

PrecisionHD 4X Camera Connectivity Problems

Recently camera connectivity and camera boot problems with PrecisionHD 4X cameras (CTS-PHD-1080P4XS1=) have been reported in the field. We have narrowed down the reported problems to three scenarios:

1. Failed Camera Boot – Failed Initialization (solution available); please see the following Field Notice for more information: <http://www.cisco.com/en/US/partner/ts/fn/635/fn63534.html>
2. Failed Camera Boot – Continuous Reboot (Preventable but once failed requires RMA)
3. Wake from Standby (solution and workaround available)

The only scenario above where a replacement is required is scenario 2, as the camera cannot be returned to a working state. All other scenarios have a solution that will restore full camera function. Root cause has been identified for all issues and are listed below.

IMPORTANT NOTES REGARDING FAILED CAMERA BOOT – FAILED INITIALIZATION:

1. Once a PrecisionHD 4X camera displays symptoms of a failed camera initialization (details listed under “Failed Camera Boot – Failed Initialization”) the camera must be recovered using either recovery procedure listed in this document. Corrected camera firmware is found in TC4.2.4 and TC5.1.3 and higher; however, merely upgrading to a higher version will not correct a camera already in this state.
2. Starting with TC5.1.5, a recovery method for cameras in this failed initialization state can be run from the command line of the codec; this is the recommended recovery method. A second, longer recovery method is also provided in this document.
3. **If a PrecisionHD 4X camera is currently not displaying any symptoms of failed initialization and otherwise operating normally, Cisco recommends upgrading all codecs with a CTS-PHD-10804XS1 connected to a software release containing fixed camera firmware (TC4.2.4 or TC5.1.3 or higher).**
4. A failed camera initialization can potentially occur at any time; if symptoms of scenario 1 are displayed on a previously working camera, the camera must be manually recovered using one of the recovery procedures found in this document.

TC4.2.4 and TC5.1.X software (CCO login required):

<http://www.cisco.com/cisco/software/release.html?mdfid=283613658&flowid=22765&softwareid=280886992&release=TC5.1.5&relind=AVAILABLE&rellifecycle=&reltype=latest>

Field Notice:

[FN - 63534 - PrecisionHD 1080p 4x Camera Does Not Boot up Due to Checksum Validation Failure - S/W Upgrade Recommended](#)

1. Failed Camera Boot – Failed Initialization

Customers are reporting the following symptoms on new installs (new out of box), new replacements (RMA'd units out of box), or after rebooting their codec on a previously working installation for certain Precision HD 4X cameras:

1. Orange LED present on camera at all times.

2. Green LED present on base of camera.
3. No PTZ control from any input device (e.g. touchpanel, remote, etc.)
4. No video on self-view from camera.
5. The admin CLI command "xstatus camera 1" shows "Connected: False" and no other camera information is displayed.

```
xstatus camera 1
*s Camera 1 Connected: False
*s Camera 1 HardwareID: ""
*s Camera 1 Manufacturer: ""
*s Camera 1 Model: ""
*s Camera 1 SoftwareID: ""
*s Camera 1 SerialNumber: ""
*s Camera 1 IPAddress: ""
*s Camera 1 MacAddress: ""
*s Camera 1 Position Pan: 0
*s Camera 1 Position Tilt: 0
*s Camera 1 Position Zoom: 16
*s Camera 1 Position Focus: 4000
*s Camera 1 Capabilities Options: ""
*s Camera 1 Flip: "Off"
** end
OK
```

6. Power cycling the camera or codec does not allow codec control of the camera.

Details regarding the above problem are documented in the following bug:

CSCua44699 Camera fails to boot due to checksum validation failure

As a result of the above, the camera cannot be controlled by the codec.

NOTE: The issue described above (failed initialization) is different than the Continuous Reboot issue mentioned below. When the camera LED is solid orange, the camera in most cases can and should be recovered to working order by upgrading to TC5.1.5 and using "systemtools camerarescue" from the CLI (page 4), or using the "Camera Recovery Procedures" (page 7) on the existing TC software version.

Root Cause:

The root cause has been found by our camera team and corrected camera firmware is found in TC4.2.4 and TC5.1.3 and higher; corrected camera firmware is found in those versions. In summary, a bug was found in our error correction; the new NAND flash found in recent cameras, being a different technology, is more likely to have correctable errors, and made the bug much more obvious.

Solution:

PREVENT - Upgrade camera firmware PRIOR to getting into this failed state:

Cisco recommends upgrading all codecs with a CTS-PHD-10804XS1 connected to a software release containing fixed camera firmware (TC4.2.4 or TC 5.1.3 or higher).

Upgrading the codec software to a fixed release upgrades the camera firmware automatically and prevents the camera from failing. Additionally, the camera firmware will not be downgraded if the codec is downgraded to a lower software version, hence it is possible to prevent cameras from failing by:

- Upgrading to a new software version containing fixed camera firmware
- Wait for the camera to finish upgrading firmware (approximately 8 minutes)
- Then downgrade to a lower version if required.

The camera firmware version containing the fix is found in 20011 and later. The current camera firmware version is found by viewing the softwareID as shown in “xstatus camera” (screenshot below). Fixed camera firmware is bundled with TC4.2.4 and TC5.1.3 or later codec software.

Upgrading cameras currently not exhibiting this failure is **MANDATORY** to prevent any future failure.

Once the camera has been successfully been upgraded, a similar output as shown here should be seen when issuing the “xstatus camera” command:

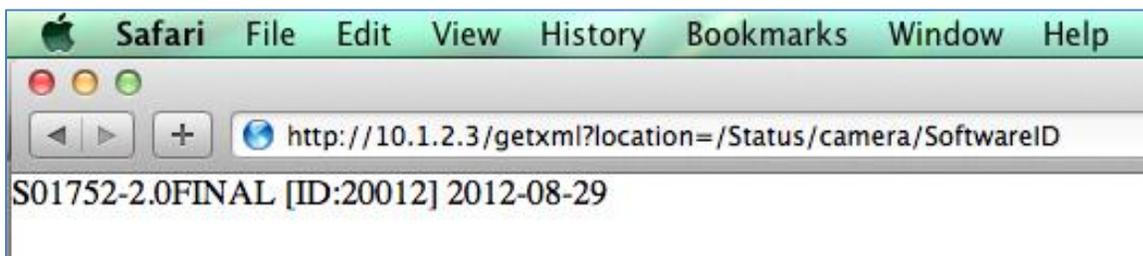
```
bash-3.2$ ssh admin@10.1.2.3
Welcome to
TANDBERG Codec Release TC5.1.5
SW Release Date: 2012-10-24, 12:55, m

OK

xstatus camera 1
*s Camera 1 Connected: True
*s Camera 1 HardwareID: "52000000"
*s Camera 1 Manufacturer: "TANDBERG"
*s Camera 1 Model: "PrecisionHD 1080p 4X"
*s Camera 1 SoftwareID: "S01752-2.0FINAL [ID:20012] 2012-08-29"
```

Similar camera software version output can be seen through the web UI by using this URL:

<http://<IP Address of the codec>/getxml?location=/Status/camera/Version/SoftwareID>



RECOVER – Fixing the Failure Once in a Failed Boot State

If the camera is currently in a failed initialization state (where the orange LED is on solid), the camera firmware must be reinstalled. Upgrading to a higher TC software version will not correct a camera already in this state.

Two recovery procedures exist:

1. Upgrade to TC 5.1.5 and use “systemtools camerarescue” from the admin CLI;
2. Perform the manual recovery procedures listed under “Camera Firmware Recovery Procedures Using a PC” on page 7.

Starting with TC 5.1.5, a recovery method for cameras in this failed boot state is can be run from the codec. This recovery method can be run remotely with the camera connected as normal to the codec. This new recovery method is initiated from the codec with the admin shell command “systemtools camerarescue”, and is the recommended recovery method. Please see the TC 5.1 Release Notes for more information:

http://www.cisco.com/en/US/docs/telepresence/endpoint/software/tc5/release_notes/tc_software_release_notes_tc5.pdf

The “systemtools camerarescue” recovery method will take roughly 10 minutes to complete. When complete, the codec and the camera should be fully operational.

Below is an example output of the shell “systemtools camerarescue” API command:

```
bash-3.2$ ssh admin@10.1.2.3
Welcome to
TANDBERG Codec Release TC5.1.5
SW Release Date: 2012-10-24, 12:55, m

OK

systemtools camerarescue ?
usage: camerarescue
  This command will put the codec into maintenance and try to upgrade the
  the camera using a low-level serial interface. The codec will then be
  rebooted automatically.

OK
systemtools camerarescue
Shutting down to run camera-rescue script and reboot

OK
Unknown message type: 4
tshell: Remote closed connection
Connection to 10.1.2.3 closed.
bash-3.2$
```

The progress of the script can be monitored if root access is enabled on the codec. Once the script is started using the “systemtools camerarescue” command, running the root shell command “tail -f /var/log/camera-rescue.log” allows progress to be monitored. This step is not necessary for the script to function, merely to monitor progress of the recovery.

If the recovery script fails to recover the camera, details of why the script failed are found in the historical log bundle in the camera-rescue.log file. Also, make sure the camera is connected to the serial port of the codec before running the shell command to recover the camera. If the fault cannot be isolated, please work with the TAC to determine the cause of the failure.

Please see the section “**Important Notes Regarding Failed Camera Boot – Failed Initialization**” on page 1 before proceeding.

2. Failed Camera Boot – Continuous Reboot

Customers are also reporting the following symptoms on new installs (new out of box), new replacements (RMA'd out of box), or after rebooting the codec on a previously working installation for certain Precision HD 4X cameras:

1. Orange LED on camera blinks on/off at a steady rate.

2. Green LED present on base of camera.
3. No PTZ control from any input device (e.g. touchpanel, remote, etc.)
4. No video on self-view from camera.
5. The admin CLI command "xstatus camera 1" shows "Connected: False" and no other camera information is displayed.

```
xstatus camera 1
*s Camera 1 Connected: False
*s Camera 1 HardwareID: ""
*s Camera 1 Manufacturer: ""
*s Camera 1 Model: ""
*s Camera 1 SoftwareID: ""
*s Camera 1 SerialNumber: ""
*s Camera 1 IpAddress: ""
*s Camera 1 MacAddress: ""
*s Camera 1 Position Pan: 0
*s Camera 1 Position Tilt: 0
*s Camera 1 Position Zoom: 16
*s Camera 1 Position Focus: 4000
*s Camera 1 Capabilities Options: ""
*s Camera 1 Flip: "Off"
** end
OK
```

6. Power cycling the camera or codec does not allow codec control of the camera.

Details regarding the above problem are documented in the following bug:

CSCua51146 Header checksum mismatch for factory image of second stage bootloaderUBL

The camera is constantly rebooting itself due to a header checksum mismatch on the factory image during bootloader initialization. As a result of the above, the camera cannot be controlled by the codec.

Root Cause:

The root cause has been found by our camera team and is fixed in TC 4.2.4 and TC 5.1.3 and higher, both of which contain corrected camera firmware. In summary, a bug was found in our error correction; the new NAND flash found in recent cameras, being a different technology, is more likely to have correctable errors, and made the bug

much more obvious. The root cause is the same as with the failed camera initialization, but with different symptoms and corrective actions.

Solution: **There is no workaround once the camera gets into the above state and replacement should be considered. Upgrading to TC 4.2.4 or TC 5.1.3 or higher will not correct a camera that is continuously rebooting.**

Corrective Actions for Failed Camera Boot – Continuous Reboot:

With a serial connection to the camera (either through the camera-rescue.log file from the new recovery method or as described in “*Serial Recovery Procedure*” on page 7) TAC is able to verify the signature of a continuously rebooting camera. Additionally, if the orange LED at the top of the camera blinks on/off at a steady rate with complete loss of camera control, that is a good indication the camera is continuously rebooting. Each flash of the orange LED is an attempt by the camera to boot its firmware image, which repeatedly fails.

Unfortunately, there are no corrective actions to bring the camera back to a working state. **There is no workaround once the camera gets into the above state and replacement should be considered. Upgrading to TC 4.2.4 or TC 5.1.3 or higher will not correct a camera that is continuously rebooting.**

Please see the section “**Important Notes Regarding Failed Camera Boot – Failed Initialization**” on page 1 for cameras that are currently in a working state but are potentially affected per the Field Notice.

3. Wake from Standby

Certain Precision HD 4X cameras may also display the following after working for a period of time:

1. System is configured for standby;
2. System goes into standby;
3. When system resumes from standby, camera cannot be controlled by the codec - no PTZ control from any input device no video on self-view.

Details are documented in the following bug:

CSCts05788 PrecisionHD 1080p 4x Camera losing connectivity after going to standby

Root cause:

A high peak current consumption when the camera wakes up from standby or when powered on causes the camera not to be detected by the codec.

Cisco has detected that some PrecisionHD 1080p 4x cameras lose connectivity to the connected codec after a reboot or when coming out of standby. The problem has been identified and fixed and a new camera software revision (20010) is included in TC4.2.3 and TC5.1.2 and higher. The TC software upgrade will automatically update the camera with the newest software.

Solution: Fixed in TC4.2.4 and TC5.1.2 and higher. The camera firmware 20010 will lower the maximum current consumption requirement. TC4.2.4 and TC5.1.3 and higher also contains fixed camera firmware 20011 or higher for the two other issues listed in this document.

It is strongly recommended to upgrade to either TC4.2.4 or TC5.1.3 or higher to prevent this failure from occurring.

Workaround:

- Rebooting the camera, or rebooting the entire system with the camera attached, may allow the camera to reestablish a connection.
- Turning off standby on the system may prevent the issue from recurring.
- Powering the camera separately with its own power supply will also prevent the issue (PSU-CAM-V=).
- Connecting the camera to a codec running TC4.2.4 or TC5.1.3 or higher should upgrade the camera version, which also contains a fix for this problem. Faster booting codecs such as C40, C60 or C90 are more likely to successfully upgrade a camera with the older firmware.

Corrective Actions for Wake from Standby:

The permanent fix is to upgrade to TC 4.2.4 or TC 5.1.2 and higher, which contains fixed camera firmware.

NOTE: It is recommended to upgrade to TC 4.2.4 or TC 5.1.3 or higher, depending on your currently installed release, as it contains further updated camera firmware that corrects the other two issues listed in this document.

TC4.2.4 and TC5.1.X software (CCO login required):

<http://www.cisco.com/cisco/software/release.html?mdfid=283613658&flowid=22765&softwareid=280886992&release=TC5.1.5&releind=AVAILABLE&rellifecycle=&reltype=latest>

Camera Firmware Recovery Procedures

Starting with TC5.1.5, a recovery method for cameras in this failed initialization state can be run from the command line of the codec; this is the recommended recovery method.

The previous recovery method documented here still works but requires additional steps and configuration when compared to the newer codec “systemtools camerarecue” method.

The rest of this document pertains to the previous recovery method; this should not be used if you have upgraded to TC5.1.5. Again, the “systemtools camerarecue” command in TC5.1.5 has superseded the method below.

In order to recover the camera image using a PC, we need a direct connection to the camera via the serial port from the PC. There are two ways to apply an updated firmware image to the PrecisionHD 4X cameras:

1. **Serial recovery procedure – If upgrading to TC5.1.5 and using “systemtools camerarecue” was not chosen as a recovery method, then the serial recovery procedure should be used,** and can be run on any PC operating system (Windows XP or Windows 7). The only connection required is a serial connection directly to the camera, along with the camera firmware image and terminal program.
2. **USB recovery procedure – this method is only recommend as a last resort;** it can ONLY be used on Windows XP operating system, as it requires additional USB drivers.
 - **NOTE: The USB drivers do not work on Windows 7. If you install the USB drivers on a Windows 7 machine, it will disable all other USB, NIC, and modem devices until the USB drivers are uninstalled and the PC rebooted.**

In order to perform either recovery procedure, serial/console access to the camera is required. You must connect the camera directly to your PC and access the unit via a serial connection, no matter which recovery method you

choose. Certain cables and software are required depending upon the recovery procedure you choose (serial or USB).

Serial Recovery Procedure

If upgrading to TC5.1.5 and using “systemtools camerarescue” was not chosen as a recovery method, then the serial recovery procedure should be used.

Required cables

1. VISCA camera control cable (shipped with the C20, DB9 M to RJ45)
2. Null modem cable (DB9 F/F)
3. Serial to USB cable (DB9 M to USB M)
4. External power - 12V/2A - use the C20 power supply if needed, this has been tested and will power up the 4X camera.

The pinouts for the null modem cable are below, if you wish to make one. Please note you must have a null modem cable (DB9 F/F) in order to establish a successful serial connection to the camera. Null modem cables can also be purchased at any large computer outlet.

NULL MODEM CABLE (DB9-DB9). PINOUT AND SIGNALS FOR BUILDING A SERIAL (RS232) NULLMODEM CABLE
Use this cable between two **DTE** (Data Terminal Equipment) devices (for instance two computers).

	DB9-1	DB-2	
Receive Data	2	3	Transmit Data
Transmit Data	3	2	Receive Data
Data Terminal Ready	4	6+1	Data Set Ready + Carrier Detect
System Ground	5	5	System Ground
Data Set Ready + Carrier Detect	6+1	4	Data Terminal Ready
Request to Send	7	8	Clear to Send
Clear to Send	8	7	Request to Send

Note: DSR & CD are jumpered to fool the programs to think that they are online.

Please note: the camera needs to be powered by means of an external power supply - either use the power supply from the C20 codec, or use the external 12V/2A power supply for the 4X camera; part number PSU-CAM-V=. See also http://www.cisco.com/en/US/prod/collateral/ps7060/ps11307/ps11335/data_sheet_c78-669051.html

Required Software

1. Serial_shell - Terminal emulation program (Cisco-provided). Only this terminal emulation program will allow for asynchronous data transfer of the upgrade file.
2. Sangam.zip - Updated camera firmware ID 20011 (Cisco-provided) and file upgrade binary.

Please download the software at either of the following links:

<https://supportforums.cisco.com/servlet/JiveServlet/downloadBody/25127-102-4-62751/SerialRecoverySoftware.zip>

<https://supportforums.cisco.com/docs/DOC-25127>

Cisco can also provide the software on an as-needed basis.

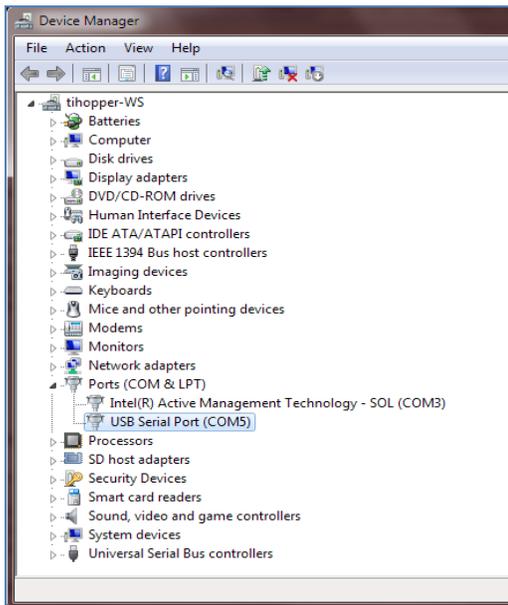
Connecting to Serial Port of Camera

1. Connect to your Windows 7 or Windows XP PC via null modem/serial cable to the camera. Connect the cables as follows:

Camera control RJ45 port --> Camera control DB9 M port --> DB9 F/F Null modem cable --> USB to serial DB9 M --> PC USB port

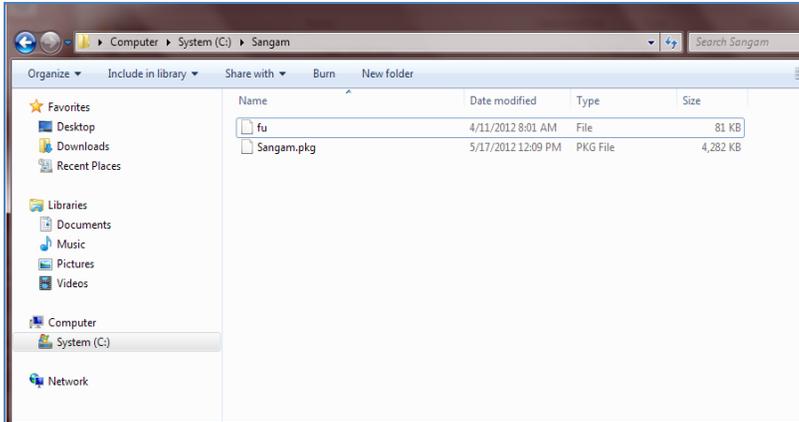
*****Do not apply power to the camera, only connect to the camera using the RJ45 connector on the camera console cable. You will power up the camera later on in this procedure.**

Once the cables are connected, you must determine the COM port number assigned by the system. The COM port number can be determined by opening up Device Manager and expanding "Ports (COM & LPT)" and checking which COM has been assigned to your USB to Serial connection. You will need the COM port number to connect using the terminal emulation program. In the below example, the system assigned COM5 - please note that your COM port number may be different than the one shown in the screenshot below:

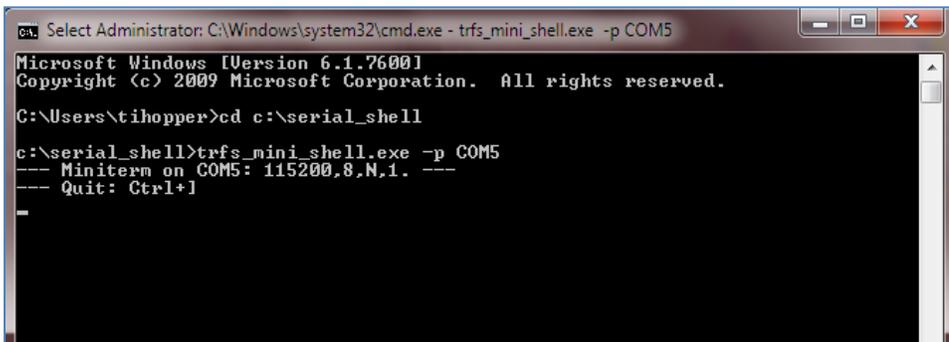


Installing and Running the Required Software on Windows 7

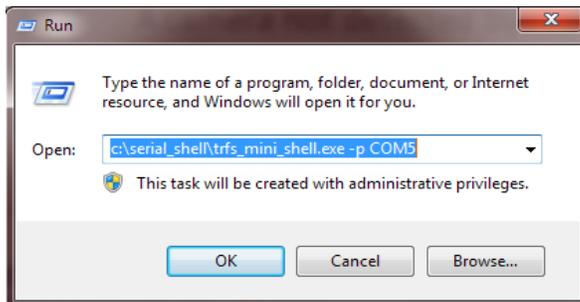
1. Extract serial_shell.zip to C:\ on your Windows 7 PC. Be sure the extraction does not create an additional serial_shell folder within the original extracted folder (e.g. serial_shell folder inside another serial_shell folder – we need one serial_shell folder with the needed files inside).
2. Create a directory called "Sangam" under root of C:\. Extract the two files in Sangam.zip to that directory; the directory should contain files named "fu" and "Sangam.pkg," as shown below:



3. Open up a command prompt (Start → Run → cmd).
4. Change directory to C:\serial_shell (cd C:\serial_shell).
5. To run the program, type trfs_mini_shell.exe -p COMX, where “X” is the number of your COM port (screenshot below).



6. Alternatively, you can open the terminal emulation program by going to Start → Run and typing in “c:\serial_shell\trfs_mini_shell.exe -p COMX” where X is the number of your COM port:



Verify Failed Camera Boot – Failed Initialization

Once all cables are connected, software is installed, and the connection to the COM port is established using the terminal emulator, apply power to the camera by either a) plugging in the C20 power cable; b) using a 12V/2A power supply with appropriate barrel connector, or c) use the Cisco power cable for the 4X cameras if available

(part number PSU-CAM-V=; see also http://www.cisco.com/en/US/prod/collateral/ps7060/ps11307/ps11335/data_sheet_c78-669051.html).

A camera not detected by the codec, i.e. "xstatus camera 1" shows "Connected: false", will show something similar to the below error on a failed initialization when using a serial connection to the camera - note the "Checksum validation fails" lines present. Keep in mind there can be variations of the below, but the camera will always fail to initialize. When the camera fails to initialize, **the base LED will be solid green, and the camera LED will be solid amber/orange:**

```
DM365 chip initialization done
Board id : 00000001 Version id : 00000006
version 5
Configuring FPGA
Done
boot_delay : 3
boot_delay : 2
boot_delay : 1
boot_delay : 0
version1 = 3308 version2 = 3628
Checksum validation fails
Checksum validation fails
Looking for factory image....
Checksum validation fails
No valid Uimage presentSANGAM#
```

If you encounter the above, then please proceed on to the next section *"Recovering the Camera Software"* [below](#).

However, if you encounter the output listed below at the serial console, then the camera is constantly rebooting – **the base LED will remain solid green, but the orange LED on the camera will be flashing on/off at a steady rate:**

```
DM365 chip initialization done
Board id : 00000001 Version id : 00000006

Possible file checksum failure for shell read value =
Header checksum mismatch for factory image of second stage bootloaderUBL stage3:
Nand boot failed
Undefined mode

VPSS Reset Done
```

If the above is seen from the serial console of the camera on initial boot, the camera software cannot be recovered and replacement is the only option.

Recovering the Camera Software

Once the camera is powered on a failed initialization has been identified, upgrade the camera software using the following commands:

1. On a failed boot, you will automatically boot up to the SANGAM# prompt. Once there, run:
/s0/fu /s0/Sangam.pkg -d
2. You should see the following indicator showing the software being transferred:
82481 bytes transfered 8 secs
\
-

3. After approximately 8 minutes, the camera should be upgraded; there will be upgrade and validation information displayed on the terminal.
4. Type "reset" at the #SANGAM prompt to reboot the camera.
5. Camera should now boot properly; you will know if the camera boots properly if a) you see no checksum validation failures and b) you see "Booting Linux..." with no further text, as shown below:

```
DM365 chip initialization done  
Board id : 00000001 Version id : 00000006
```

```
version 5  
Configuring FPGA  
Done  
boot_delay : 3  
boot_delay : 2  
boot_delay : 1  
boot_delay : 0
```

```
Booting Linux.....
```

USB Recovery Procedure

This method is NOT the preferred method, and should ONLY be used as a last resort!

- **NOTE: The USB drivers do not work on Windows 7. If you install the USB drivers on a Windows 7 machine, it will disable all other USB, NIC, and modem devices until the USB drivers are uninstalled and the PC rebooted.**

Required cables

1. VISCA camera control cable (shipped with the C20, DB9 M to RJ45)
2. Null modem cable (DB9 F/F)
3. Serial to USB cable (DB9 M to USB M)
4. External power - 12V/2A - use the C20 power supply if needed, this has been tested and will power up the 4X camera.
5. USB A male to micro USB B

The pinouts for the null modem cable are below, if you wish to make one. They can be purchased at any large computer outlet as well.

NULL MODEM CABLE (DB9-DB9). PINOUT AND SIGNALS FOR BUILDING A SERIAL (RS232) NULLMODEM CABLE

Use this cable between two **DTE** (Data Terminal Equipment) devices (for instance two computers).



	DB9-1	DB-2	
Receive Data	2	3	Transmit Data
Transmit Data	3	2	Receive Data
Data Terminal Ready	4	6+1	Data Set Ready + Carrier Detect
System Ground	5	5	System Ground
Data Set Ready + Carrier Detect	6+1	4	Data Terminal Ready
Request to Send	7	8	Clear to Send
Clear to Send	8	7	Request to Send

Note: DSR & CD are jumpered to fool the programs to think that they are online.

Here is what the USB cable looks like; the USB cable needs to have USB A (male) connector, and 1 micro USB B connector.



Please remember: the camera needs to be powered by means of an external power supply - either use the power supply from the C20 codec, or use the external 12V/2A power supply for the 4X camera; part number PSU-CAM-V=. See also http://www.cisco.com/en/US/prod/collateral/ps7060/ps11307/ps11335/data_sheet_c78-669051.html

Required Software

- Terminal Emulation program - To access the camera via the serial port one needs to have a terminal emulation program which can talk to a serial COM port set to 115200 bps, 8 bits, 1 stopbit and no parity. PuTTY, which works well, can be downloaded at <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

- For the USB control we need to install LibUSB-Win32 drivers on a PC. Along with LibUSB-Win32, we need a program named tb_usb. The tb_usb program provides a USB shell to upgrade the camera firmware; this cannot be done via the camera console cable.

NOTE: The LibUSB-Win32 drivers cannot be used on any PC running Windows 7 32-bit or 64-bit OS.

After installing LibUSB-Win32, the installed drivers will disable all other USB, NIC, and modem devices until the drivers are uninstalled and the PC rebooted. Windows XP installed on a PC and Windows XP on a VMWare virtual machine has been tested in our environment successfully.

- A copy of the camera software named Sangam.pkg is also required.
- The required files needed are:
 - Libusb-win32-filter-bin-0.1.12.1.zip
 - Usb_shell.zip
 - Sangam.zip

The zipped file can be downloaded at either of the following links:

<https://supportforums.cisco.com/servlet/JiveServlet/downloadBody/25088-102-4-62752/4XUSBRecoveryFiles.zip>

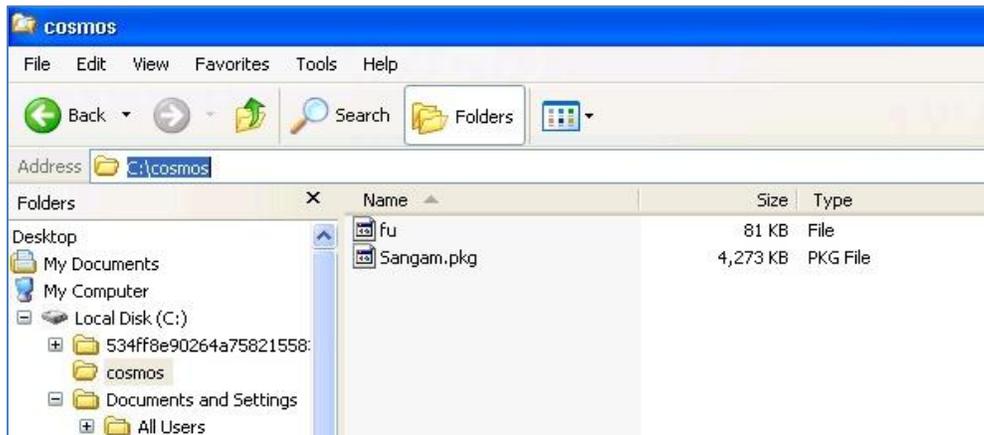
<https://supportforums.cisco.com/docs/DOC-25088>

Cisco can provide a zipped file containing all of the above on an as-needed basis.

Installing the Required Software on Windows XP

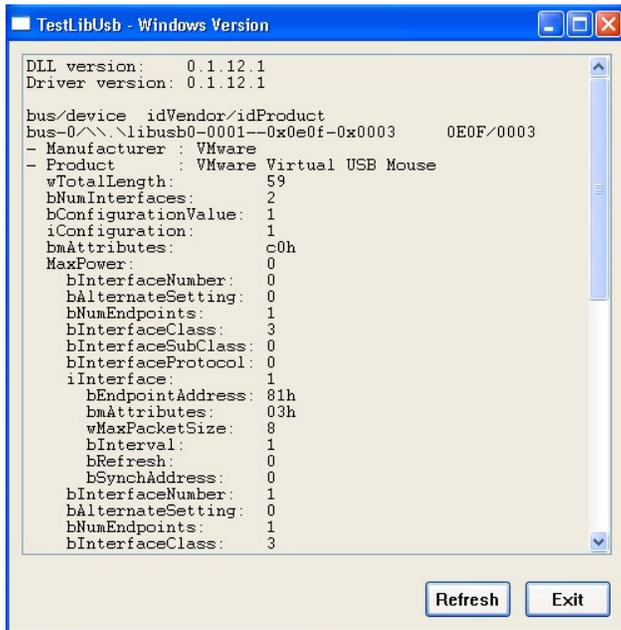
Again, the USB drivers only be used on a Windows XP computer or Windows XP on VMWare. The USB recovery procedure currently does not function using Windows 7.

- Unzip the complete folder name usb_shell to your PC, e.g. under C:\Program Files.
- Create a directory named C:\cosmos and unzip Sangam.zip to that directory. You should have two files under the C:\cosmos directory: fu and Sangam.pkg:



- Install the executable libusb-win32-filter-bin-0.1.12.1.exe onto your PC. When the installation is complete, leave the self-test option checked and click “Finish.” If the self-test is successful you will see

the following screen listing idVendor/idProduct information. **If no idVendor/idProduct information is present, the libUSB program will not be able to transfer the software image via USB.**



```
TestLibUsb - Windows Version
DLL version: 0.1.12.1
Driver version: 0.1.12.1
bus/device idVendor/idProduct
bus-0/\.\libusb0-0001--0x0e0f-0x0003 0E0F/0003
- Manufacturer : VMware
- Product : VMware Virtual USB Mouse
wTotalLength: 59
bNumInterfaces: 2
bConfigurationValue: 1
iConfiguration: 1
bmAttributes: c0h
MaxPower: 0
bInterfaceNumber: 0
bAlternateSetting: 0
bNumEndpoints: 1
bInterfaceClass: 3
bInterfaceSubClass: 0
bInterfaceProtocol: 0
iInterface: 1
bEndpointAddress: 81h
bmAttributes: 03h
wMaxPacketSize: 8
bInterval: 1
bRefresh: 0
bSynchAddress: 0
bInterfaceNumber: 1
bAlternateSetting: 0
bNumEndpoints: 1
bInterfaceClass: 3
```

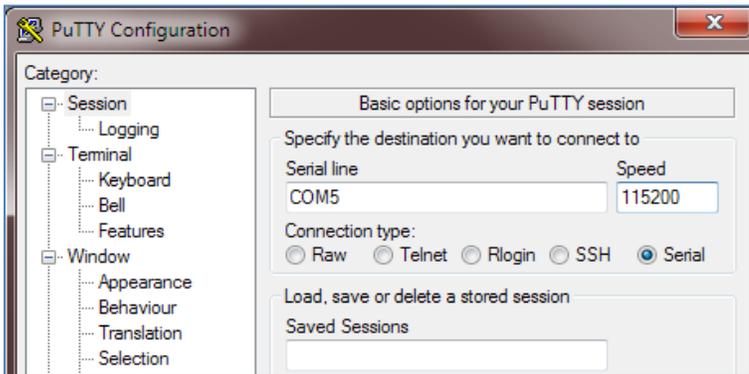
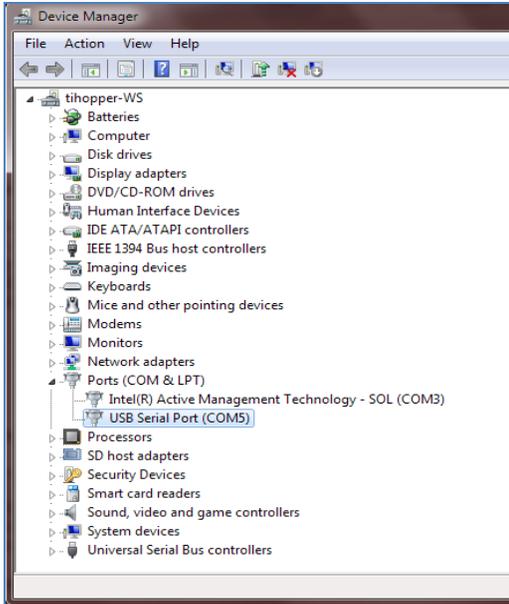
Refresh Exit

Connecting to Serial Port of Camera

Connect PC via null modem/serial cable to the camera. Connect the cables as follows:

Camera control RJ45 port --> Camera control DB9 M port --> DB9 F/F Null modem cable --> USB to serial DB9 M --> PC USB port

Once the cables are connected, open up PuTTY and connect via Serial connection. The COM port number can be determined by opening up Device Manager and expanding "Ports (COM & LPT)" and checking which COM has been assigned to your USB to Serial connection. In the below example, the system assigned COM5 - please note that your COM port number may be different:



Once PuTTY is configured correctly and opened on the COM port, plug in the external power supply to the camera. You should see the boot process of the camera; press the Enter key just after boot, during the "boot_delay" messages below. You should now see the SANGAM prompt.

Verify Failed Camera Boot from Serial Connection

A camera not detected by the codec, i.e. "xstatus camera 1" shows "Connected: false", will show something similar to the below error on initial boot from console - note the "Checksum validation fails" lines present. Keep in mind there can be variations of the below, but the camera will always fail to initialize. When the camera fails to initialize, the base LED will be solid green, and the camera LED will be solid amber/orange:

```
DM365 chip initialization done
Board id : 00000001 Version id : 00000006
version 5
Configuring FPGA
Done
boot_delay : 3
boot_delay : 2
boot_delay : 1
```

```
boot_delay : 0
version1 = 3308 version2 = 3628
Checksum validation fails
Checksum validation fails
Looking for factory image....
Checksum validation fails
No valid Uimage presentSANGAM#
```

3. Once at the SANGAM prompt, enable USB mode by setting the variable USB_mode 1. Be sure to save these settings, as shown below:

```
SANGAM#setenv USB_MODE 1
SANGAM#saveenv
```

Connect to USB port of Camera

1. Start the program tb_usb on your PC. Be sure to start the tb_usb.exe program from the "C:\Program Files" directory.
2. Reboot the camera; you should see the camera startup in the tb_usb window.
3. Press enter in the USB shell window. You should get the same SANGAM prompt as on serial connection, but now in the USB shell.

Recovering the Camera Software

From the SANGAM prompt, execute the command /u0/fu /u0/Sangam.pkg -d which will copy the Sangam.pkg on the camera. Below is the output of a correct upgrade.

```
SANGAM#
SANGAM# /u0/fu /u0/Sangam.pkg -d
-
total file size read: 4375240
Package file size = 4375240 read = 0
Id is correct
Version check passed
Package checksum is correct
File size checksum correct
File table offset = 0000002C
Number of files= 00000005
Optional field len is = 00000012
=====
file table record len 66
Optional len = 0000001C , filelen = 0000001B , Filetab file offset = 0000016C
Tandberg Signature found sucessfully
(1.)File name = /system/upgrade/sid.bin
File checksum for file1 matched
Files checksum = 000003F5 ;
=====
file table record len 67
Optional len = 0000001C , filelen = 0000002B , Filetab file offset = 00000188
Tandberg Signature found sucessfully
(2.)File name = /system/upgrade/sid2.bin
File checksum for file2 matched
```

```
Files checksum = 0000060E ;
=====
file table record len 49
Optional len = 0000001C , filelen = 0001A800 , Filetab file offset = 000001B4
Tandberg Signature found successfully
(3.)File name = bl_tts
File checksum for file3 matched
Files checksum = 00A8613D ;
=====
file table record len 51
Optional len = 0000001C , filelen = 0001C800 , Filetab file offset = 0001A9B4
Tandberg Signature found successfully
(4.)File name = fpga_tts
File checksum for file4 matched
Files checksum = 00519277 ;
=====
file table record len 53
Optional len = 0000001C , filelen = 003F5114 , Filetab file offset = 000371B4
Tandberg Signature found successfully
(5.)File name = ulmage_tts
File checksum for file5 matched
Files checksum = 212E9ADF ;
=====
Pkg is a valid package and its contents are uncorrupted
```

```
Invalid file name for file 0
Invalid file name for file 1
TBL upgrade : src_buf = 0x834FB0B0, sz = 0x0001A800
Min Version = 0, max version = 3748
Current version = 3749 New version = 3749
block_start: 1, block_end: 12
TBL upgrade successful
```

```
FPGA upgrade : src_buf = 0x835158B0, sz = 0x0001C800
Min Version = 0, max version = 2542
Current version = 2543, New version = 2543
block_start: 406, block_end: 445
FPGA Upgrade Successful
```

```
UImage upgrade : src_buf = 0x835320B0, sz = 0x003F5114
Min Version = 0, max version = 3749
New version: 3749 Len total = 4149524
block_start: 600, block_end: 919
block written to: 600
Updating Env... done
UImage Upgrade Successful
```

SANGAM#

After the files have been downloaded completely, we need to turn USB mode off. **Please remember to turn off USB mode while in the USB terminal emulation program.** Once back at the SANGAM prompt, disable USB mode:

```
SANGAM#setenv USB_MODE 0
```

SANGAM#saveenv

Power cycle the camera. USB access is now disabled, so the VISCA serial connection is now active.

Verify using serial access that the camera boots to linux properly. You should no longer see any checksum errors; you should see the camera go straight to "Booting Linux..." as in the below example:

```
DM365 chip initialization done  
Board id : 00000001 Version id : 00000006
```

```
version 5  
Configuring FPGA  
Done  
boot_delay : 3  
boot_delay : 2  
boot_delay : 1  
boot_delay : 0
```

```
Booting Linux.....
```

Once successful camera boot is verified, connect the camera with its newly programmed software to the C-series codec. Verify the camera is successfully connected using "xstatus camera 1" and that it shows a firmware ID of 20011 or higher.