**MPSS:( (Modem peripiral sub subsystem)**

**MPSS I&T** team is responsible for sofware integration of modem subsystems/components, sanity testing, intial triage of issues found during testing and relesae of MPSS images/CRM buils on all modems/targets

**As part of MPSS I & T Team Responisibilities:**

* MPSS Integration and Testing activities for NI, DI PL/CPLs based on Multiple Targets including MDM9x15, MSM8960, MSM8930, MDM9x25, MSM8974, MSM8926
* PL, CPL bringups
* ARIS/Continuous Integration
* Follow-up on the Tech labels submitted in ARIS and Integrate them
* Follow-up on all Internal and Strip/Pack build failures
* Follow-up on all RoboCPL build failures
* Moving CRs to BUILT state
* Continuous Test automation
* Ownership of CIT/MTF stations
* Ensure Basic modem sanity is done for all the build releases
* Root cause analysis of the Test case failures (QXDM Log analysis, Debug setup issues etc)
* Tracking and resolution of JIRA tickets for test case failures
* Follow-up of critical CR fixes for issues (functional or stability) identified during CIT/MTF testing.
* MPSS build releases to Target
* Customer Support with
* SBA/Delta releases, Test SBA/Service Task releases

**Meta Work flow:**

**1. Customer**

**2. TAM (Technical assesment manager)**

**3. Meta**

**4. Mpss (Modem peripiral sub subsystem)**

**CR States:**

* **Open:** New CR
* **Analysis:** add / approve In this two states are there :1)TBD(Tobedone) 2)yes

If CR is Approvel the state becomes yes if not apporvel then the CRstate become TBD

* **FIX:**Tech team fix the CR
* **Ready:**CR is in Ready state the **vu release** team should be fix the cr in ready state
* **Built:MPSS**

**Setup (How to load build into device):**

**1. JTAG** (Join test action group): load build manully

**2. ATM (Automated test manager)**

**MPSS Intigration Tools:**

* **Qpst**(Qualcomm product supproting tool)

Build id, port number, device information, and also get the dump logs from Sahara folder

* **Qxdm**(Qualcomm extensible diagnostic monitor)

Writing NV items manually check the status of the device

Calls status

* **Qmitestpro**(Quallcom machine inteface)

We can connect data calls, Network, voice calls and files

Used for connectivity test cases

* **Mars**

We can check cpl/pl status (status is active or inactive)

List of CR’s between build

* **Atm**(Automated test manager)

Monitor jobs

See results of a job

We can see queue jobs

We can abort jobs

We can pass upcoming

* **Tiberium**

To get the build details

To trigger the build manually with custom changes

* **ARIS**

ARIS is a tool which  is used for  triggering a build.

Using RUN ASSESSMENT we can trigger a build for a corresponding PL

Before ggoing to Run we have to check  statesforpickingVUforcorrespondingVUdifferent States are there .. they are

Inintegration(wher build is in progress)

Integrated(wher build is passed)

Submitted(VU is submited + ready )

Defined

Resubmitted

Rejected.(when build fail)

* **EC**(Electric connector)

To view the work flow of the triggerd build (view total work flow of the build)

We can also create device assemnts (which job shold be go and run in which station)

* **Releasenote**

We can generate release note and wiki for that corresponding build

* **Findit ()**

To find the build path and build details with Pl’s and state of the build

* **CR-checker**

To get the CR details of a particular build whether the fix already present are not and what state the CR is for the CPL

* **Crashscope**

We can raise a stability **jira** using crash scope which will have all the dumps details which caused the device to crash

**Build id**: MPSS.D1.4.0.1.c1\_0089\_M8926\_DAAAANAAM-1

**Pl/CPl** **: MPSS.D1.4.0.1.c1**

**Version** **: 0089**

**Client/flavour**: **M8926\_DAAAANAAM**

**Spin**  : **1**

Four types of builds are ther:

1. Internal build

2. External build

3. Debug distribution build

4. Mirror build

**1. Internal build**

In inernal build the client/flavour last letters shoul be like this: "NAAM"

**2. External build**

External build also called as strip/pack build

In external build the client/flavour last letters shoul be like this: "NAZM

**3. Debug build**

In debug build the client/flavour last letters shoul be like this: "NADM"

**4. Mirror build**

If you have one build in another location (SD) the same build will be created in Qips (hyd)

Client/flavour is like this NAAM-H

In this spin values also changed in SD spin value is 1 after crreating same build in qips the spin value should be 2

**MSM:**

In MSM is a user interface and only the mobile related information is tested, eg: calls, wifi, bluetooth, audio, video

**MDM:**

In MDM not the userinteface only the mobile data related informatoin is tesed, eg: router, switches

**Fusion4:**

MSM+Apps

**Modems:**

**1. Ni -Nikel: (old)**

**Chips:** 8930, 8960, fusion3: (MSM)

9x15: **MDM**

**2. Tr-Triton:**

**Chips**: 8626, 8610 :( MSM)

**3. TL-Thor:**

**Chips:** 9x45(**lte+MDM**)

9x40**(MDM**)

8996(**MSM)**

**4. DI-Dime:**

**Chips:** 8926(**MSM)**

8974(**MSM)**

9625(**MDM**)

**5. DPM (Dime plus minus)**

**6. BO -Bolt**

**7. JO- Jolokia**

**8. FE -Ferro (latest)**

**9. TA -Tabasco (latest)**

**Chips:** 8953

**10. AT -ATLAS**

**META Build haves 7 components:**

**1. Apps**

**2. Boot**

**3. LAPSS**

**4. MPSS (Modem periperal subsytem)**

**5. RPM**

**6. TZ (TrustZone)**

**7. WCNSS**

**ROOT Directory Files:**

* PL-id **:** pl-id inforamiton( product line)
* Syncsource **:** base build file with an error file and then submit file for that build file with out an error file
* Target\_components.plf(perforce list in file) **:** component Cpl
* Target\_main.plf(perforce list in file) **:** main pl
* Verifysrc(logfile) **:**  if the built failure with bl-cr then it shoul be ther
* Build id.brfx **:** it contains Build information like build execution time ,location,spin
* Log(total compilation file)
* Csfiles(cspecs) **:** source location \_ destination location

**PERFORCE:** Perforce applications are used to transfer files between the file repositories

And individual users’ workstations (work space)

**How to trigger a build:**

(First check the V.U status by using Aris you can check)

**Step 1:** Go to **Aris tool** and next go to you can view **Software image** in this two options are ther one is **create** and another one is **view**

**Create** for submit the build to target team

**View** for trigger a build

**Step 2:** copy the build PL/CPl into Aris Next click filter

**Step** 3: Go to software Image Right click that one after go to Runassesment

**Setp 4**: you get the work flow diagram

**Step 5**: submit that workflow (URL) to the target team