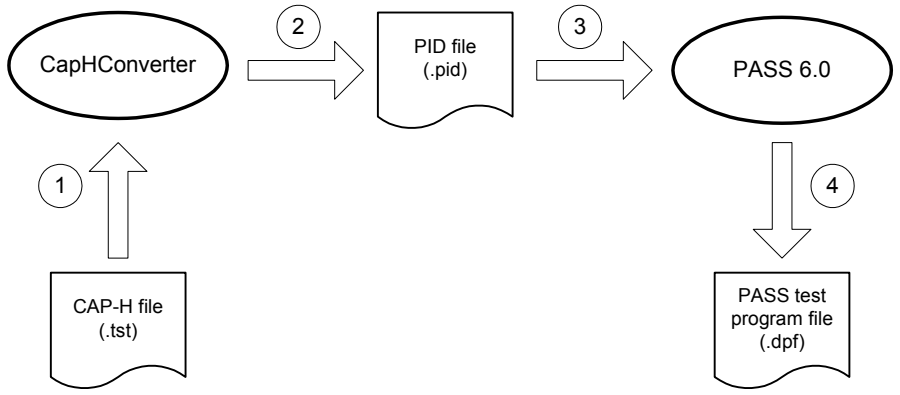
XL APPLICATION NOTE
Cap-H to PID Converter

The Cap-H to PID Conversion Utility builds a Dynalab PID file from a Cap-H test output file. This document describes the operation and limitations of the Converter utility.

Overview

The following diagram illustrates the process for converting an output file from the Cap-H system to a Dynalab PASS 6.0 program file:



1 The CAP-H Conversion utility accepts a file generated by the CAPITAL H system as the input file.

2 The CAP-H Conversion utility converts the data in the Cap-H file to a file suitable for importation into PASS 6.0 (PID file).

3 Using PASS 6.0, the PID file is imported.

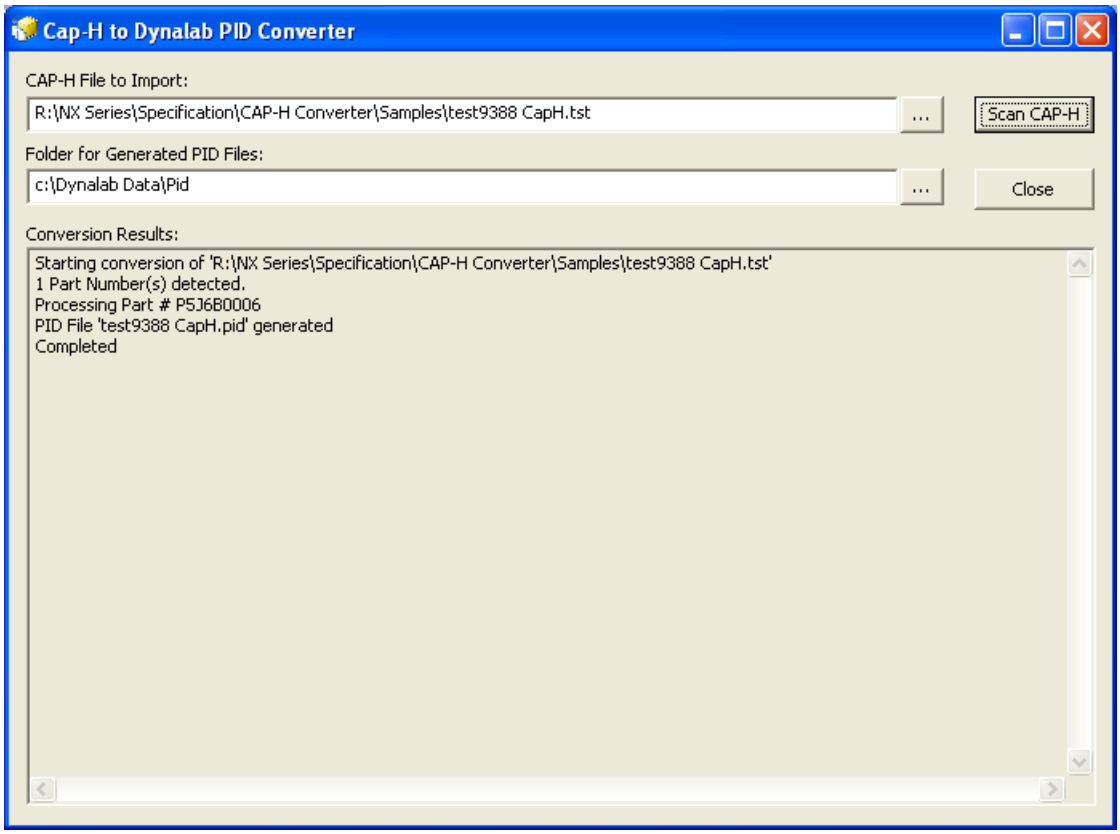
4 PASS 6.0 creates a test program suitable for execution on Dynalab XL Series Analyzers.

Running the Converter

The Converter is a standalone program that runs on Windows 2000 or later. The user can control the utility either though a Windows interface or via a command line.

Windows Interface Operation

When executing the Converter, the following window appears:



The fields are described below:

Cap-H File To Import

This field is the file to be processed by the Converter. To browse for a file press the "…" button

Folder for Generated PID files

The directory in which generated PID files will be placed. See "Conversion Process Details" for more information.

Conversion Results

All error and status messages are displayed in this field. The contents can be copied to the clipboard by right clicking on the field.

Scan Cap-H

This button starts the conversion process. It is only enabled if both the input and output file have been specified.

Close

Closes the application.

Command Line Operation

The Converter command line has the following format:

`CapHConverter [/Q] [/Y] [`*`input file name`* `[`*`output file directory`*`]]`

Parameter Details:

/Q	Run in Quiet mode (does not open any display windows)
/Y	Overwrite output files without asking
Input file name	Path and file name of the input Cap-H .tst file. Required if /Q is set.
Output file Directory	Directory (folder) for the generated PID files. Required if /Q is set.

Parameters are not case-sensitive. The flags can be proceeded by '/' or '-'.

The behavior of the converter depends on the /Q (Quiet) flag. Without the /Q flag the normal Windows user interface is displayed. If not in Quiet mode if both filenames are provided the Converter will automatically start the conversion.

When the /Q flag is set, the conversion will not provide any visible user feedback. Instead, any error or status messages will be added to a log file with the name "*input file name.log"*. If /Q is set and /Y is **not** set the Converter will fail if the output file(s) already exist.

Example

CapHConverter /Q /Y "test9388 CapH.tst" "C:\Dynalab Data\Pid"

Conversion Process Details

Like any conversion process the Converter must follow some conventions and handle some limitations of the two file formats. These are described below.

Part Numbers

The Cap-H test file format support multiple part numbers in the same .TST file. However, the PID file format and PASS 6.0 only support one part number per file. Therefore, if the Cap-H file has more than one part number, multiple output files will be generated with one part number in each PID file. The file names are generated by using the input Cap-H file name for the first part number then appending "_#" to each subsequent part number where "#" is a number starting from 1. For example if the Cap-H file contains 4 part numbers and the Cap-H file name is "Sample.tst" then the output files would be:

Sample.pid

Sample_1.pid

Sample_2.pid

Sample_3.pid

The Conversion Results field (or .log file in Quiet mode) will display which part numbers are in which output files.

Names

Spaces in the names of harness elements such as wires and connectors are not allowed in the PID file. Any connectors, pins, diodes, or wires with names that contain spaces will have the spaces converted to the underscore ('_') character.

Field Mapping

This section describes the details of how Cap-H fields are mapped to PID fields.

Comments

Each generated PID file has a comment section at the top describing the source Cap-H file, the date and time of the conversion, and the part number associated with that particular PID.

Header Fields

Header data from the Cap-H file is mapped to the [header] section of the PID file. When the PID is imported to PASS 6.0 the data is placed in the PASS Message Table. Part Number data is mapped into each part number's PID file after the regular header data. The mappings are as described below:

Header Field	Message Table Entry	Comments
FAMILY	101	
CLIENTFAM	102	
DATE	103	
CLIENT	104	
EXTRA	105	
Part Number Fields		
NAME	106	
CLIENTPN	107	
LEVEL	108	
DATE	--	Not mapped
EXTRA	109	

Connection Fields

In general, all data from the Cap-H file is imported into the PID except for the COLOR_A and COLOR_B fields for wires and splices. COLOR fields are imported as a combination of wire color and stripe color if there is a "/" in the color. Otherwise only a wire color is imported and the stripe color is empty.

Diodes in Cap-H do not have wire names so when a diode is imported the wire names are generated from the diode's name in the format *diodename*_a for the anode side and *diodename_c* for the cathode side.

Connector Fields

Fixture DRAWING fields are not preserved from the Cap-H file. All other Fixture information is mapped to the PID file. Splice names are not supported in PASS so splice names are not preserved. Splices are instead assigned splice numbers in the order in which the splice appears in the CONNECTOR section of the Cap-H file.

CAP-H File Format

The Conversion Utility described in this document is based on the assumption that the input CAP-H file is consistent with a specific format specification. The following pages describe the format specification for the input file.

CAPITAL H TO TEST SYSTEMS PROPOSAL FILE STRUCTURE

The file structure is the following:

[HEADER]
FAMILY = *char[14]* *(optional*
CLIENTFAM = *char[22]* *(optional)*
DATE = *YYYYMMDD(optional)*
CLIENT = *char[40]* *(optional)*
EXTRA = *char [80]* *(optional)*
[PARTNUMBER]
NAME = *char[14]*
CLIENTPN = *char[22]* *(optional)*
LEVEL = *char[14]* *(optional)*
DATE = *YYYYMMDD(optional)*
EXTRA = *char [80]* *(optional)*
[ENDPARTNUMBER]
.
.
.
[PARTNUMBER]
NAME = *char[14]*
CLIENTPN = *char[22]* *(optional)*
LEVEL = *char[14]* *(optional)*
DATE = *YYYYMMDD(optional)*
EXTRA = *char [80]* *(optional)*
[ENDPARTNUMBER]
[ENDHEADER]

[CONNECTION]
[PN = *char[14]*]
[WIRE]
NAME = *char[10]*
COLOR = *XX/XX* *(XX colour codes as other specifications) (optional)*
CONNECTOR_A = *char[10]*
PIN_A = *char[10]*
SPLICE_A = *char[10]*
COLOR_A = *XX* *(XX colour codes as other specifications) (optional)*
CONNECTOR_B = *char[10]*
PIN_B = *char[10]*
SPLICE_B = *char[10]*
COLOR_B = *XX* *(XX colour codes as other specifications) (optional)*
[ENDWIRE]
[DIODE] *(Extremity A – Anode, Extremity B – Cathode)*
NAME = *char[10]*
COLOR = *XX/XX* *(optional)*
CONNECTOR_A = *char[10]*
PIN_A = *char[10]*
SPLICE_A = *char[10]*
COLOR_A = *XX* *(optional)*

```
                CONNECTOR_B = char[10]
                PIN_B = char[10]
                SPLICE_B = char[10]
                COLOR_B = XX (optional)
            [ENDDIODE]
            ...
        [ENDPN]
        ...
        [PN = char[14]]
            ...
        [ENDPN]
[ENDCONNECTION]

[CONNECTOR]
        [FIXTURE]
            NAME = char[10]
            DRAWING = char[80] (optional)
            [PIN]
                NAME = char[10]
            [ENDPIN]
        [ENDFIXTURE]
        ...
        [SPLICE] (optional)
            NAME =char[10]
        [ENDSPLICE]
        ...
[ENDCONNECTOR]
```

The file will be an ASCII file. The physical structure is composed by lines. These lines can be

- begin/end block lines ([block name])
- data lines (VARIABLE NAME = value).

The contents of this structure is the "minimal" information to make the electrical test of one family harness. The logic structure is composed by three main blocks: [HEADER], [CONNECTION] and [CONNECTOR].

The structure [HEADER] – [ENDHEADER] contents the definition of the family of each derivative ([PARTNUMBER] – [ENDPARTNUMBER]).

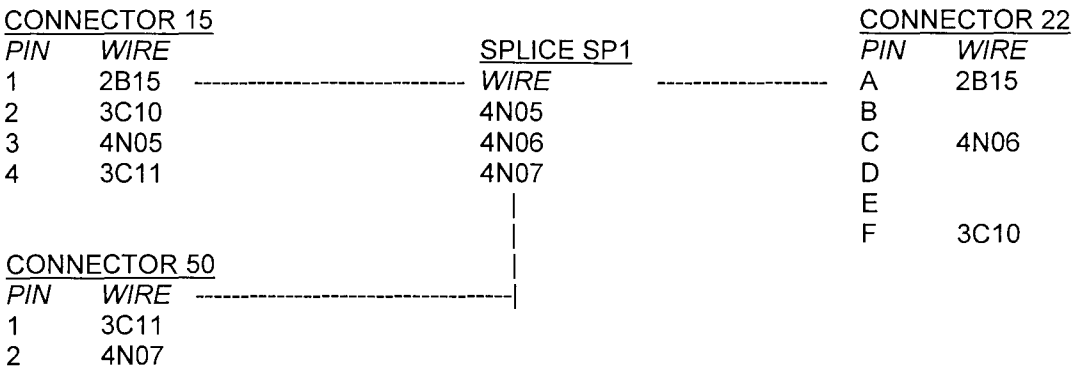
The structure [CONNECTION] – [ENDCONNECTION] contents the information of the connections for each derivative [PN = *value*] – [ENDPN].
Each connection describes their type; currently only considerated the components [WIRE] and [DIODE] (data to detect only the presence). New connection types can be used in the future with this structure.

The structure [CONNECTOR] – [ENDCONNECTOR] is used to describe the necessary pins. The substructure necessary is [FIXTURE] – [ENDFIXTURE] , that has the information of the connector pins [PIN] – [ENDPIN]. The sub-

structure [SPLICE] – [ENDSPLICE] is not necessary to build the harnesses connection, but is useful to check the structure consistency.

We must considerer that the variables order is not fix, for the reading file process, and it can appear new variables in one structure, or it can appear new structures. If we follow this recommendations it is possible that the file can be read after future variations.

This is an example:




```
[HEADER]
        FAMILY = 554633167
        CLIENTFAM = VCM9844576
        DATE = 19980211
        CLIENT = VOLVO
        EXTRA =
        [PARTNUMBER]
                NAME = 554633210
                CLIENTPN = VCM9844580
                LEVEL = 003
                DATE = 19980211
                EXTRA =
        [ENDPARTNUMBER]
[ENDHEADER]


[CONNECTION]
        [PN = 554633210]
                [WIRE]
                        NAME = 2B15
                        COLOR = RD
                        CONNECTOR_A = 15
                        PIN_A = 1
                        SPLICE_A =
                    COLOR_A =
                        CONNECTOR_B = 22
                        PIN_B = A
                        SPLICE_B =
                        COLOR_B =
                [ENDWIRE]
                [WIRE]
                        NAME = 3C10
                        COLOR = GR/BL
```

```
                CONNECTOR_A = 15
                PIN_A = 2
                CONNECTOR_B = 22
                PIN_B = F
            [ENDWIRE]
            [WIRE]
                NAME = 4N05
                COLOR = BL
                CONNECTOR_A = 15
                PIN_A = 3
                SPLICE_B = SP1
            [ENDWIRE]
            [WIRE]
                NAME = 3C11
                COLOR = YE
                CONNECTOR_A = 15
                PIN_A = 4
                CONNECTOR_B = 50
                PIN_B = 2
            [ENDWIRE]
            [WIRE]
                NAME = 4N06
                COLOR = BL
                SPLICE_A = SP1
                CONNECTOR_B = 22
                PIN_B = C
            [ENDWIRE]
            [WIRE]
                NAME = 4N07
                COLOR = BL
                SPLICE_A = 15
                CONNECTOR_B = 50
                PIN_B = 2
            [ENDWIRE]
        [ENDPN]
[ENDCONNECTION]

[CONNECTOR]
        [FIXTURE]
            NAME = 15
            DRAWING =
            [PIN]
                NAME = 1
                NAME = 2
                NAME = 3
                NAME = 4
            [ENDPIN]
        [ENDFIXTURE]
        [FIXTURE]
            NAME = 22
            DRAWING =
            [PIN]
                NAME = A
                NAME = B
                NAME = C
                NAME = D
                NAME = E
                NAME = F
            [ENDPIN]
```

```
        [ENDFIXTURE]
        [FIXTURE]
                NAME = 50
                DRAWING =
                [PIN]
                        NAME = 1
                        NAME = 2
                [ENDPIN]
        [ENDFIXTURE]
        [SPLICE]
                NAME = SP1
        [ENDSPLICE]
[ENDCONNECTOR]
```