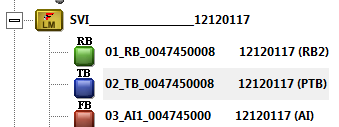
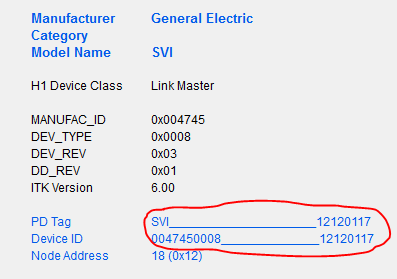
Inter-processor data exchange testing

# Default device identification

Verify that a device starting with R3 firmware for the first time receives block ids, tags, etc. depending on AP CPU “device id”, e.g.



and

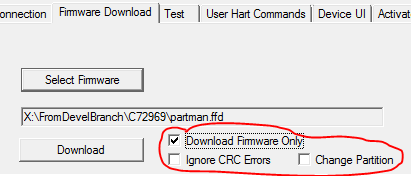


The test setup is different depending on how the device received the R3 firmware

## Fresh device

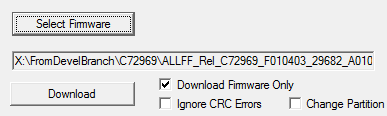
Make sure the **terminal board on the device is capable of receiving firmware download over ISP** connector. Otherwise, the operation will destroy the device CPU, and to recover you will need the compatible terminal board anyway.

Download the partition manager firmware over ISP using Smarts Assistant, e.g.:



Note the options for download.

This will remove the existing device firmware. Now, download the SVI FF firmware, e.g.



Cycle the power and observe how NI Configurator finds the device.

## Update from R2

Upgrade the device with a valid R2 firmware using VV3/DTM:

1. Download the image
2. Activate

## Update from same or different R3

“Upgrade” the device with the same or different build of R3 firmware using VV3/DTM:

1. Download the image
2. Activate

# Performance

The following scenarios should be covered, with or without reading TB parameters with NI Configurator or Dialog or such:

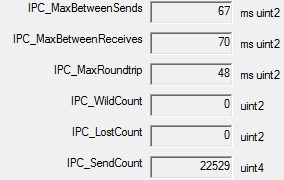
1. Empty FBAP
2. A realistic FBAP
3. Firmware download while a realistic FBAP is running

Use Smarts Assistant User Commands page, command 255.15 and 255.14 to establish baseline. Note that those commands clear previous results, so they should be executed before the tests and once after each test.

## Expected results for 255.15:

1. There are 0 wild messages (receive without corresponding send)
2. There are 0 lost messages (send without corresponding receive)
3. Intervals between consecutive sends and between consecutive receives are close to each other and close in each of the 3 scenarios above
4. Message roundtrip times are close in each of the 3 scenarios above
5. Document the results of #3 and #4 for each scenario as a baseline.

E.g.

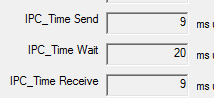


To ensure representative statistics, scenarios 1-2 should run at least 15 min each, and scenario 3, for the full download (but before any activation)

## Expected results for 255.14:

1. Times send/wait/receive should be close across scenarios 1-3
2. Document the results of #1 for each scenario as a baseline.

E.g.



# Error recovery

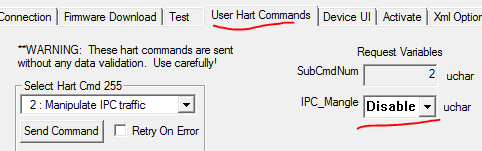
These test cases verify device behavior in case of IPC errors

## Errors in periodic traffic

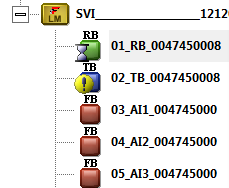
### Common steps

Let NI Configurator synchronize with the device.

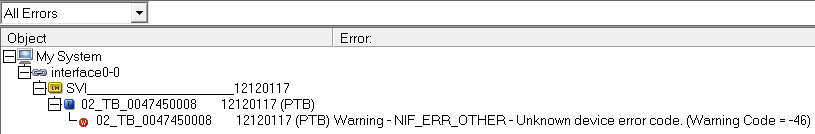
Using SA command 255.2, disable periodic traffic:



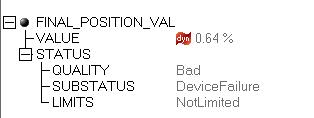
Observe that the device is still online



Observe errors in communications with TB



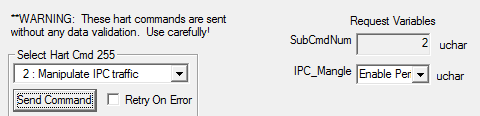
Observe that variables from AP CPU are bad, e.g.



And TB is in OOS

### Emulation of self-recovery

Enable periodic traffic again soon after TB is in OOS:



Observe that communication resumes OK

### Manual recovery

Issue RB.RESTRAT=Processor soon after TB is in OOS.

Even if the command appears to fail, observe that the device recovers.

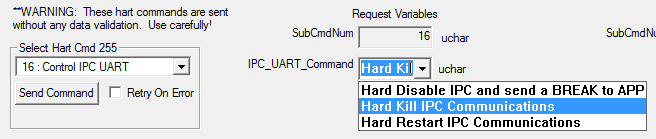
NOTE: After some time, NI loses connection to open blocks, and when device is deleted and rediscovered by Configurator, it has problems connecting. The problem appears to be in NI Communications Manager and Configurator, and they may need to be restarted.

## Failure of IPC

The test emulates complete seizure of IPC and manual recovery.

Have RB and TB in AUTO, APP in Normal.

Using SA, disable all IPC with command 255.16:



Wait a minute or so. Observe that FF communications are generally intact but APP-owned TB parameters are unavailable (show as NULL).

Observe TB mode is OOS and APP in failsafe.

Issue RB.RESTART=Processor.

Observe all parameters in TB updated