



# Calypso Tips and Tricks

Image Design Technology Pty Ltd  
Unit 2, 33-35 Alleyne Street  
Chatswood NSW 2067  
Phone (61) 1300 666 099  
Fax (61) 1300 555 069  
[www.idt.com.au](http://www.idt.com.au)  
ABN 82 082 281 543

## TECHNOLOGY DISTRIBUTORS

Last Updated – 31/08/2006

### (ProI/O and ION series)

#### 1. Mixing Hex and ASCII

How to add single hex characters within an ASCII string ?

Use **\x** before the Hex character.

Example - The Pioneer Power ON string is written as: 02h\*\*PON03h  
02h and 03h are hex, \*\* are wildcards for units address, PON is ASCII

This requires hex and ASCII commands to be sent.

Use **\x** followed by the character required to be sent as hex.

```
#COM1[T1,"\x02**PON\x03"]; - (Pioneer plasma example for POWER ON)
```

Beware! The some devices may actually want all of the characters in Hex. We saw this recently with a Panasonic projector that listed ASCII commands very similar to these with a Hex start and end byte, but in reality it wanted the Hex equivalent of each character. There is an ASCII to Hex conversion table in the manual, so that can assist you if Hex is required. The command, if all of it was in Hex, would look like this:

```
#COM1[T2,"02**50494E03"];
```

#### 2. Sending RS-232 code, examples

Sending **hex** strings:

NEC LT35 projector example - ON and OFF

**Function ON code data** 02H 00H 00H 00H 00H 02H

**Function Off code data** 02H 01H 00H 00H 00H 03H

Calypso strings would be written as;

```
ON #COM1[T2, "020000000002"];
```

```
OFF #COM1[T2, "020100000003"];
```

T2 dictates that the string be in hex

Sending **ASCII** strings:

Clarity Baycat-X example – ON and OFF

**Operation** display.power **Command type = or ? Value** OFF or ON

Calypso strings would be written as ;

```
ON #COM2[T1,"OP**display.power=ON\r"];
```

```
OFF #COM2[T1,"OP**display.power=OFF\r"];
```

T1 dictates that the string be in ASCII

OP and \*\* are Clarity specific commands

\r is carriage return special character for ASCII

Pioneer DVD-5000 example - PLAY

**Function** PLAY **code data** PL<CR>

Calypso strings would be written as ;

```
PLAY #COM1[T1, "PL\r"];
```

\r is carriage return special character for ASCII

### 3. Adding a "Wait" condition

You can't implement a wait between actions in the same event, but you can implement a wait between two distinct events. For example, a user may enter the room, press a button to turn on a system, then expect that once the projector warms up the correct input will be selected and the lights will dim, etc. Thus giving the user time to load a DVD.

#### Event 1 - System On

- Action 1 - power on projector (COM or IR command)
- Action 2 - power on audio receiver (IR command)
- Action 3 - power on DVD player (IR command)
- Action 4 - Call Event 2

#### Event 2 - System On 2

- Action 1 - power on Cable/Satellite TV receiver (IR command)
- Action 2 - power on VCR (IR command)
- Action 3 - Wait 30 seconds then call Event 3 #EVT3[W30];

#### Event 3 -

- Action 1 - switch projector to desired input (COM or IR command)
- Action 2 - dim lights (IR command?)

### 4. Mini-Jack Configuration

The stereo mini jacks tip is 1, ring is 2 and sleeve is ground. No additional configuration is required to use the rear IR jacks as inputs. If using a mono cable to send IR to the rear inputs you can use any of the jacks.

All four rear IR jacks and the front IR window are considered #XR11 in the database. If you want to maximize inputs and outputs using a splitter you would connect the input(s) to 1, 3, 5, or 7, and the output(s) to 2, 4, 6, or 8.

### 5. Dealing with Inter Character Delays

When devices require a delay to be placed between characters being sent out the com ports, (e.g. Magenta MV8x8), it is necessary to utilise the "Dn" flag with the #COM command. Where D = delay period and n = period or units of time.

e.g. to recall preset #7 on a Magenta MV8x8 matrix the command is R, preset number, carriage return, or R7<CR>.

Therefore a Calypso Event would be made up of the following actions;

- Action 1 = #COM1["R",D1]
- Action 2 = #COM1["7",D2]
- Action 3 = #COM1["\r",D3]

Note #1: \r is a Calypso special character for Carriage Return

Note #2: As each action in an event is sent simultaneously, it is required that the delay period increase, to allow each character to be sent after a short delay

## 6. Net Commands

A #NET command is a command the Pro I/O or Ion will use to send a command via ethernet. If there is a device available that can receive commands via ethernet you will need the unit's IP address, the port through which it will receive the commands, and in quotes the Calypso device will send the data payload. One common example is ethernet-to-serial converters. Perhaps you are sending a command via ethernet to an E-to-S converter attached to a switcher where the converter's IP address is 192.168.1.111, the port through which it receives ethernet commands is 2180, and the switcher wants to see "I1O2" to switch input 1 to output 2. The command might look like this:

```
#NET[I192.168.1.111, P2180, "I1O2"];
```

The NET command can also be used to send commands to other Calypso devices:

```
#NET[I192.168.1.101, P5000, '#COM1[T1, "PWR ON"];'];
```

## 7. "Command timed out" whenever a button (event) is called

The "timeout" will generally indicate a networking issue. Usually it means the desired IP address isn't available. If you are getting a custom page directly from the device, the device is obviously available. However, the HTML code maybe pointing to a specific device--in this case 192.168.1.101--which may or may not be the same device as the one serving the page, and may or may not be available on the network.

## 8. c-Link Command format

<http://192.168.1.101/button.cgi?Event=111>

Simply change the IP and event number to correspond to the event to be called.

## 9. c-Link Command Syntax

When using or creating c-Link commands or hyperlinks ensure that the correct syntax is used as follows; <http://192.168.1.100/button.cgi?Event=25>

example:

<http://192.168.1.100/button.cgi?event=1>

The above will not work because the e in event must be E (capital)

## 10. Calling Button Events from a Custom Page

To use a button URL that will always point to the device serving the page, drop the "<http://192.168.1.100/button.cgi?Event=25>.." stuff and just start the URL with "/button.cgi...". With partial URLs, the default action is to automatically add on the current location, in this case the IP address of the actual device serving the page. Setting up a web page this way would allow the same "custom\_page.htm" file to be used on multiple Calypso devices and work correctly with each one (assuming the events are all defined in each case).

## 11. One important note to consider when creating your database

If you have serial compare event triggers and are trying to do a serial receive from the command line, the Pro I/O event database will take priority. It will take the data in the receive buffer and first compare it to possible triggers in the event database. Finding none it will discard the data.

You should temporarily disable the event triggers for Serial Compare events before doing a serial receive on the Command Line.

## 12. Timer functions

### #Timed Events:

You first have to go to Configuration --> Miscellaneous and set the day and time of the Pro I/O in 24hr time (1:00pm = 13:00) and click Save Setting. Then in the Event Database --> Event Editor under Trigger QuickList select Timer. In the Trigger Input Event field you now have two ways of setting up a timed event. You can set it up based on a specific day/hour/minute, or you can set up the event to repeat every X minutes. Here are some examples:

If you want an event to occur weekly at a given time you would use the format

**#TMR[Wd:h:m];**

W defines weekly, d is the day of the week (1 = Sunday), h is hour, m is minute. So say you want an event to occur every Monday morning at 9:00. You would enter the following in the Trigger Input Event field:

**#TMR[W2:09:00];**

Daily at 5pm

**#TMR[D17:00];**

Hourly at 20 minutes past the top of the hour

**#TMR[H20];**

If you want an event to occur on a given time interval you would use the format

**#TMR[Mx, Ox];**

M means modulus, O is offset. The modulus is how large (in minutes) the interval is. Every 10 minutes would be M10. The offset, if the user desires to have one, is how far from zero time (in minutes) the user wants the Pro I/O to wait before initiating this event where zero time is Sunday at 12:00am. So say you want an event to occur every 20 minutes.

**#TMR[M20];**

Every 15 minutes, but not until Monday morning at 7am.

**#TMR[M15, O1800];**

### 13. Learn IR Universal with 1 skip and 2 skip

This came about from a client who was trying to learn IR codes from a Pioneer Elite DVD player. He was learning codes but they wouldn't work when he tried to re-transmit them.

We broke down the IR string and it turned out there were three distinct codes within the same blast. We now allow the user/integrator to learn the 1st, 2nd, or 3rd code in a given IR blast if necessary. Learn universal would be the normal mode, but if you wanted to learn the 2nd code you would use "1 skip". 3rd code would be "2 skips".

### 14. Toggle Events

#### Event 1

Input trigger: #XRI1[F1,"08B1"];

Action 1: #TRG["OFF",1];  
Action 2: #COM1[T1, "Data String"];  
Action 3: #GPO1["CLOSE"];  
Action 4: #EVT2[W2];

#### Event 2

Input trigger: #NOP[];

Action 1: #TRG["ON",3];

#### Event 3

Input trigger: #XRI1[F1,"08B1"];

Action 1: #TRG["OFF",3];  
Action 2: #COM1[T1, "Data String"];  
Action 3: #GPO1["OPEN"];  
Action 4: #EVT4[W2];

#### Event 4

Input trigger: #NOP[];

Action 1: #TRG["ON",1];

After you've saved all the events you will need to return to the Event Viewer page and make sure the trigger is enabled for only one of the events in the toggle cycle. Make sure the trigger for Event 1 is checked, but NOT the trigger for Event 3. That way the toggle will work properly. If you go back and edit an event in the cycle you will have to do the same thing. Every time you save an event you enable its trigger.

Another thing to be aware of regarding toggles:

A "+" in the trigger check box means the trigger is normally OFF but is currently ON  
A "-" in the trigger check box means the trigger is normally ON but is currently OFF

## 15. Factory resetting Pro I/O and Ion-e units

### Early revisions: (pre 1.7)

Power down the ION-e or Pro I/O (1.5.0).

Short the tip/ring of a 3.5mm jack plug and insert it into IR port #1.

Power up the controller and establish a browser connection.

Remove the shorted plug that was inserted into IR input #1.

Personality configurations will be reset **but** the database will not be altered !

### Latest revisions: (post 1.7)

Power down the unit

Jumper TX/RX on COM1, power on, then remove the jumper within 4 seconds. That will reset to the default.

On the LT and LT2 you will hear three quick beeps to confirm that it has been reset.

## 16. Database size

The database file size for the Pro I/O and ION series controllers should always be 276 or 277K regardless of how many of the available 128 events are actually used.

## 17. TST and VAL commands

TST is a trigger only command that is tied to the VAL command. TST1=VAL1  
Through the event database (or the Command Line) you could set, increase, or decrease VALx as needed. The corresponding TSTx would then trigger based on parameters you set for it. Great for projector monitoring to keep track of what state a projector is currently in. 0 is off, 1 is on, 2 is standby, etc.

Trigger:

#TSTx[=y]; Event will trigger when equal to y

#TSTx[!y]; Event will trigger when NOT equal to y

#TSTx[>y]; Event will trigger when greater than y

#TSTx[<y]; Event will trigger when less than y

#TSTx[y-z]; Event will trigger when between y and z (inclusive)

Action:

#VALx[+y]; Increase by y units

#VALx[-y]; Decrease by y units

#VALx[y]; Set to y

There can be 50 #VAL's per unit with a maximum value of 32767 (?)

## 18. Loading Flash files as Custom pages

'.swf' files CAN be loaded onto the 'page' area (not the 'image' area). At least that is true with the 1.5 releases (not true of early versions).

In both the 'page' area and 'image' area, attempting to upload an illegal file type (i.e. '.doc') should bring up a dialog window with a warning which includes the list of legal types for that area.

## (CW-1000)

## (UT-500)

### 1. The TEST function.

The TEST function outputs on BOTH comports irrespective of the header assignments. By default COM 1, \01, is placed in the header HOWEVER it is easy to assume you have the correct com port assignment when using the TEST function. When using the TEST function, the strings are sent out both ports irrespective of the header.

### 2. Header assignment.

Watch for Com Port assignment when doing serial translation. By default COM 1, \01, is placed in the header however it may have to be changed to COM 2, \02, for correct operation under RUN mode.

### 3. String Triggers

There is an (8) character limit on a serial trigger string, the string can be 32 char long, HOWEVER it searches for an 8 char match within that string.

e.g The string !KEY\_VOL+\$0D is too long but you could search for just VOL+ and use that to trigger off.

4.

5.

## (CA series)

### 1. Power cycling

To exit "PROG" mode back to "RUN" mode, we are able to power cycle the units remotely using #PGM0 command

### 2. Channel description

All channels are stereo (ch1, ch2, ch3) except channel4 (mono). Channel4 has a permanent jumper linking Left and right side. Ch1 thru Ch3 have separate left and right stereo lines through all stages of AMP (or Pre-AMP) in and out.

### 3. Mic Level Scanning

When the units monitor for CHx microphone input levels – it is only the LEFT side – hence if you have a single channel microphone - make sure it is connected to the LEFT side input.

4.

5.