stopwatcher:

#reboot

Expected Results:

Provide average boot time and dmesg log

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	performance
target:	crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-659: memory footprint

Summary:

collect data of the used/free memory

Steps:

With default installtion, launch terminal and type 'free' to read the used/free disk space

Expected Results:

Provide 'free' output

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	core
target:	crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-660: powertop log

Summary:

collect powertop data

Steps:

1. Run "powertop -d" and record output

2. Save the percentage of deepest C state(C3 or C2)

Expected Results:

Provide powertop output

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	core
target:	crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-681: Idle power consumption	
Summary: Collect idle power consumption of target system	
Steps: 1. Use power meter to collect ilde power consumption of target system for 10 minutes 2. Save it and compare it with old data	
Expected Results: There should be no regression between old and new ilde power data	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	performance
target:	crownbay, sugarbay
image profile:	sato-sdk

1.5 Test Suite : Graphics

Test Case TC-648: Graphics ABAT
Summary: Yocto on SugarBay should pass Intel graphics ABAT testing

Steps:
 	
1. Download ABAT test suite from internal git repository, git clone
git://tinderbox.sh.intel.com/git/abat
2. Apply following patch to make it work on yocto environment
3. Run "./abat.sh" to run ABAT test
 	
########
diff --git a/glxgears_check.sh b/glxgears_check.sh
index 17622b8..c4d3b97 100755
--- a/glxgears_check.sh
+++ b/glxgears_check.sh
@@ -31,7 +31,7 @@ else
 	
    sleep 6
 	
-    XPID=$(ps ax | awk '{print $1, $5}' | grep glxgears | awk '{print $1}')
+   XPID=$(ps | awk '{print $1, $5}' | grep glxgears | awk '{print $1}')
    if [! -z "$XPID"]; then
        kill -9 $XPID >/dev/null 2>&1
        echo "glxgears can run, PASS!"
diff --git a/x_close.sh b/x_close.sh
index e287be1..3429f1a 100755
--- a/x_close.sh
+++ b/x_close.sh
@@ -22,7 +22,7 @@
 #

```
 function close_proc(){
 echo "kill process Xorg"
-XPID=$(  ps ax | awk '{print $1, $5}' | egrep "X$|Xorg$" | awk '{print $1}')
+XPID=$(  ps | awk '{print $1, $6}' | egrep "X$|Xorg$" | awk '{print $1}')
 if [ ! -z "$XPID" ]; then
     kill $XPID
     sleep 4
diff --git a/x_start.sh b/x_start.sh
index 9cf6eab..2305796 100755
--- a/x_start.sh
+++ b/x_start.sh
@@ -24,7 +24,7 @@
 X_ERROR=0

 #test whether X has started
-PXID=$(ps ax |awk '{print $1,$5}' |egrep "Xorg$|X$" |grep -v grep | awk '{print $1}')
+PXID=$(ps |awk '{print $1,$6}' |egrep "Xorg$|X$" |grep -v grep | awk '{print $1}')
 if [ ! -z "$PXID" ]; then
     echo "[WARNING] Xorg has started!"
     XORG_STATUS="started"
@@ -35,9 +35,11 @@ else
     #start up the x server
     echo "Start up the X server for test in display $DISPLAY....."

-    $XORG_DIR/bin/X >/dev/null 2>&1 &
+    #$XORG_DIR/bin/X >/dev/null 2>&1 &
+    #sleep 8
+    #xterm &
+    /etc/init.d/xserver-nodm start &
     sleep 8
-    xterm &
 fi
     XLOG_FILE=/var/log/Xorg.0.log
     [ -f $XORG_DIR/var/log/Xorg.0.log ] && XLOG_FILE=$XORG_DIR/var/log/Xorg.0.log
@@ -54,7 +56,7 @@ fi
         X_ERROR=1
     fi

-    XPID=$( ps ax | awk '{print $1, $5}' | egrep "X$|Xorg$" |grep -v grep| awk '{print $1}')
+    XPID=$( ps | awk '{print $1, $6}' | egrep "X$|Xorg$" |grep -v grep| awk '{print $1}')
     if [ -z "$XPID" ]; then
         echo "Start up X server FAIL!"
  echo
########
```

Expected Results:

All ABAT test should pass

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	bsp
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato, sato-sdk

Test Case TC-649: openarena - 3D

Summary:

Run opernarena testing and compare the result with upstream graphics result

Steps:

1. Download and build openarena through phoronix test suite. first download a new phoronix from its website, then download the game in it. The openarena we use is v0.8.5.

####
phoronix-test-suite list-tests
phoronix-test-suite install openarena
####
2. Run the test suite with following command
####
vblank_mode=0 openarena +exec pts +set r_mode -1 +set r_fullscreen 1 +set r_customWidth $VIDEO_WIDTH +set r_customHeight $VIDEO_HEIGHT
####

The VIDEO_WIDTH and VIDEO_HEIGHT set the game's resolution，you can get current resolution by command "xrandr"

Expected Results:

Compare the result of Yocto with upstream graphics

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	bsp
target:	sugarbay
image profile:	sato, sato-sdk

Test Case TC-650: urbanterror - 3D

Summary:

Run urbanterror and compare the result of Yocto with upstream graphics

Steps:

1. download and build: This game also can get through phoronix-test-suite. 2. we should set some environments as following before test:
###
OS_TYPE=Linux
OS_ARCH=`uname -i`
LOG_FILE=${LOGNOW_DIR}/${LOG_FILE}
###
3. Run urbanterror with following command
###
vblank_mode=0 ./urbanterror +timedemo 1 +set demodone 'quit' +set demoloop1 'demo pts1; set nextdemo vstr demodone' +vstr demoloop1 +set r_customwidth $VIDEO_WIDTH +set r_customheight $VIDEO_HEIGHT
###

Expected Results:

Get the FPS data of Yocto and compare it with upstream graphics

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	bsp
target:	sugarbay
image profile:	sato, sato-sdk

Test Case TC-651: x11perf - 2D

Summary:

Get fps data of x11per running
Steps: 1. Run "x11perf -aa10text" and "x11perf -rgb10text" 2. Get the FPS result and compare it with upstream graphics data on Sandybridge
Expected Results: There should not be big regression between Yocto and upstream linux

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	bsp
target:	sugarbay
image profile:	sato, sato-sdk

1.6 Test Suite : Mulitimedia

Test Case TC-638: sound on/off
Summary: check if sound can be turned on/off
Steps: 1. copy amixer is installed 2. Run "amixer set Master on" to turn on audio device 3. Run "amixer set Master 64" to adjust to maxium volumn 4. Run "amixer set Speaker on" to turn on speaker 5. Run "amixer set Speaker 64" to adjust to maxium volumn 6. Run "amixer set Master off" to turn off audio device 7. Run "amixer set Speaker off" to turn off speaker
Expected Results: Above commands can run without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-633: audio play (mp3)
Summary: make sure music player cannot play mp3 format file
Steps:

1. copy sample mp3 file to system 2. launch music player and make sure it cannot play the mp3 file
Expected Results: mp3 file can not be played

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-634: audio play (ogg)
Summary: check if music player can play ogg format file
Steps: 1. copy sample ogg file to system 2. launch music player can play the ogg file
Expected Results: ogg file can be played without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-636: audio stop (ogg)
Summary: check if music player can play ogg format file
Steps: 1. copy sample ogg file to system 2. launch music player can play the ogg file 3. click "stop" button to stop playing 4. click "start" button to resume playing
Expected Results: ogg file can be start/stop without problem

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-635: audio play (wav)	
Summary: check if music player can play wav format file	
Steps: 1. copy sample wav file to system 2. launch music player can play the wav file	
Expected Results: wav file can be played without problem	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-637: audio stop (wav)	
Summary: check if music player can stop playing with wav format file	
Steps: 1. copy sample wav file to system 2. launch music player can play the wav file 3. click "stop" button to stop playing 4. click "start" button to resume playing	
Expected Results: wav file can be start/stop without problem	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-639: video play (mpeg)
Summary: make sure video player cannot play mpeg format file
Steps: 1. copy sample mpeg file to system 2. launch video player and make sure it cannot play the mpeg file
Expected Results: mpeg file cannot be played

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-640: video play (ogg)	
Summary: check if video player can play ogg format file	
Steps: 1. copy sample ogg file to system 2. launch video player can play the ogg file	
Expected Results: ogg file can be played without problem	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-641: video stop (ogg)	
Summary: check if video player can play ogg format file	
Steps: 1. copy sample ogg file to system 2. launch video player can play the ogg file 3. click "stop" button to stop playing 4. click "start" button to resume playing	
Expected Results: ogg file can be start/stop without problem	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	multi-media
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

1.7 Test Suite : Compliance

Test Case TC-653: LTP subset test suite

Summary:

LTP subset test suite

Steps:

For real hardware, run following component,
syscalls
fs
fsx
dio
io
mm
ipc
sched
math
nptl
pty
admin_tools
timers
commands

For QEMU, run following component
syscalls
mm
ipc
sched
math
nptl
pty
admin_tools
commands

Run Instructions:
LTP download: http://sourceforge.net/projects/ltp/files/LTP%20Source/ltp-20101031/ltp-full-20101031.bz2/download
build steps: refer to http://ltp.sourceforge.net

Run steps:
1. Build LTP with toolchain or in sdk image
2. For QEMU, create the qemu target with "-m 512", which makes some memory stress cases pass. For some issues, we could only set 128M for qemuarm and 256M for qemumips.
3. Copy LTP folder into target, for example, /opt/ltp. Modify script "runltp", remove test suites not to be tested
4. Comment runtests/sched: hackbench, which is not suitable to run in emulators
5. Prepare a tmp folder under your ltp folder, for example, create a tmp folder under your ltp folder, like /opt/ltp/tmp
6. ./runltp -p -l result-M2-20101218.log -C result-M2-20101218.fail -d /opt/ltp/tmp &> result-M2-20101218.fulllog
   (assume you mount your LTP disk at /opt and create your own tmp dir at /opt/ltp/tmp)

Expected Results:

Check the result on wiki, https://wiki.yoctoproject.org/wiki/LTP_result, there should be no regression failure met.

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Semi-Auto
Case State:	Ready
Feature:	core
target:	qemuarm, qemuppc, qemumips, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, sugarbay
image profile:	sato-sdk, lsb-sdk

Test Case TC-654: POSIX subset test suite	
Summary: Run subset test suite of POSIX test suite	
Steps: POSIX test suite download: http://sourceforge.net/projects/posixtest/files/posixtest/posixtestsuite-1.5.2/posixtestsuite-1.5.2.tar.gz/download build: refer to http://posixtest.sourceforge.net/ Run steps: 1. Get POSIX test suite as above 2. Start target and copy test suite into it 3. For qemu, option "-m 512" should be added 4. Make sure below is uncommented from LDFLAGS file: #-D_XOPEN_SOURCE=600 –lpthread –lrt –lm 5. Run following commands under POSIX test suite run_tests SIG run_tests SEM run_tests THR run_tests TMR run_tests MSG run_tests TPS run_tests MEM	
Expected Results: Compare the test result on wiki, https://wiki.yoctoproject.org/wiki/Posix_result, there should be no more regression failures met.	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Semi-Auto
Case State:	Ready
Feature:	core
target:	qemuarm, qemuppc, qemumips, blacksand, beagleboard, mpc8315e-rdb, routerstationpro, sugarbay
image profile:	sato-sdk, lsb-sdk

Test Case TC-652: LSB subset test suite	
Summary: Run LSB subset test suite in target	
Steps: 1. Get LSB image and start the image(if it is QEMU) with option "-m 512M" 2. Get the LSB test suite or run script creat-lsb-image under poky source directory "scripts/creat-lsb-image" 3. Setup environment for lsb image in target with script LSB_Setup.sh, it could be found under poky source directory "/meta/recipes-extended/lsb/lsbsetup/LSB_Setup.sh" 4. Select LSB test items in LSB web interface and run them	
Expected Results: Check the result on wiki, https://wiki.pokylinux.org/wiki/index.php?title=LSB_result&action=edit&redlink=1. No regression failures should be met.	
Test Execution Cycle Type:	Fullpass

Case Automation Type:	Manual
Case State:	Ready
Feature:	core
target:	qemuarm, qemuppc, blacksand, mpc8315e-rdb, sugarbay
image profile:	lsb-sdk

1.8 Test Suite : Core Build System

Test Case TC-664: kernel interactive targets	
Summary: Check if yocto can support kernel interactive target build	
Steps: 1. download yocto source tree 2. prepare yocto build environment 3. Run "bitbake linux-yocto -c menuconfig" 4. Check if a new bash terminal pop up and menuconfig can be triggered	
Expected Results: menuconfig for kernel can be triggered with yocto build command	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-665: KVM enabled with qemu	
Summary: qemu can be started with KVM enabled	
Steps: 1. build a kernel with KVM enabled 2. Start qemu with option "kvm" with runqemu 3. Check if qemu starts up and if kvm_intel is used 4. If kvm_intel is not used when starting qemu, it will shows 0 in "Used by" column when you run "lsmod | grep kvm_intel"	
Expected Results: KVM enabled with qemu	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky

target:	build_system
image profile:	

Test Case TC-666: non-GPLv3 build check

Summary:

Check if non-GPLv3 build could pass and it does not has any GPLv3 packages installed

Steps:

1. Set following sentences in local.conf to GPLv3
#####
INCOMPATIBLE_LICENSE = "GPLv3"
#####
2. Build core-image-minimal and core-image-basic
3. Start up target after build is finished
4. Run following script to check if any GPLv3 packages installed

```
###################
#!/bin/sh

temp=`mktemp`
rpm -qa > $temp
ret=0

for i in `cat $temp`
do
	rpm -qi $i | grep License | grep -i gplv3 > /dev/null 2>&1
	if [ $? -eq 0 ]; then
		license=`rpm -qi $i | grep License | awk -F"License:" '{print
$2}'`
		echo "package $i has inconsistent license: $license"
		ret=1
	fi
done

rm -rf $temp
exit $ret
####################
```

Expected Results:

non-GPLv3 build pass and no GPLv3 packages installed in the image

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-667: yocto build in Fedora 14

Summary:

Build latest yocto in x86_64 Fedora 14 host

Steps:

1. By following the yocto handbook, download latest yocto source

2. Build core-image-minimal on Fedora 14	
Expected Results: Yocto build should pass on Fedora 14	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-669: yocto build in OpenSuse 11.4	
Summary: Build latest yocto in x86_64 OpenSuse 11.4	
Steps: 1. By following the yocto handbook, download latest yocto source 2. Build core-image-minimal on OpenSuse 11.4	
Expected Results: Build should pass on OpenSuse 11.3	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-670: yocto build in Ubuntu 11.04	
Summary: Build latest yocto in x86_64 Ubuntu 11.04	
Steps: 1. By following the yocto handbook, download latest yocto source 2. Build core-image-minimal on Utuntu 11.04	
Expected Results: Yocto build should pass on Utuntu 10.04	
Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-668: yocto build in KVM

Summary: Build yocto in KVM should work
Steps: 1. Setup a VM environment with KVM enabled, for example, RHEL6 2. Prepare a VM for yocto build testing, for example, OpenSuse 11.3 3. By following the yocto handbook, download latest yocto source into the VM 4. Build core-image-minimal in the VM
Expected Results: Yocto build in VM should work same as in real host

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

Test Case TC-671: sstate work on local host
Summary: Check if sstate could work with local cache
Steps: 1. Follow the wiki steps to setup a sstate cache on local machine, https://wiki.yoctoproject.org/wiki/Enable_sstate_cache 2. Prepare another yocto source directory and set the SSTATE_DIR the cache you setup in step 1) 3. Run poky build, for example, "bitbake core-image-minimal". You should note following things if sstate works: ######## NOTE: Preparing runqueue NOTE: Executing SetScene Tasks NOTE: Running setscene task 118 of 155 (virtual:native:/home/lulianhao/poky-build/edwin/poky/meta/recipes-devtools/pseudo/pseudo_git.bb:do_populate_sysroot_setscene) NOTE: Running setscene task 119 of 155 (/home/lulianhao/poky-build/edwin/poky/meta/recipes-devtools/quilt/quilt-native_0.48.bb:do_populate_sysroot_setscene ########
Expected Results: sstate should work and reduce build time

Test Execution Cycle Type:	Fullpass
Case Automation Type:	Manual
Case State:	Ready
Feature:	poky
target:	build_system
image profile:	

1.9 Test Suite : Regression

Regression test cases from bugzilla

Test Case TC-682: disk space check	
Summary: There should be enough disk space for QEMU rootfs	
Steps: 1. Launch QEMU targets(with rootfs.ext3 file) 2. Check the output of command df 3. If there is less than 5M disk space available, we assume it a failure	
Expected Results: There should be enough disk space for QEMU targets	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips
image profile:	sato, sato-sdk

Test Case TC-683: click terminal icon on X desktop	
Summary: terminal icon should work without problem on X desktop	
Steps: 1. After system launch and X start up, click terminal icon on desktop 2. Check if only one terminal window launched and no other problem met	
Expected Results: there should be no problem after launching terminal	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay
image profile:	sato, sato-sdk

Test Case TC-684: Add multiple files in music player
Summary: music player should be no problem when adding multiple files at same time
Steps: 1. Launch music player 2. Add multiple files(5 files) in music player at same time

Expected Results: music player should be OK with this action

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	e-menlow, blacksand, crownbay, sugarbay
image profile:	sato-sdk

Test Case TC-685: system shutdown with UNFS
Summary: system shutdown with UNFS should work
Steps: 1. Use UNFS to start QEMU targets 2. Run shutdown in QEMU targets
Expected Results: QEMU shutdown with UNFS should work

Test Execution Cycle Type:	BAT
Case Automation Type:	Manual
Case State:	Ready
Feature:	sdk
target:	qemux86_32, qemux86_64, qemuarm, qemuppc, qemumips
image profile:	sato, sato-sdk

Test Case TC-686: no connman-gnome icon on desktop
Summary: there should be no connman-gnome icon on desktop
Steps: 1. Launch sato image 2. There should be no connman-gnome icon on desktop, and connman-properties should be only invoked by toolbar
Expected Results: There should be no connman-gnome icon on desktop, and connman-properties should be only invoked by toolbar

Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay
image profile:	sato, sato-sdk

Test Case TC-687: application contact should work	
Summary: application contact should work without problem	
Steps: 1. Start up Sato image 2. Check if there is "contact" icon on desktop and run it 3. Check if there is any error by checking the output of this action and dmesg log	
Expected Results: "contact" launch should not cause any error	
Test Execution Cycle Type:	Weekly
Case Automation Type:	Manual
Case State:	Ready
Feature:	system usage
target:	qemux86_32, qemux86_64, qemuarm, qemumips, e-menlow, blacksand, mpc8315e-rdb, routerstationpro, crownbay, sugarbay
image profile:	sato, sato-sdk