

Command Line Functions for Improved Diagnostics

The console commands are a powerful set of command line utilities that are extremely useful when troubleshooting errors and issues with the openPDC.

The **help** command can be used to see a list of all commands that can be entered into the console

Command	Description
-----	-----
Clients	Displays list of clients connected to the service
Settings	Displays queryable service settings from config file
Processes	Displays list of service or system processes
Schedules	Displays list of process schedules defined in the service
History	Displays list of requests received from the clients
Help	Displays list of commands supported by the service
Status	Displays the current service status
Start	Start a service or system process
Abort	Aborts a service or system process
ReloadCryptoCache	Reloads local cryptography cache
UpdateSettings	Updates service setting in the config file
ReloadSettings	Reloads services settings from the config file
Reschedule	Reschedules a process defined in the service
Unschedule	Unschedules a process defined in the service
SaveSchedules	Saves process schedules to the config file
LoadSchedules	Loads process schedules from the config file
Version	Displays current service version
Time	Displays current system time
Health	Displays a report of resource utilization for the service
ResetHealthMonitor	Resets the system resource utilization monitor
List	Displays status for specified adapter or collection
Connect	Connects (or starts) specified adapter
Disconnect	Disconnects (or stops) specified adapter
Invoke	Invokes a command for specified adapter
ListCommands	Displays possible commands for specified adapter
Initialize	Initializes specified adapter or collection
ReloadConfig	Manually reloads the system configuration
UpdateConfigFile	Updates an option in the configuration file
Authenticate	Authenticates network shares for health and status exports
Restart	Attempts to restart the host service
RefreshRoutes	Spawns request to recalculate routing tables
TemporalSupport	Determines if any adapters support temporal processing

Command: List /i

LIST:Success - System Uptime: 7 Hours 17 Minutes 9 Seconds

>> All defined adapters in Input Adapter Collection (4 total)

ID	Name

2	SHELBY Up for 7.28 hours, 0 errors 08/18/2012 02:42:07.433 27.95 fps
8	PPAREADER Not currently publishing data
9	SHELBY2 Up for 7.28 hours, 0 errors 08/18/2012 02:42:07.433 28.15 fps
10	SHELBY3 Up for 19.4 minutes, 0 errors 08/18/2012 02:42:07.433 28.35 fps

Command: List /a

LIST:Success(/a) - System Uptime: 7 Hours 43 Minutes 56 Seconds

>> All defined adapters in Action Adapter Collection (6 total)

ID	Name

0	PHASOR!SERVICES Type "LISTCOMMANDS 0" to enumerate service commands.
3	TESTSTREAM Total input measurements: 39, total output measurements: 0
5	EXTERNALDATAPUBLISHER Publishing data to 0 clients.
6	STATISTIC!SERVICES Currently publishing 95 statistics
7	ALARM!SERVICES 0 events processed since last start
11	PWRCALC Total input measurements: 4, total output measurements: 2

Command: List /o

LIST:Success(/o) - System Uptime: 7 Hours 49 Minutes 55 Seconds

>> All defined adapters in Output Adapter Collection (2 total)

ID	Name

1	PPA
	Archived 24534879 measurements locally
4	STAT
	Archived 267520 measurements locally

ListCommands displays the possible commands for a specified adapter

Example: listcommands 2

LISTCOMMANDS:Success(2) - Adapter "SHELBY" [Type = PhasorMeasurementMapper]
Command List:

SendCommand(TVA.PhasorProtocols.DeviceCommand command)
Sends the specified command to connected phasor device.

ResetStatistics()
Resets the statistics of all devices associated with this connection.

ResetDeviceStatistics(UInt16 idCode)
Resets the statistics of the device with the specified ID code.

DeleteCachedConfiguration()
Attempts to delete the last known good configuration.

RequestCurrentConfiguration()
Requests the current configuration frame and returns it to the caller.

LoadCachedConfiguration()
Attempts to load the last known good configuration.

LoadConfiguration(String configurationFileName)
Attempts to load the specified configuration.

Start()
Starts the adapter or restarts it if it is already running.

Stop()
Stops the adapter.

SetInitializedState(Boolean initialized)
Manually sets the initialized state of the adapter.

SetTemporalConstraint(String startTime, String stopTime, String constraintParameters)

Defines a temporal processing constraint for the adapter.

Example: listcommands 3

LISTCOMMANDS:Success(3) - Adapter "TESTSTREAM" [Type = Concentrator] Command List:

Start()

Starts the action adapter or restarts it if it is already running.

Stop()

Stops the action adapter.

StartDataChannel()

Manually starts the real-time data stream.

StopDataChannel()

Manually stops the real-time data stream.

UpdateConfiguration()

Reloads the phasor data concentrator configuration.

ExamineQueueState()

Examines concentration frame queue state.

ResetStatistics()

Resets the statistics of the action adapter.

SetInitializedState(Boolean initialized)

Manually sets the initialized state of the action adapter.

SetTemporalConstraint(String startTime, String stopTime, String constraintParameters)

Defines a temporal processing constraint for the adapter.

Application of Console Commands:

Using the **List /a** command, we get the ID of the concentrator

ID	Name	Status
3	TESTSTREAM	Total input measurements: 39, total output measurements: 0

We can also see there are 39 input measurements, 13 from each of the input adapters

Next, we use the **ExamineQueueState** command to evaluate the concentrator queue details with the **invoke** command

Invoke 3 ExamineQueueState
(console view)



The results return the following:

INVOKE:Success(3 examinequeuestate) - Command "examinequeuestate" successfully invoked.
[8/17/2012 00:00:00 PM] [TESTSTREAM] Concentrator frame queue detail:

Ordered frame queue count: 89
Frame hashtable count: 89

This shows that only
26 of 39 measurements
were being processed

Frame 0000 @ 17-Aug-2012 02:18:27.566 - 26 measurements, 66.67% received
Frame 0001 @ 17-Aug-2012 02:18:27.600 - 26 measurements, 66.67% received
Frame 0002 @ 17-Aug-2012 02:18:27.633 - 26 measurements, 66.67% received
Frame 0003 @ 17-Aug-2012 02:18:27.666 - 26 measurements, 66.67% received
Frame 0004 @ 17-Aug-2012 02:18:27.700 - 26 measurements, 66.67% received
Frame 0005 @ 17-Aug-2012 02:18:27.733 - 26 measurements, 66.67% received
Frame 0006 @ 17-Aug-2012 02:18:27.766 - 26 measurements, 66.67% received
Frame 0007 @ 17-Aug-2012 02:18:27.800 - 26 measurements, 66.67% received
Frame 0008 @ 17-Aug-2012 02:18:27.833 - 26 measurements, 66.67% received
Frame 0009 @ 17-Aug-2012 02:18:27.866 - 26 measurements, 66.67% received
Frame 0010 @ 17-Aug-2012 02:18:27.900 - 26 measurements, 66.67% received
Frame 0011 @ 17-Aug-2012 02:18:27.933 - 26 measurements, 66.67% received
Frame 0012 @ 17-Aug-2012 02:18:27.966 - 26 measurements, 66.67% received
Frame 0013 @ 17-Aug-2012 02:18:28.000 - 26 measurements, 66.67% received
Frame 0014 @ 17-Aug-2012 02:18:28.033 - 26 measurements, 66.67% received
Frame 0015 @ 17-Aug-2012 02:18:28.066 - 26 measurements, 66.67% received

The problem was determined to be a measurement mapping error found in an input adapter:

[8/17/2012 3:24:58 PM] [SHELBY2] Loaded 34 active device measurements...
[8/17/2012 3:24:58 PM] [SHELBY] Loaded 34 active device measurements...
[8/17/2012 3:24:58 PM] [SHELBY3] Loaded **24** active device measurements...

After correcting the problem with measurement mapping, use the “**examinequeuestate**” command to review the output:

**Invoke 3 ExamineQueueState
(console view)**



The results return the following:

INVOKE:Success(3 examinequeuestate) - Command "examinequeuestate" successfully invoked.
[8/17/2012 00:00:00 AM] [TESTSTREAM] Concentrator frame queue detail:

Ordered frame queue count: 90

Frame hashtable count: 90

Frame 0000 @ 17-Aug-2012 14:57:50.433 - 39 measurements, 100.00% received
Frame 0001 @ 17-Aug-2012 14:57:50.466 - 39 measurements, 100.00% received
Frame 0002 @ 17-Aug-2012 14:57:50.500 - 39 measurements, 100.00% received
Frame 0003 @ 17-Aug-2012 14:57:50.533 - 39 measurements, 100.00% received
Frame 0004 @ 17-Aug-2012 14:57:50.566 - 39 measurements, 100.00% received
Frame 0005 @ 17-Aug-2012 14:57:50.600 - 39 measurements, 100.00% received
Frame 0006 @ 17-Aug-2012 14:57:50.633 - 39 measurements, 100.00% received
Frame 0007 @ 17-Aug-2012 14:57:50.666 - 39 measurements, 100.00% received
Frame 0008 @ 17-Aug-2012 14:57:50.700 - 39 measurements, 100.00% received
Frame 0009 @ 17-Aug-2012 14:57:50.733 - 39 measurements, 100.00% received
Frame 0010 @ 17-Aug-2012 14:57:50.766 - 39 measurements, 100.00% received
Frame 0011 @ 17-Aug-2012 14:57:50.800 - 39 measurements, 100.00% received
Frame 0012 @ 17-Aug-2012 14:57:50.833 - 39 measurements, 100.00% received
Frame 0013 @ 17-Aug-2012 14:57:50.900 - 39 measurements, 100.00% received
Frame 0014 @ 17-Aug-2012 14:57:50.933 - 39 measurements, 100.00% received

We can now see that all of the measurements are being processed

Conclusions:

- The console commands provide a quick and convenient way to analyze the system
- Just to name a few:
 - list of adapters (the list command)
 - view low level data (the status command)
 - concentration frame queue state (the examinequeuestate command)
 - Invokes a command for specified adapter (invoke command)

A few example screen shots

```
C:\openPDC\openPDCConsole.exe
list
System Uptime: 1 Hour 12 Minutes 9 Seconds
>> All defined adapters in Input Adapter Collection <4 total>

  ID      Name
-----
  2      SHELBY
        Up for 1.20 hours, 0 errors 08/18/2012 05:42:39.766 24.16 fps
  8      PPAREADER
        Not currently publishing data
  9      SHELBY2
        Up for 1.20 hours, 0 errors 08/18/2012 05:42:39.766 24.16 fps
  10     SHELBY3
        Up for 1.20 hours, 0 errors 08/18/2012 05:42:39.766 24.16 fps
```

```
C:\openPDC\openPDCConsole.exe
list /a
System Uptime: 1 Hour 13 Minutes 43 Seconds
>> All defined adapters in Action Adapter Collection <6 total>

  ID      Name
-----
  0      PHASOR!SERVICES
        Type "LISTCOMMANDS 0" to enumerate service commands.
  3      TESTSTREAM
        Total input measurements: 39, total output measurements: 0
  5      EXTERNAL!DATAPUBLISHER
        Publishing data to 0 clients.
  6      STATISTIC!SERVICES
        Currently publishing 95 statistics
  7      ALARM!SERVICES
        0 events processed since last start
  11     PWRCALC
        Total input measurements: 4, total output measurements: 2
```

ABOUT THE AUTHOR

Steve DeGange is a project manager at GPA who has direct utility experience in the installation and maintenance of the openPDC.
