| | |
|---|---|
| **Document Title** | **CMtest Programmers Guide** |
| **Products Affected** | All product manufactured by COMPANY |
| **Document Part Number** | TST-00002 |
| **Revision and Date** | 01 05/22/17 |
| **Status** | Preliminary |
| **Author** | Joe White |

**This is an uncontrolled copy when in printed form.**

| | REQUIRED APPROVAL | SIGNATURE | DATE |
|---|---|---|---|
| ☒ | **TEST ENGINEERING** | **Joe White** | **05/22/17** |
| ☐ | **MANUFACTURING ENGINEERING** | | |
| ☐ | **NEW PRODUCT INTRODUCTION** | | |

# REVISION HISTORY

| REVISION | REVISION DESCRIPTION | ECO # | DATE | ORIGINATOR |
|---|---|---|---|---|
| A | Initial release Adapted from ver 1.4 release from Paul Tindle. | | 418/07 | Joe White |
| 1 | Made More generic | | 05/22/17 | Joe White |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# TABLE OF CONTENTS



# 1.0 Introduction

This procedure defines the basic coding, structure, archive and distribution features of CMTEST. Portions of this procedure are taken from Paul Tindles CMTEST installation guide and CMTEST users guide. Paul Tindle cane be reached at Paul@Tindle.org.

Portions of this procedure contain specific references to locations and setup. These are left in place as examples

## 1.1 Scope

This procedure applies to the following departments:

- COMPANY Test Engineering
- COMPANY subcontractors.

This document is divided in the following section:

- Hardware Requirements
- Software Requirements
- Procedure
- Appendix

## 1.2 References

- Guide to RedHat Linux or equivalne
- COMPANY CMtest Installation Guide
- COMPANY CMtest Programmers Guide

# 2.0 Hardware Requirements

The following hardware items are needed to perform this procedure:

- PC with minimum 1 serial port with Fedora Core 3 ore equivalent installed
- CMTEST installed and configured
- Network requirements met
- Test station Hardware requirements met.

# 3.0 Software Requirements

- CMtest installed
- Recommended: Optiperl or equivalent IDE for test development.
- PERL Experience

# 4.0 Procedure

## 4.1 CMTEST directory structure

1) Directory Structure – Development(what gets checked out of the “cmtest” CVS repository)

```
Cmtest
        | - Bin
        | - Cfgfiles
        | - Cmdfiles
        | - Java
        | - Lib
        | - Tftpboot
                | - OS_CODE
                | - Diag_Code
        | - Uutcfg
        | - www
                | - cgi-bin
                        | - lib
                | - html
```

2) Directory Structure – Server

```
/a
        |- Cmtest ( l/var/local/cmtest linked to here)
                | - cfgfiles – Not used
                | - dist
                        | - bin
                        | - cfgfiles
                        | - cmdfiles
                        | - java
                        | - lib
                        | - uutcfg
```

```
                        | - www
                | - logs
                        | - logfiles
                | - stats
                | - tmp
        | - ftp
        | - tftpboot
/var/www
        | - cgi-bin
        | - html
/a/tftpboot(/tftpboot linked here)
        | - OS_CODE
        | - Diag_Code
/a/ftp
```

3) Directory Structure –Client

```
/usr/local/cmtest
        | - a
        | - b
        | - c
        | - d
        | - e

        | - Bin
        | - Cfgfiles
        | - Cmdfiles
        | - Java
        | - Lib
        | - Tftpboot
                | - OS_CODE
                | - Diag_Code
        | - Uutcfg
        | - www
                | - cgi-bin
                        | - lib
                | - html
/usr/home/mfg ( assuming a “mfg” user has been created)
/var/local/cmtest ( mounted/linked to server at /var/local/cmtest)
```

## 4.2 Cmtest code base

Majority of CMTEST is written in Perl using Perl-expect, perl-tk extensions for the text send/expect. Perl is available to the developer, but the majority of the coding/scripting is done in custom command files. In-addition COMPANY uses

some parallel port extensions for the option of using the parallel port to control power supplies.

## 4.3 Directory/File Descriptions

Below are some short descriptions of how each directory is used. There are repetitions from the Development area to the /var/local/cmtest to the usr/local/cmtest areas

### 4.3.1 Development Areas

1) bin – bin area contains all of the Perl executables. Below is a short description for each, local help can be found at each executable:

abort.pl - abort a session that has terminated, but is still tagged as running

chkrel.pl - runs a showdiff on files in push.cfg

cmtest.pl – main test execution script. This is was operators will run to start a test and bring up a menu

console.pl – Used to interact with terminals, not in regular use by COMPANY

fixdevs.pl – Used to change device permissions, may be used in a cron job to fix pesky devices that change permissions automatically

logevent.pl – used to manual add to the event log files, not used by COMPANY

mkhost.pl – main script used when building new servers and controls with the CMTEST environment

pc.pl – cmtest even long SN/TID process checker, not used by COMPANY

power.pl – a temporary power control script, not in use by COMPANY

powerswitch.pl – script to manually control APC switches

push.pl – main script to update site servers master distribution area( /var/local/cmtest). For COMPANY this also updates the tftpboot files. Web status files are also updated with the correct command line options

Release.id – Hold the descriptions of the current release. Also determines the release “pipe” to be used when updating the master image to the local/client areas(/usr/local/cmtest/x)

Settestenv.pl – Used to move the release pointer for end users on the clients(pointer to (/usr/local/cmtest/x) where x=a,b,c,d,or e)

Test.pl – a temporary test script not used by COMPANY

Uid.pl – used to modify/add user ID’s prompted for at test time in cmtest.pl. these are intern used to control which tests are displayed/allowed

Update.pl – main script to move the master image(/var/local/cmtest) to the release bucket at (pointer to (/usr/local/cmtest/x) where x=a,b,c,d,or e)

Yield.pl – script to extract yield information from the sites logfiles(only on the site server at /var/local/cmtest/logs). This also has the hooks for implementing Flow control

2) – cfgfiles – contains release configuration information, user environment configs and Yield configs.

.aliases,.bashrc,.cmtestrc, bash.rc – CMTEST user environment configs. These are in installed by mkhost.pl.

Erc_Defa.txt – Hash with Error Code to text message translations, Not fully implemented by COMPANY

Parts.cfg – depreciated, this file has been moved to the uutcfg directory.

Push.cfg – used by Update.pl,push.pl,mkhost.pl to decide what files are moved and to which bucket

Testctrl..default.cfg – default testctrl.cfg file installed by Mmkhost.pl

Users.cfg – Hash with encrypted userID and access level

Vssver.scc – not used by COMPANY. Use unclear.

X.cfg – X windows user config, Not used by COMPANY, use Unclear.

Yield_defaults.cfg – Default Yileld configuration.

Yield_defs.cfg – COMPANY Yield definitions.

3) – cmdfiles – Contains the COMPANY test and expect commands. These commands are interpreted and acted on by the CMTEST environment. Below are the top level (*.dat) files. The remaining *.inc files are nested include files used by the command files. A few modules in this directory are duplicated in the WWW directory structure. These are not guaranteed to be identical files at this time.

Bench.dat – COMPANY bench level tests, these are tests that do not require a chassis, Currently on a power and a slot ID is necessary, may be used with the Single slot chassis in the future.

Bench.dat – Basic board bring up to program card types and serial numbers. This test assumes firmware is programmed to a sufficient level by preprogramming at Procomm driven test programming station or preprogrammed components.

Bench_GLC_loop.dat – Not used, To be deleted

Chassis_BI.dat – 5 Slot chassis burin script. Supports 1 or 2 IMCs all slots must be filled, 24 Hour test

Chassis_Extended.dat – 5 Slot Chassis Extended test, Similar to BI script, but includes power cycling and run the external Adtech traffic generator. External Traffic is only support at COMPANY at this time, but is intended for Sample testing at Contract Manufacture during Production testing. 72 Hour Test

Chassis_ORT.dat – Similar to BI, but runs for 7 days

Chassis_Pre_BI.dat – Test in a full chassis, Must have 1 or 2 IMCs and a full chassis. Boards are updated to correct software release for BI testing. All software/firmware is programmed to shipping levels.

Chassis_Post_BI.dat – Final test in a full chassis, Must have 1 or 2 IMCs and a full chassis. Boards are qualified to ship to customers after this point and single boards. All software/firmware is programmed to shipping levels.

Chassis_Config.dat – Final test in a partial chassis, Minimum configuration is 1 IMC in slot 0. Chassis must be filled

starting from Slot 0 working up to slot 4. Boards and Chassis are qualified to ship to customers as a configured chassis. All software/firmware is programmed to shipping levels.

Chassis_Prog.dat – Low level tests and programs chassis with card types and serial numbers. This test uses a test IMC with a Debug jumper installed. No other cards should be installed in Slots. This is intended to be a replacement for the ProComm currently in use at the Chassis Manufacture

Chassis_test.dat – Pre and Post BI Chassis only tests. This test uses a test IMC with a Debug jumper installed. No other cards should be installed in Slots. This is intended to be a replacement for the ProComm currently in use at the Chassis Manufacture. TLV 99 is programmed in this card if the High speed connector is missing for Post BI testing.

Debug.dat – For debug use only. Carious scripts may be renamed to this for the purpose of debugging a test or Issue, Test type logged is Debug

Order_entry.dat – Allows some Customer order data to be maintained in the CMTEST logs

MAC_assign.dat – intended to Automate the MAC address generation and logging. Not implemented see \\harp\www-int\ops\Process\MAC_Assigments\ for current assignment history.

Demo.dat, Demo2.dat,Demo_Telnet.dat, Test.dat – unused files from the original implementation.

4) – java – To contain Java extensions for a GUI/Web interface. This was TBD. Previous implementation had been done in CGI.

5) Lib – This is the main code area for CMtest. This will not detail all of the functions for each file. This is left to Programmer experience or Training by qualified personnel.

Demo.dat, Demo2.dat,Demo_Telnet.dat, Test.dat – unused files from the original implementation.

APC_Power.pm – main APC power switch library used by CMTEST.pl and powerswich.pl

AX4000.TCL – main routine for Adtech External traffic generator used in Chassis_Extended.dat

Begin.pm – CMTEST begins here.

DataFile.pm – CMTEST HASH file read/create functions

File.pm – CMTEST file read/write operations.

Globals.pm – Defines CMTEST native Globals

GUI.PM – Not used by COMPANY.

Init_HA.pm – Init.pm library modes for COMPANY HA(Dual IMC) systems. Intended to be temporary until functions can be properly folded back to Init.PM

Init.pm – CMTEST native initializations routines for Session, consoles, logs etc.

Init_COMPANY.pm – COMPANY custom initialization routines. Mostly for Defining Globals and configurations.

Logs.pm – CMTEST native logging routines.

Power.pm – CMTEST native power switching routines. For APC and Parallel port

PT.pm,PT2.PM,PT_CGI.pm, CMTEST routines for Various items

PT_DISTY – CMTEST code for, code distribution(Push, update)

PT_QUERY – Code/forms for performing log queries. This function is duplicated in the WWW directory

PTML – CMTEST CGI generation functions.

SigmaProbe – Directory for Sigmaquest Extension to CMTEST. SigmaProbe is not currently in use by COMPANY. Log files should be compatible with SigmaQuest, but no Testing has been done in this area.

Stats.pm – CGI Generation

COMPANY.pm, COMPANY2.pm – COMPANY custom routines for Menus, and Perl extensions/parsing to command files to extract such things as serial numbers.

Utils.pm – CMTEST routines for various items

6) – tftpboot – Contains the COMPANY code that is used on the COMPANY platform. This is typically tftp'd to the system and then used. The directories are currently divided by COMPANY release code(x.xRx_yyyymmdd) directories and COMPANY diag code diagmmddyy directories. These directories and their contents are copied to the Site servers tftp directories where thy can be tftp'd to the system under test. The above method has created some issues with CVS and is to transition to something like OS_[A,B,C,D,E] and diag_{A,B,C,D,E] to match the release pipe features of CMTEST and stop the creation of additional directories in CVS

Diag_mmddyy or diag_{A,B,C,D,E] typical contents:

Diag.bin – Stripped COMPANY OS image for manufacturing test, for the most part this is the same as the COMPANY.bin image, but it does not start-up all of the services

Onscript_GLC_Rx.ksh – KSH script for local execution of tests on each GLC in the system – contains memory tests, temperature and voltage tests

Onscript_MC_Rx.ksh – KSH script for execution of IMC test on each IM in the system , contains Drive, voltage and temperature tests.

onscriptR1.ksh – depreciated, to be removed

readme.txt – contains diag specific release notes

x.xRx_yyyymmdd or OS_{A,B,C,D,E] typical contents:

COMPANY.bin – COMPANY OS image, eventually gets placed on the internal compact flash of the IMC

Bootloader.bin – COMPANY bootloader flash image, eventually gets flashed in each card

Noodle.bin – COMPANY diag flash image, eventually gets flashed in each card

Release.txt – release notes, does not get tftp into system, for developer information only

COMPANYboot – COMPANY initial bootstrap image(was originally u-boot). Eventually gets flashed in each card

COMPANY.cfg – COMPANY default empty configuration, eventually gets placed on the internal compact flash.

7) License.html – Open source License text

8) – uutcfg – Contains the configuration files for each UUT type to be tested. These files get translated into variables used by command files and Perl routines one the UUT part number is determined.

   Parts.cfg - product code(first three digits of serial number) to part number hash, partnumber_revison to ECO hash (allowable revisions)

   UUT_defaul.cfg – default config files

   xxxxx.cfg – Configuration file by part number, May have multiple Hashes for different version of the part number.

9) www – Contains the files used by the site server webpages. Some library functions are duplicates of files in the lib directories, these may or may not be identical files. These files eventually get pushed to the site servers html

   Cgi-bin- files used to build CGI files for status and quires.

   Html – files used for default html page and background.

## 4.4 CVS

CMTEST is under CVS control in the repository called “cmtest”. COMPANY CMTEST official releases are tagged in the following format: cvs tag REL_X_RX_CMTEST_Y_Y_Y. REL_X_RX relates to the COMPANY OS getting release status such as 2.0R1. CMTEST_Y_Y_Y relates to the CMTEST version being release. Currently this has been along the lines of CMTEST_1_8_1 and would relate to files checked out into a //harp/<user>/Test/TestScripts/cmtest18 directory.
Typical use of CVS:
1) cvs checkout cmtest
2) make edits
3) cvs -n update - Check changes but do not update any files
3) cvs update
4) cvs add (directories) - shown as ? in cvs update
5) cvs add (new files) - shown as ? in cvs updatecvs checkput cmtest
6) cvs -q update (verify all ? are gone)
7) cvs diff ( check for your changes)
8) cvs commit (edit(vi) commit message, save and exit starts commit)
Further notes can be found at \\harp\www-int\ops\Test\DebugNotes\.

## 4.5 CYGWN

CYGWN is a free GNU tool for Windows PC. It recommended least the basics of this software be installed on Windows systems used for development. More and more other utilities are relying on the cygwn.dll for there functions, this includes optiperl.

## 4.6 Perl Editors

Optiperl has been the editor of choice for COMPANY development, the final decision is up to the programmer what works best. Optiper is a Visual editor for IDE CGI and Perl programming. See www.xarka.com/optiperl for more details. Most development has been done on personal copies installed on manufacturing PC's MFG-TEST1 in the Manufacturing lab has a copy and licenses provided by Paul Tindle, COMPANY does not have the license to reinstall this software. The cost for single user license is about $60 US. Optiperl when used to edit command files(.dat or .inc) may insert hidden characters. This is only occasionally an issue.

OptiperPerl configuration notes:

CYGWN is needed for the Perl functions in optiperl,

Tools->Internal Server-> Associations: cgi=c:\Perl\bin\perl.exe,pl= c:\perl\bin\perl.exe

Or(May vary depending on your cygwn installation)

Tools->Internal Server-> Associations: cgi=c:\cygwin\\bin\perl5.8.7.exe,pl= c:\cygwin\\bin\perl5.8.7.exe

Tools->Editor->Behavior-> check Show special symbols if you want to see paragraph/EOL marks.

## 4.7 Terminals

The terminal of preference has been TeraTerm.
The final choice is a mater of preference for the programmer.

## 4.8 Windows File Viewer / Editors

The software of choice has been Textpad. This has been useful to watch the CMTEST temporary logfiles in ~/tmp/sx area(Explained more below). The final choice is left to the programmer.

## 4.9 What Happens when you run cmtest

This will give the programmer some idea of what happens when cmtest is executed. Not all routines will be shown, but enough to give a sense of program flow to hopefully navigate the routines when necessary.

1) Execute cmtest.pl

2) Init() -> Init_All < init.pm> -> Init_Also <Init.pm> are executed. These routines setup file paths for logs, initialize global variables, analyze command line options, and rotate temporary logs (~/tmp/Sx x=session number) where necessary. CMtest checks if a session is available and then lock that session so it can not be used by anyone else. The operator is prompted for an ID which is looked up in the users.cfg file to see what level of access if any is available.

3) Menu1() <COMPANY.pm> is called to build the menu presented to the operator. Menu items are qualified based on user level. COMPANY uses a level of 0-10 to change what is presented to the operator. The only operator defined at this time is “test” which shows all level of the test/

4) Menu_Exec <COMPANY.pm> is executed when the operator chooses a menu number a sub in <COMPANY.pm> is called for the particular menu item. The test ID is set to {Debug|SO|EXT|IST|SHIP|Program|FST|BI|ORT|ISTBench] . The sub then call Cmd_Expect <Connect.pm> with the appropriate command file and session port

5) Cmd_Expect <Connect.pm> checks for the command file to exsist, it then opens the appropriate com or Terminal server port. The Command file and Include files will then be process and check for errors. Once completed the Command file will be execution n Process_Cmd_File <Connect.pm>

6) Process_Cmd_File <Connect.pm> continues in a while loop reading commands from a DAT files a associated include files. Calls are made to Exec_Cmd <Connect.pm> as needed to process the DAT and INC files. When entering a loop command alls command inside the loop will be que’d into a flat file that is reexcuted each loop until time expires on the loop.

7) Once command execution is complete control will return to the calling sub in COMPANY.pm. Final <Init.pm> will be called to close down all of the logfiles, session and determine a pass fail.

## 4.10 General Coding Rules

1) Libraries must be <file.pm> located in the lib directory and included with use <file>. 2) Executables targeted for Win32 systems should be names <file>.pl (for association with Perl)
3) Executables targeted for Unix systems should be names <file> (without the extension – to save typing!).
4) CMTEST is not currently running under Perl “srict” error checking. But coding practices should take this into account
5) Use statements may only be used by main scripts, not from within modules (i.e. recursively) ?Why – Perl doesn’t care!Cmtest Menu description
6) Globals - Defined in main::init, init::init_all, , globals.pm –or COMPANY specific globals defined in Init_COMPANYpm, nowhere else! OOP rules may apply later, but not yet!
7) Maintain Perl subroutines in Alphabetical order(.pl,.pm files)
8) $UUT_IP: The IP address and range in the cfg file are used to send to the UUT prior to any IP activity. The Sessions use the sequentially, so session 1 will use the 1st, etc. You can access it by using "<SEND> $UUT_IP" in a command file.
9) Variables in Command and Include files must be declared as Globals

## 4.11 Perl Notes

The following is taken from Paul Tindles CMTEST User_Guide.

Check syntax on a module (library)

Perl –c <module>.pm

References:

Method invocation ( -> ) PP312

My vs. our vs. local PP130

Pseudo CASE structures PP124

Notes:

Assigning hashes - PP11, PP77

```
%Hash = (
        ‘key’ => ‘value’,
        ‘key’ => ‘value’,
    );
```

Arrow (->) operator: - PP90, PP253

<Ref> -> <subscript> or <arg list>

Method call ( ->) – PP91, PP253

<Object or class name> -> <method name>

## 4.12 Command Files.

Below is the current command set for COMPANY CMTEST as of 1.8.x, Taken from Exec_Cmd < Connect.pm>

```
#  \/ - - - - - - - - - white space(s) (indentation) allowed
#                \/ - white space(s) required
#
#       <Alert>                Suspend processing and display a dialog box
#       <Ask>                  'Text16' Var Label
#       <CheckData>     Check for the existence of <Arg> in $Buffer
#       <CheckDataX>    Check for the exclusion of <Arg> in $Buffer
#       <Comment>       Comment line. Deprecated in favor of a #
#       <Ctrl-<x>>             chr - send a <ctrl>-<x> to the raw port, not via comm
#       <Ctrl-Send>            <a char>
#       <End>                  Stop processing and close cmd file
#       <ETTC>                 Expected Time To Completion
#       <Exec>                 <&Subroutine>
#       <GetData>              Returns all data prior to prompt -> $Buffer
#       <Include>              file.inc. May include embedded global var: file${Type}.inc
#       <IncludeX>             Same as <Include> but in $Tmp dir, no synatx check and ok if ! -f
#       <Loop>                 Start of loop, Arg contains ATT to end
#                              Note: No parametric substitution performed inside loop!
#       </Loop>                End of loop
#       <Msg>                  "Info for the operator and log"
#       <Power>                [ ON | OFF | CYCLE <int sec delay>[default: 5 secs]]
#       <Prompt>               "a string" | a_word,  ^ at beginning turns on regexp (use sparingly)
#       <Send>                 "a string" | a_word | <$global_var>        - sends literal + <cr>
#       <Sendslow>          "a string" | a_word | <$global_var>        - sends literal + <cr> paced
          characters
#       <SendChar>      <char><Char><Char>...        - sends literal (without <cr>)
#       <Set>                  <$global_var> [ = 1 ]
#       <SendCharSlow>      <char><Char><Char>...        - sends literal (without <cr>)
           *Note: sendslow can not send a single "0" scalar. Why?
#       <Set>           <$global_var> [ = 1 ] paced characters
#       <Sleep>         <real> secs
#       <Timeout>       <int> secs
#       <UnSet>         <$global_var> [ = 0 ]
#       <Wait>          <$TimeOut> for $Prompt
#       <WaitFor>       "some string" | a_word | <$global_var>
#COMPANY...
   #       <Bypass>          Start of Bypass, Arg contains 0 bypass occurs
   #                      Note: No parametric substitution performed inside loop!
   #       </Bypass>         End of bypass
#/COMPANY...
```

See Appendix 5.2 for some example command files.

## 4.13 Log Files

The following is taken from Paul Tindles CMTEST User_Guide.
There are 3 classes of log files:

- Test logs.
  - These start out life as the command output file, often set with the –O option switch, with a default of ~/tmp/s<n>/cmtest.xml. At the start of a test, any existing file is deleted. On completion of the test, the file is copied to

/var/local/cmtest/logs/logfiles (or the path defined in the local testctrl.cfg file) using a unique file name <UUT_serial_number>-<date_int>.log

- Data logs.
  - These are the master logs that record information about, and the result of, a test. The table below shows the use / content of each, with further details below. The location of these logs is defined in the local test.cfg file, the default location being /var/local/mfglogs

    Data log examples:

### Event Log(/var/local/cmtest/logs/Event.log)

```
1;1175550257;SMV;mfg-svr1;joe;00648-02 Rev
    01;01101400500000027;Bench;FAIL;ATT=687,Diag_Ver=layne010307,ERC=0,Power=APC_OFF,SW_Ver=COMPANYOS Release 3.0R1
    .2007011110.,TEC=6,TID=Bench,TOLF=1175550460,TSLF=485,TTF=154,TTT=688,Ver=1.8.1_Released;0110140050000027-
    20070402.144417.xml;
1;1175551684;SMV;mfg-svr1;joe;00648-02 Rev
    01;01101400500000027;Bench;PASS;ATT=679,Diag_Ver=layne010307,ERC=0,Power=APC_OFF,SW_Ver=COMPANYOS Release 3.0R1
    .2007011110.,TEC=0,TID=Bench,TTT=680,Ver=1.8.1_Released;0110140050000027-20070402.150804.xml;
1;1175285854;SMV;mfg-svr1;joe;00315-01 Rev
    01;0070105370100002;EXT;PASS;ATT=284862,Diag_Ver=layne010307,ERC=0,Power=APC_OFF,SW_Ver=COMPANYOS Release
    3.0R1 .2007011110.,TEC=0,TID=EXT,TTT=284870,Ver=1.8.1_Released;0070105370100002-20070330.131734.xml;
1;1175636384;SMV;mfg-svr1;joe;00292-03 Rev
    01;00201280500000008;Bench;FAIL;ATT=434,Diag_Ver=diag_A,ERC=0,Power=APC_ON,SW_Ver=COMPANYOS Release 3.0R1
    .2007011110.,TEC=9,TID=Bench,TOLF=1175636825,TSLF=0,TTF=37,TTT=441,Ver=Pre-Alpha_Main;0020128050000008-
    20070403.143944.xml;
1;1175714541;SMV;mfg-svr1;joe;00315-01 Rev
    01;0070105370100002;IST;PASS;ATT=1476,Diag_Ver=layne010307,ERC=0,Power=APC_OFF,SW_Ver=COMPANYOS Release 3.0R1
    .2007011110.,TEC=0,TID=IST,TTT=1478,Ver=1.8.1_Released;0070105370100002-20070404.122221.xml;
1;1175716813;SMV;mfg-svr1;joe;00292-03 Rev
    01;00201280500000008;Bench;FAIL;ATT=687,Diag_Ver=diag_A,ERC=0,Power=APC_ON,SW_Ver=COMPANYOS Release 3.0R1
    .2007011110.,TEC=78,TID=Bench,TOLF=1175717502,TSLF=0,TTF=31,TTT=689,Ver=Pre-Alpha_Main;0020128050000008-
    20070404.130013.xml;
```

### Cfg Log(/var/local/cmtest/logs/Cfg.log)

```
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;0;00292-03 Rev 01;0020128050000010;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;1;00722-01 Rev 03;0130204060000005;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;2;00722-02 Rev 02;0130114060000025;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;3;00648-02 Rev 01;0110140050000018;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;4;00648-03 Rev 01;0110133050000033;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;14;00301-03 Rev 01;0080105370100001;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;17;00299-02 Rev 01;0090105370100016;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;18;00299-02 Rev 01;0090105370100017;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;20;00297-02 Rev 01;0100105370100012;
2;1175285854;SMV;mfg-svr1;joe;0070105370100002;21;00297-02 Rev 01;0100105370100003;
```

### Mac Log(/var/local/cmtest/logs/MAC.log)

```
4;1163195460;0070117060100012;00 12 73 00 0A D8 00 08
4;1163195452;0110435060000122;00 12 73 00 0B A0 00 08
4;1163196234;0110323060000118;00 12 73 00 09 90 00 08
4;1163197381;0110323060000114;00 12 73 00 09 70 00 08
4;1163197845;0110323060000112;00 12 73 00 09 60 00 08
4;1163198068;0110323060000115;00 12 73 00 09 78 00 08
4;1163198295;0110323060000117;00 12 73 00 09 88 00 08
4;1163198542;0110323060000116;00 12 73 00 09 80 00 08
4;1163198727;0110323060000119;00 12 73 00 09 98 00 08
4;1169668236;0130204060000002;00 12 73 00 08 20 00 08
4;1175188245;0110133050000043;00 12 73 00 0A 48 00 08
4;1175189678;0110133050000031;00 12 73 00 0A 10 00 08
4;1175191034;0110128050000020;00 12 73 00 02 C0 00 08
4;1175192544;0110133050000047;00 12 73 00 0A 1C 00 08
4;1175279611;0130204060000005;00 12 73 00 08 08 00 08
4;1175280030;0130114060000025;00 12 73 00 07 E0 00 08
```

### History Log(/var/local/cmtest/logs/history.log)

```
03/30 13:17:   1175285854   joe   mfg-svr1     Starting cmtest ...
04/02 13:54:   1175547241   joe   mfg-svr1     Starting cmtest ...
04/02 14:03:   1175547241   joe   mfg-svr1-1    Ending cmtest
04/02 14:24:   1175549047   joe   mfg-svr1     Starting cmtest ...
04/02 14:33:   1175549047   joe   mfg-svr1-1    Ending cmtest
04/02 14:44:   1175550257   joe   mfg-svr1     Starting cmtest ...
04/02 14:55:   1175550257   joe   mfg-svr1-1    Ending cmtest
```

- Execution logs
  - – Currently 6 different logs are maintained:

1) cmtest.log – Shows time stamp and command reading and executing log
2) cmtest.xml – Shows test log in XML format for Sigmaquest use. This file is archived at /var/local/cmtest/logs/logfile at completion of test and is viewable from the web interface at that time. Event.log will contain a link to the file archived.
3) Comm.log – Raw data from comports, HA_Comm.log is also created for a second IMC if installed
4) ExComm.log – Raw Data from terminal server, HA_ExComm.log is also created for a second IMC if installed
5) Expect.log - A log of the expect dialog that Connect::Cmd_Expect had with the UUT. The log file has demarcation lines immediately before every 'send' command to ease readability. File contains the 5 element list returned by the Expect module immediately after the last <Wait> or <WaitFor> command.
6) FlatCmdFile.dat – Dat and Inc files are flattened into this file as the test progresses.
7) _logs_xx~xx_xx~xx~x - Rotated(old logs) are placed in a subdirectory with a time/date stamp for the directory name

The location of all of these logs is ~/tmp/s<n> where n is the session number.

Example Execution Logs

ls –als ~/tmp

```
 4 drwxrwxr-x   2 mfg mfg   4096 Apr  4  2006 s2
16 drwxrwxr-x   2 mfg mfg  16384 Apr  4 01:00 s1
```

ls –als ~/tmp/s1

```
192 -rw-rw-rw-   1 mfg mfg 188673 Apr  4 13:38 cmtest.log
 52 -rw-rw-rw-   1 mfg mfg  50596 Apr  4 13:38 cmtest.xml
 52 -rw-rw-rw-   1 mfg mfg  52893 Apr  4 13:38 Comm.log
 84 -rw-rw-rw-   1 mfg mfg  79220 Apr  4 13:38 ExComm.log
120 -rw-rw-rw-   1 mfg mfg 115391 Apr  4 13:38 Expect.log
112 -rw-rw-rw-   1 mfg mfg 109070 Apr  4 13:38 FlatCmdFile.dat
  4 drwxrwxrwx   2 mfg mfg   4096 Mar 19 11:59 _logs_03~02_14~44~56
  4 drwxrwxrwx   2 mfg mfg   4096 Mar 19 13:58 _logs_03~19_11~59~58
  4 drwxrwxrwx   2 mfg mfg   4096 Mar 22 11:03 _logs_03~19_13~58~34
  4 drwxrwxrwx   2 mfg mfg   4096 Mar 22 11:36 _logs_03~22_11~03~59
  4 drwxrwxrwx   2 mfg mfg   4096 Mar 22 11:37 _logs_03~22_11~36~05
```

## 4.14 Log Files – Table(comparison)

| Field | Event Log | Cfg Log | Mac Log | Description |
|---|---|---|---|---|
| Record Type | 1 | 2 | 4 | |
| Date | 1175285854 | 1175285854 | 1163108683 | Perl 'time' function (local) |
| Location | SMV | SMV | SMV | Location where record was created Current SMV or PW1 |
| HostID | mfg-svr1 | Mfg-svr1 | - | ID of the test host (mfg-lws[1-7] or mfg-svr1 |
| OpID | Joe | Joe | - | Operator ID (ptindle,joe or mfg) |
| Part No | 00315-01 Rev 01 | - | - | Part number of the UUT |
| Serial No | 0070105370 100002 | 00701053701 00002 | 00701170601000 13 | Serial number of the UUT |
| TID | EXT | - | - | Test ID |
| Result | PASS | - | - | |
| Data:ATT | 284862 | - | - | Data string: key=val,key=val, etc<br>Actual test time |
| Data:Diag Ver | layne010307 | - | - | Diagnostics version from INIT_COMPANY.pm |
| Data:ERC | 0 | - | - | Error Code if any |
| Data:Power | | - | - | Current Power status |
| Data:SW_Ver | COMPANYO S Release 3.0R1 .2007011110 . | - | - | COMPANY OS version from INIT_COMPANY.pm |
| Data:TEC | 0 | - | - | Total Error count if any |
| Data:TTT | 284870 | - | - | Total test time |
| Data:Ver | 1.8.1_Releas ed | - | - | CMTEST release version from Release.id |
| Ptr | <file> 0070105370 100002- 20070330.13 1734.xml | - | - | Absolute path to a log file |
| Slot locator | - | 0 | - | Child locator |
| Part No - Child | - | 00292-03 Rev 01 | - | |
| Serial No - Child | - | 00201280500 00010 | - | |
| MacAddr | - | - | 00 12 73 00 0A D0 00 08 | Base Mac Address and range |
| Mac Range | - | - | - | Mac Address consumption |

## 4.15 Web interface

The Web interface is used to monitor test execution, current test status and look at serial number history. The ability to view Child serial numbers is also included. Example web shots are provided in the Appendix. The web interface is located in the CVS/development www directories. Library calls are only the libraies under this directory structure and not the main cmtest directory structure. The files in the www directory will get pushed to a site server with a –s option, otherwise they are ignored during the push process.

CMTest.cgi is the main routine used by index.html

## 4.16 Maintenance/Updating(Push/update)

Notes from Paul Tindle CMTEST installation guide, More information to be provided in the CMTEST developer guide.

There are 4 separate utilities performing these different functions, some of which are duplicated. The table below describes there functions, and which utility is capable of each, and how:

| Function | mkhost | push | update | switch | <cls> |
|---|---|---|---|---|---|
| | | | | | |
| Create local source buckets | Yes | | | | [def] |
| Get distribution master from SCCS tree* | Yes | | | | -x -X |
| Create distribution floppy master | | Yes | | | -f |
| Create <cmd> -> <cmd>.pl links (deprecated) | No | No | No | | [def] |
| Create / configure expansion serial ports* | Yes | | | | -E |
| Copy master files to local bins | Yes | Yes | Yes | | [def] |
| Distribute (new) master files to all known hosts | | Yes | | | -a |
| Update ‘Production Release’ pointer | Yes | | Yes [-s] | Yes | [def] |
| Gets master files from NFS mount* | | Yes | Yes | | |
| Gets master files from floppy* | | | Yes | | |
| Pointer control to prior / future releases | | | | Yes | many |
| Switch approve release control | | | | Yes | -a |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

*Function not tested in COMPANY environment

Typical COMPANY update process:

1) Edit file in Development tree
2) From Devlopment tree cmtest/bin execute ./push.pl –X .. to push the development area to the distribution area, add a –s to include site server(web page) updates
3) From Devlopment tree cmtest/bin execute ./update.pl to update local distribution buckets(if needed)
4) From Devlopment tree cmtest/ execute ./push.pl –a or ./push.pl –H mfg-lws1 to push to all siteservers in the push.cfg or specify a single site server with –H. Add a –s if the site server(www) files are to be updated.

5) From Devlopment tree cmtest/ execute ssh mfg-lws1 /var/local/cmtest/dist/bin/update.pl to remotely update a distribution bucket. Repeat for all remote systems to be updated.
6) The above is used for mostly a single developer environment, in a multi-developer environment the above should be done from a SCCS(COMPANY cmtest CVS) tree that has been checked out and tested.
7) Additional notes in \\harp\www-int\ops\Test\DebugNotes\update example.txt

# 5.0 Appendix

## 5.1 Glossary of Terms

- CMTest - Current* label for the Automated Test Executive
- UUT - Unit Under Test
- ATT - Actual Test Time [Secs] (excluding operator wait time)
- TEC - Total Error Count
- TID - Test ID
- TOLF - Time Of Last Failure*
- TSLF - Time Since Last Failure*
- TTF - Time To [1st] Failure [Secs]
- TTT - Total Test Time [Secs]
- Time [tick count] format.

## 5.2 Example-Command File(DAT and INC).

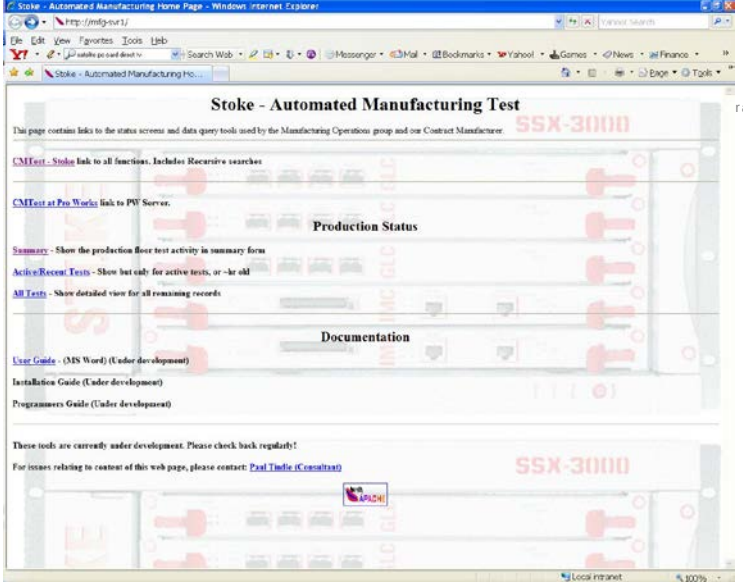
This example the Bench.dat turns on power and calls Stop_COMPANY_boot.inc. Stop_COMPANY_boot.inc stops the bootup process then the command <GetData> POST parses the data collected during bootup with the routines Get_Data <COMPANY.pm> and Check_POST_INFO <COMPANY2.pm>. After executing the routines Stop_COMPANY_boot.inc will return to Bench.dat with the $Bench_card_type_gbl variable set to Bench_Test_GLC.inc. Bench_Test_GLC.inc will then execute to completion and then return to Bench.dat which will turn off the power and exit.

```
#################################################################################
#
# Module:     Bench.dat
#
# Author:     Joe White
#
# Descr:      Main Execution for IMC Bench test
#
# Version:    1.1
#
```

```
# Changes:    Created from Bench_IMC.dat  10/19/06
#
# Includes:  Check_POST_IMC.inc
#                                   Show_IMC_Cpld.inc
#                                   i2c_scan_imc.inc
#                                   check_thermal.inc
#                                   tftp_boot_imc_diag.inc
#                                   format_IMC_drives.inc
#                                   Check_IMC_PCI_V.inc
#                                   Check_Disco_imc.inc
#          Check_IMC_drives.inc
#                                   tftpcopy_COMPANY.inc
#                                   cfint_boot_imc.inc
#                                   Bench_Test_IMC.inc
#                                             Check_PCI_V.inc
#
# Still ToDo:
#
#
#          Copyright (c) 2006 COMPANY. All rights reserved.
#
################################################################################
<Sleep>               5
<Prompt>"COMPANY>>"
<TimeOut>             30

<Msg>                 "Turning on UUT Power ..."
<Power>               ON

<Include>Stop_COMPANY_boot.inc

#<Msg>                $Bench_card_type_gbl
<Include>${Bench_card_type_gbl}
<Msg>                 "Turning off UUT Power ..."
<Power>               OFF

<End>

################################################################################
#
# Module:     Stop_COMPANY_boot.inc
#
# Author:     Joe White
#
# Descr:      Called by Dat and Inc files
#
# Version:    2.1
#
# Changes:    Modified for R1 release 01/03/06
#                                    Modified for 3.0
#
#
# Still ToDo:
#
#
#          Copyright (c) 2006 COMPANY. All rights reserved.
#
################################################################################
<Prompt>"COMPANY>>"
<TimeOut>             30

<Msg>                 "Stopping COMPANY bootup.."
<WaitFor>             "Initializing Network Hardware"
<SendChr>                         uboot
# For backward compatibility to 2.0R2 Only need to keep this until the next release
<Sleep>               1
<SendChr>                         "COMPANY"
<Wait>
```

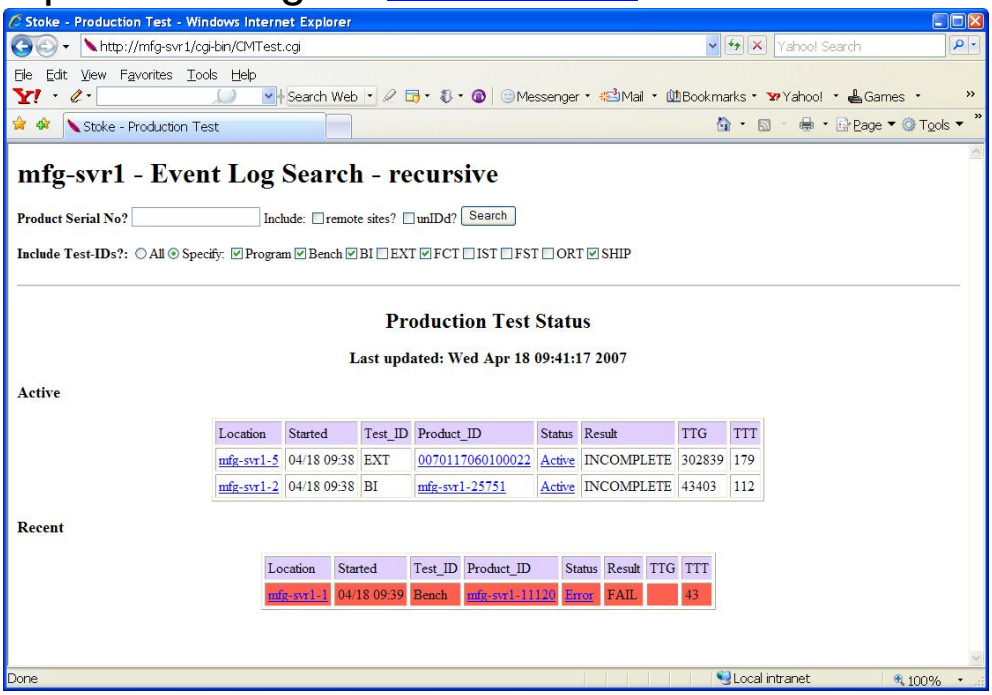
```
<GetData>  POST

        ##############################################################################
#
# Module:     Bench_Test_GLC.inc
#
# Author:     Joe White
#
# Descr:      Include for Bench_GLC.dat
#
# Version:    6.6
#
# Changes:    updated for R1 release 010306
#                                 Updated for IXP2805 052506
#                                 Change Xscalup msg for 2805
#                                 Updated for 3.0
#                                 Added Disable "noreload"
#                                 010507 Updated for laynes pfeffa test tcam(bypass qtalk)
#
# Used By:
#                                         BenchGLC.dat
# Includes:
#                                         Bench_Test_GLC.inc
#                                         Check_PCI_V.inc
#
# Still ToDo:
#
#          Copyright (c) 2006 COMPANY. All rights reserved.
#
##############################################################################


<Msg>               "Test Disco"
<Prompt>"> "
<Timeout>           10
<Send>              "pfeffa"
<Wait>
#Take IXF out of reset
<Send>              "w8 f4000010 1"
<Wait>
<Send>              "w8 f4000010 0"
<Wait>
#MAC soft reset control - enable all MACs
<Send>   'pfeffa  wr ixf 505 0'
<Wait>
<Send>              "test ixf access"
<Wait>
<CheckData>         "IXF1110 Access Test PASS"
<Msg>               "Test Disco DX"
<Send>              "test dx access"
<Wait>
<GetData>
#<GetData>          TESTING_PCI_ACCESS_TO_PRESTERA
<CheckData>         'TESTING PCI ACCESS TO PRESTERA 0 ...  PASSED'
<CheckData>         'TESTING PCI ACCESS TO PRESTERA 1 ...  PASSED'
…<file truncated>
```

## 5.3 Example-WebPage – Home Page

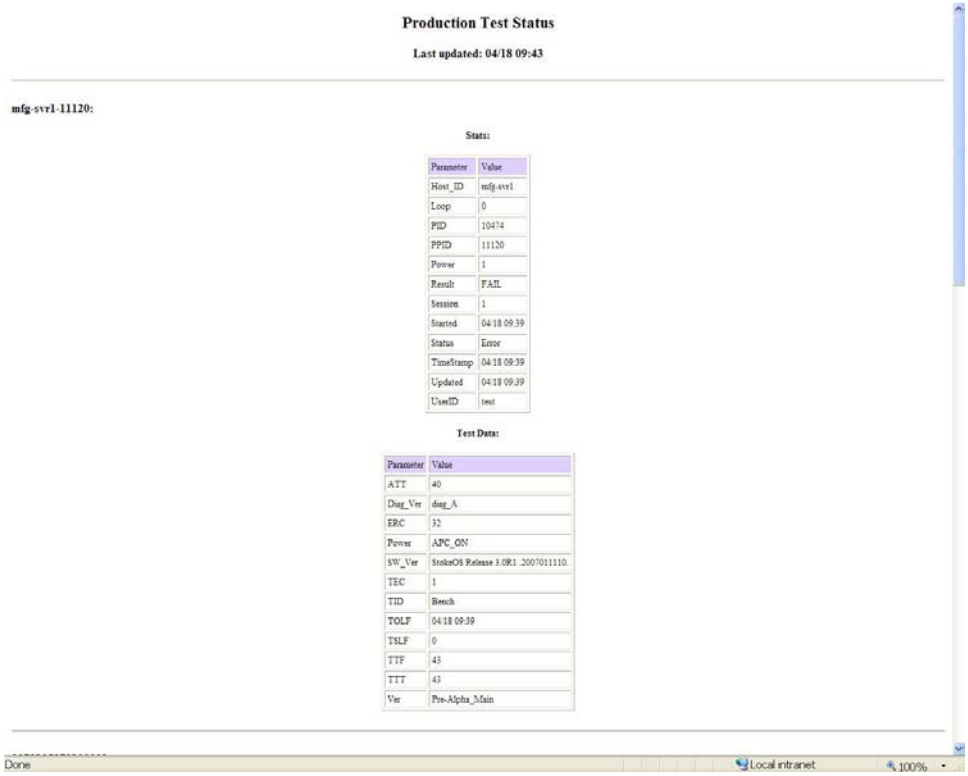



1) **CMTest - COMPANY** link to all functions. Includes Recursive searches - Show Production status for that Site server and has area to do Searches on serial numbers for test information
2) **CMTest at Pro Works** link to PW Server – Link to the PW site server(mfg-lws1). Note: There is no link to return to the COMPANY Site server (mfg-svr1). Currently the HTML code is not Site server aware
3) **Summary** - Show the production floor test activity in summary form – Shows only the production status, no fields for doing Serial number searches
4) **Active/Recent Tests** - Show but only for active tests, or ~hr old - Shows test detail for Active tests
5) **All Tests** - Show detailed view for all remaining records – Shows test detail for a tests information remaining in /var/local/cmtest/stats/*.txt. There is currently a CRON job that deletes files older than 30 days.
6) User, Installation,Programers Guide – Not currently linked.

## 5.4 Example-WebPage – CMTest - COMPANY **link to all functions**



## 5.5 Example-WebPage – Active/Recent Tests - **Show but only for active tests,**



**Production Test Status**

**Last updated: 04/18 09:43**

**mfg-svr1-11120:**

**Stats:**

| Parameter | Value |
|---|---|
| Host_ID | mfg-svr1 |
| Loop | 0 |
| PID | 10474 |
| PPID | 11120 |
| Power | 1 |
| Result | FAIL |
| Session | 1 |
| Started | 04/18 09:39 |
| Status | Error |
| TimeStamp | 04/18 09:39 |
| Updated | 04/18 09:39 |
| UserID | test |

**Test Data:**

| Parameter | Value |
|---|---|
| ATT | 40 |
| Diag_Ver | diag_A |
| ERC | 32 |
| Power | APC_ON |
| SW_Ver | StokeOS Release 3.0R1 .2007011110. |
| TEC | 1 |
| TID | Bench |
| TOLF | 04/18 09:39 |
| TSLF | 0 |
| TTF | 43 |
| TTT | 43 |
| Ver | Pre-Alpha_Main |

Done Local intranet 100%

## 5.6 Conventions

- `Plain courier` Names of variables shown in command syntax that you replace with your own network information. This convention also identifies actual display output that has been copied from the router.
- **`Bold courier`** Command names and keywords shown in the text and references. These commands must be entered exactly as shown. It also highlights significant lines in the sample output.
- {} Curly braces indicate a choice of required keywords or variables. You must enter at least one of the enclosed parameters. < > Angle brackets indicate variables for user input. Replace the angle brackets and variable name with information that is indicative of your setup. | Pipe operators indicate a choice. You can enter one of the parameters on either side of this operator.
- [ ] Square brackets indicate a choice of optional keywords or variables.