Building PLATO from Source

To perform this procedure, you must be working on a relatively unmodified version of CYBIS(PLATO) via PTERM and all systems must be up and running properly before it can be properly executed.

Overview and Warnings

Assumptions

- It is expected that the implementer has sufficient understanding of
  - dtCyber Operation
  - NOS2 Commands and Syntax
- The running version of dtCyber is at least 5.7 or later.
- The running version of PTerm is at least version 6.0.4 or later.
- While it is preferable to build from the latest “ready to run” distributions, you may wish to apply these modifications to a running CYBIS Release 1 system. If that is the case, this tape of modifications will be MANUALLY APPLIED during this procedure. The most critical files from this tape contain some required system modifications and tools.

Building PLATO from sources is a little complicated, but having a running PLATO system is essential because some of the configuration information must be taken from within PLATO code files:

- **PLMODS** Contains “persistent” modifications to *platopl* during build
- **TPMODS** Contains “temporary” modifications to *platopl* during build
- **GOGOMODS** Contains modsets used all assemblies, except the code-sharing full assemblies, and not applied permanently to *platopl*.

INFO History

The modsets in this file are active during gogo only; they are partialled out before the codesharing assemblies are run. The programmers must reactivate their modsets each week after codesharing is finished.

- **CYBMODS** Contains modsets used in all assemblies and applied permanently to *cybopl* each week. These mods should only contain cybil code.

It may be more educational to perform many of these steps manually because there is an opportunity to learn more about the process when performing these steps interactively.

IMPORTANT NOTE

- This procedure assumes that you are working from a “ready to run” version of the NCC NOS/PLATO installation.
You must be running from the deadstart configuration you intend to use as the foundation for building the NEW deadstart configuration.

• BACK UP everything - at least you'll have something to fall-back on if anything gets destroyed

You Have Been Warned

Correct the NET Config

During the NCC automated installation, a tape CYBIS-BINS-Development.tap in PFDUMP format which is used to build and pre-load ALL of the procedures and files necessary to run a DEVELOPMENT version of PLATO.

• As installed, NDL (cybis-ndl.job) needs to be modified to correct a problem with PNI errors. To correct this, open the file located in opt and locate the input block. Add the parameters ,ABL=7, DBL=5 to the TERMDEV statement as shown below:

```plaintext
CYBIS
*IDENT CYBIS
*DECK NETCFG
*D 85,86
LI02P: GROUP, PORT=14, L9600, NI=16.
TE02P: TERMDEV, TASX364, ABL=7, DBL=5.
*I 196
CYBUSER: DEFINE, MFAM=CYBER, MUSER=SYS, MAPPL=CYBIS.
*D 220,221
TE02P14: USER, CYBUSER.
TE02P15: USER, CYBUSER.
TE02P16: USER, CYBUSER.
TE02P17: USER, CYBUSER.
TE02P18: USER, CYBUSER.
TE02P19: USER, CYBUSER.
TE02P1A: USER, CYBUSER.
TE02P1B: USER, CYBUSER.
TE02P1C: USER, CYBUSER.
TE02P1D: USER, CYBUSER.
TE02P1E: USER, CYBUSER.
TE02P1F: USER, CYBUSER.
TE02P20: USER, CYBUSER.
TE02P21: USER, CYBUSER.
TE02P22: USER, CYBUSER.
TE02P23: USER, CYBUSER.
```

Re-Submit the deck at the dtCyber console. You will need to completely restart NAM once this job is complete.
Extract Current Configuration

Extracting the current configuration files from the current Deadstart tape and maintaining them in one place will make creation of deadstart tapes much easier in the future.

Safety First

This procedure assumes that the distribution of baseline masterfiles and configuration parameters have been untouched, but keeping a current copy of them in a PLATO file dstmods will make the merge process much easier when upgrading in the future.

1. On PLATO
   1. Add file dstmods to account system as a code file with 2 parts of space.
   2. create three NORMAL CODE BLOCKS in code file dstmods containing only a single character (asterisk is fine):
      1. eqpdeck
      2. config
      3. mfnx

2. On the NOS CONSOLE, enter the following commands:

   X.DIS. SUI,377777 GTR(SYSTEM,EPD)TEXT/EQPD01
   GTR(SYSTEM,CONFIG)TEXT/CONFIG GTR(SYSTEM,MFNX)PROC/MFNX
   PF(W,PB,EPD,DSMODS,EPD,S)
   PF(W,PB,CONFIG,DSMODS,CONFIG,S) PF(W,PB,MFNX,DSMODS,MFNX,S) DROP.

   1. The steps above
      1. obtain the current system file
      2. retrieves the three decks from the current system file
         1. EQPD01 the Equipment Deck
         2. CONFIG the PLATO Configuration file
         3. MFNX the PLATO Masterfile Configuration file
      3. writes the three decks from local files to PLATO files
   2. ends the DIS session

Load the Configuration Files from Tape

On the NOS CONSOLE, enter the following commands:

   X.DIS. USER(PLATO,PLATO) RESOURC(GE=4) LABEL(TAPE,VSN=CYBL1,PO=RF) on the dtCyber Console - load the tape example: load_tape 13,0,1,r,oprttapes\CYBIS-lessons.tap REWIND(TAPE) SKIPF(TAPE). Skip past the prefixed procedures "install" and "update" SKIPR(TAPE,11). This positions the tape at the file "wsbatch" COMMENT. WNS BATCH FILE FOR PLATO STUFF
   PF(W,BIN,TAPE,Z)-REW,LEDIT,ABTDT/WSBATCH COMMENT. PLMODS FOR THIS RELEASE
   PF(W,BIN,TAPE,Z)-REW,LEDIT,ABTDT/PLMODS COMMENT. CYBMODS FOR THIS RELEASE
   PF(W,BIN,TAPE,Z)-REW,LEDIT,ABTDT/CYBMODS COMMENT. **** INSTALL COMPLETE UNLOAD(TAPE) DROP.
Verify that **PLMODS**, **WSBATCH**, **CYBMODS** all are correct by viewing them through PLATO.

**Partialed Blocks**

All of the necessary mods have been partialed-in EXCEPT those found in block devuol.

Block devuol should be partialed

- OUT for **Production** systems.
- IN for **Development** system.

**Build PLATO**

At this point, all of the necessary setup is complete and it is now possible to build PLATO.

**Idle Throttling**

Make sure the idle throttling is OFF while performing this build.

```
X.DIS. USER(PLATO,PLATO) GET,XSES. XSES. . (Begins AUTO MODE) GET,BALL871. BALL871. .
(Opens AUTO MODE)
```

Automatic processing will continue ...

... When it completes:

Examine all files and pay particular attention to the file ERRS. If the file size is 0, then the procedure was completed successfully.

If ERRS is not empty, review its contents to make sure that you may proceed safely.

```
REWIND(ERRS) COPYSBF(ERRS,OUTPUT) ROUTE,OUTPUT.
```

Retrieve the file from the printer and examine it to ensure that no critical errors occurred.

If all is well, save the listing file “FULL” to “LISTF”

```
DEFINE(LISTF) REWIND(FULL) COPYEI(FULL,LISTF) RETURN(LISTF) DROP.
```

**Make Product Distribution Tapes**

Product Distribution tapes are made so they may be used to reload the necessary binaries if needed in the future.
CONFIG and MFNX

If you have made customizations to the PLATO pack layout, you must note that the distribution files contain the default CONFIG and MFNX files which do not contain any locally created masterfiles.

Start a new DIS session to make the new tape files:

```
X.DIS. USER(PLATO,PLATO) GET,MKPDIST. MKPDIST. . (Begins AUTO MODE)
```

When the procedure is complete, you may mount a new empty tape (this procedure assumes that the tape is to be loaded on tape drive 13,0,1 / Unit 51).

**Load a **RELEASE** Tape**

*If you are building DEVELOPMENT Release tape, skip to the next section.*

Load a tape on dtCyber:

```
lt 13,0,1,w,tapes/20221123_0900.PLATOREL.tap
```

LABEL(TAPE,VSN=CYBREL,PO=UW,W) * (switch out of DIS to issue the VSN command)
VSN,051,CYBREL.

**Load a **DEVELOPMENT** Tape**

Load a tape on dtCyber:

```
lt 13,0,1,w,tapes/20221123_0900.PLATODEV.tap
```

LABEL(TAPE,VSN=CYBDEV,PO=UW,W) * (switch out of DIS to issue the VSN command)
VSN,051,CYBDEV.

**Make Distribution Tape**

```
REWIND(PDIST) COPYBF(PDIST,TAPE) UNLOAD(TAPE)
```

Dismount the tape from dtCyber:

```
unload_tape 13,0,1
```
Make New Deadstart Tape

Creation of a new Deadstart tape is a CONTINUATION of the previous step.

Mount a fresh tape to receive the deadstart file:

On dtCyber:

```
lt 13,0,1,w,deadstart\20221123_0900.prel.ds.tap
```

Mount the CURRENT Deadstart Tape

```
LABEL(ODS,VSN=***050,LB=KU,F=I,PO=UR) REWIND(PDIST) COPYBF(PDIST,LGO).
LIBEDIT(P=ODS,B=LGO,N=NEWDS) ROUTE,OUTPUT.
```

Examine the LIBEDIT output to ensure that everything looks correct.

Then write NEWDS to the new deadstart tape:

```
REWIND(NEWDS) ITEMIZE,NEWDS. ROUTE,OUTPUT. REWIND(NEWDS). ASSIGN,051,NDST.
COPYBF(NEWDS,NDST). UNLOAD(NDST). DROP.
```

At this point, the new deadstart tape should be ready to run.

Edit the new deadstart tape name into cyber.ini and restart dtCyber/NOS.

Verify that everything works properly.

Apply Local Changes

Local modifications

If your configuration deviated from the baseline configuration, a new deadstart tape will need to be made from the files saved earlier in this procedure.

On dtCyber load up a new empty deadstart tape:

```
lt 13,0,1,w,deadstart\20221123_0900.prel2.ds.tap
```

We retrieve the previously saved configurations from the running system:

```
X.DIS. SUI,377777. COMMON(SYSTEM) PF(PB,CONFIG,DSTMODS,CONFIG,S) PF(PB,MFNX,DSTMODS,MFNX,S) UNLOAD(DIR) NOTE(DIR,NR).*FILE CONFIG NOTE(DIR,NR).*FILE MFNX PACK(DIR) REWIND(*) LIBEDIT(P=SYSTEM,I=DIR) REWIND(*) ASSIGN,51,T,F=I. COPY(NEW,T,V)
```
DROP.

Edit the new deadstart tape into cyber.ini

Shutdown everything and deadstart with the new tape.

All masterfiles should be in the right place!