EC-Net-AX-3.x* Andover Infinity Driver Guide

User Guide

DISTECH
CONTROLS™

Innovative Solutions for Greener Buildings™

Document Revision History:

Version 1.0 - Initial Release – November 2011

EC-Net-AX-3.x∗ Andover Infinity Driver Guide - User Guide v1.0

05DI-UGANDIN-10

©, Distech Controls Inc., 2011. All rights reserved.

First printing – November 2011. Printed in Canada.

While all efforts have been made to verify the accuracy of information in this manual, Distech Controls is not responsible for damages or claims arising from the use of this manual. Persons using this manual are assumed to be trained HVAC professionals and are responsible for using the correct wiring procedures, correct override methods for equipment control and maintaining safe working conditions in fail-safe environments. Distech Controls reserves the right to change, delete or add to the information in this manual at any time without notice.

Distech Controls, the Distech Controls logo, and Innovative Solutions for Greener Buildings are trademarks of Distech Controls, Inc.; Niagara^{AX} Framework and Niagara Framework are registered trademarks of Tridium, Inc.; BACnet is a registered trademark of ASHRAE; All other trademarks are property of their respective owners; All specifications subject to change without notice or liability to provide changes to prior purchasers. Information and specifications published here are current as of the date of publication of this document; Products or features contained herein may be covered by one or more U.S. or foreign patents.

TABLE OF CONTENTS

Compatibility 5
  Tested Versions 5
  Platforms 5
  Mapping of Infinity Points to Proxy Extensions 5
Installation 7
Quick Start 8
Infinity Component Guides 10
  InfinityNetwork 10
    Network Action Menus 11
  Communicator 12
    Transmitter 12
    Receiver 12
    Transaction Manager 15
    Poll Scheduler 15
    Unsolicited Mgr 15
    Serial Port Parameters 15
  InfinityNetworkDevice 16
  InfinetDevice 17
  InfinityProxyExt 18
Andover Infinity Views 20
  Device Manager 20
  Point Manager 22
  Infinity Virtual Terminal 25
    Terminal Mode Usage Notes 26
    Useful keystrokes and commands in Terminal view 27
    Performing a Backup 28
    Performing a Restore 29
Special Point Value Conversion Considerations 32
  Conversion from Infinity point values to EC-Net^{AX} Boolean points 32
  Conversion from Infinity point values to EC-Net^{AX} Numeric points 32
  Conversion from Infinity point values to EC-Net^{AX} Enum points 33
  Conversion from Infinity point values to EC-Net^{AX} String points 33
Maintenance Support Features 34
Document Change Log 35

COMPATIBILITY

Tested Versions

The Andover Infinity driver was developed against the following two Infinity systems:

1) CMX-240, firmware version 1.4
    a. SCX920, firmware version 1.31
    b. LCX800, firmware version 2.16
2) Continuum NetController CPU-4M
    a. TCX867,
    b. TCX840,
    c. LCX800, firmware version 2.16

The Andover Infinity driver has been field tested against the following Infinity systems:

1) CMX-246, firmware version 2.16
2) CX-9200 (firmware version unknown)

Platforms

The ANDOVER INFINITY driver will function on all EC-Net^{AX} platforms that support serial RS-232 communications. EC-Net^{AX}-3.2 or higher is required.

Mapping of Infinity Points to Proxy Extensions

The Andover Infinity driver allows the creation of any of the following ControlPoint/ InfinityProxyExt combinations:

- NumericPoint with InfinityProxyExt
- NumericWritable with InfinityProxyExt
- BooleanPoint with InfinityProxyExt
- BooleanWritable with InfinityProxyExt
- StringPoint with InfinityProxyExt
- StringWritable with InfinityProxyExt
- EnumPoint with InfinityProxyExt
- EnumWritable with InfinityProxyExt

Compatibility

The table below further shows how the point manager creates these combinations.

To determine how point values obtained from the Infinity panels are converted to data for each of the above point types, see “Special Point Value Conversion Considerations.”

```
For Infinity, the following is used for mapping points to types:
1) System Variables - created based on point name
   Point Name                     Maps to control point types (first
                                  in list is default)
   ----------------               ----------------------------------
   Date                           BStringPoint
   Poweruptime                    BStringPoint
   Timeofday                      BStringPoint
   Version                        BStringPoint
   Month                          BStringPoint,BEnumPoint,BBooleanPoint
   Weekday                        BStringPoint,BEnumPoint,BBooleanPoint
   Alarms                         BNumericPoint,BStringPoint
   Dayofmonth                     BNumericPoint,BStringPoint
   Dayofyear                      BNumericPoint,BStringPoint
   Errors                         BNumericPoint,BStringPoint
   Freemem"                       BNumericPoint,BStringPoint
   Hour                           BNumericPoint,BStringPoint
   Hourofday                      BNumericPoint,BStringPoint
   Minute                         BNumericPoint,BStringPoint
   Scan                           BNumericPoint,BStringPoint
   Second                         BNumericPoint,BStringPoint
   Year                           BNumericPoint,BStringPoint
   Powerfail                      BBooleanPoint

2) Assigned Points - created based on point value
   Point Value                    Maps to control point types (first
                                  in list is default)
   ----------------               ----------------------------------
   On,Off,1,0                     BBooleanPoint,BBooleanWritable,
```

INSTALLATION

To use the EC-Net^{AX} Andover Infinity driver, you must have a target host that is licensed for the feature “andoverInfinity”. In addition, other device limits or proxy point limits may exist in your license.

From your PC, use the EC-Net^{AX} Pro 3*.2.15 or later* installed with the “installation tool” option (checkbox “This instance of EC-Net^{AX} Pro will be used as an installation tool”). This option installs the needed distribution files (.*dist* files) for commissioning various models of remote EC-BOS^{AX} platforms. The dist files are located under your EC-Net-AX install folder in various revision-named subfolders under the “sw” folder.

When installing EC-Net^{AX} Pro on your PC, you should also select the **andoverInfinity** module.

Apart from installing the 3*.n.nn* version of the EC-Net^{AX} distribution files in the EC-BOS^{AX}, make sure to install the **andoverInfinity** module too (if not already present, or upgrade if an older revision). For more details, see “About the Commissioning Wizard” in the EC-BOS^{AX} *Install and Startup Guide*.

Following this, the station is now ready for ANDOVER INFINITY software integration, as described in the rest of this document.

QUICK START

1. Create A new station
    - Follow the installation and configuration instructions preceding this.
    - On EC-Net^{AX} Pro menu “Tools”, select “New Station”.
    - In the “Nav” pane of EC-Net^{AX} Pro, expand the new station, and double-click the “drivers” branch. This will display a “Driver Manager” in the right pane.
    - Click “New” in the “Driver Manager”. Select the “Infinity Network”, and click “Add”. Answer the remaining wizard prompts until the new network is added.
    - Right click on “config.bog” for the new station in the Nav tree, and select “Save”.
    - At this point you will need to download the new station to the EC-BOS^{AX} and start the station.
2. Download and Start the new Station on the target EC-BOS^{AX}.
3. Configure and Learn
    - Open the running station in EC-Net^{AX} Pro
    - View the property sheet for the InfinityNetwork and expand the “Communicator” property.
        - Expand the “Receiver” property. Set the following values as initial starting points:
            - Response Timeout: 5 sec.
            - Elapsed Time For End Of Message: 1.000 sec
            - Elapsed Time After Prompt For End Of Message: 0.075 sec
            - Elapsed Time While In Backup Mode For End Of Message: 10 Sec
            - Elapsed Time While in Building List For End Of Message: 15 Sec
        - Expand “Serial Port Parameters” and enter in the communication settings that match the setup of the Infinity panel you are connecting to.
    - View the property sheet for the “Network Device” under the Infinity Network. This is a special “frozen” slot that represents the Infinity panel that the RS-232 cable connects to. It includes properties for “User Name” and “Password”.
        - Set the “User Name” property – the name “acc” is the factory configured property. You should select a user that has “Administrate” privileges on the Infinity system. IT IS RECOMMENDED THAT YOU NOT USE THE DEFAULT “ACC” USER NAME, but instead create and use a new user created specifically for the Infinity driver to use.
        - Set the “Password” property
    - Save the station and restart the EC-BOS^{AX}.
    - Right-click the InfinityNetwork in the Nav tree and select “Infinity Virtual Terminal”. You should see the Infinity panel’s VT100 interface and normal driver communications to/from the Infinity panel in this manager view as it happens in real-time. You may wish to perform an interactive “console session” to the panel by clicking the “Start Console Session” button. If you are not able to perform any tasks using the console session, then you most likely have a communications parameters settings mismatch between the panel and the EC-BOS^{AX} or there is a wiring problem. BE SURE TO CLICK “STOP CONSOLE SESSION” when you are done before proceeding.

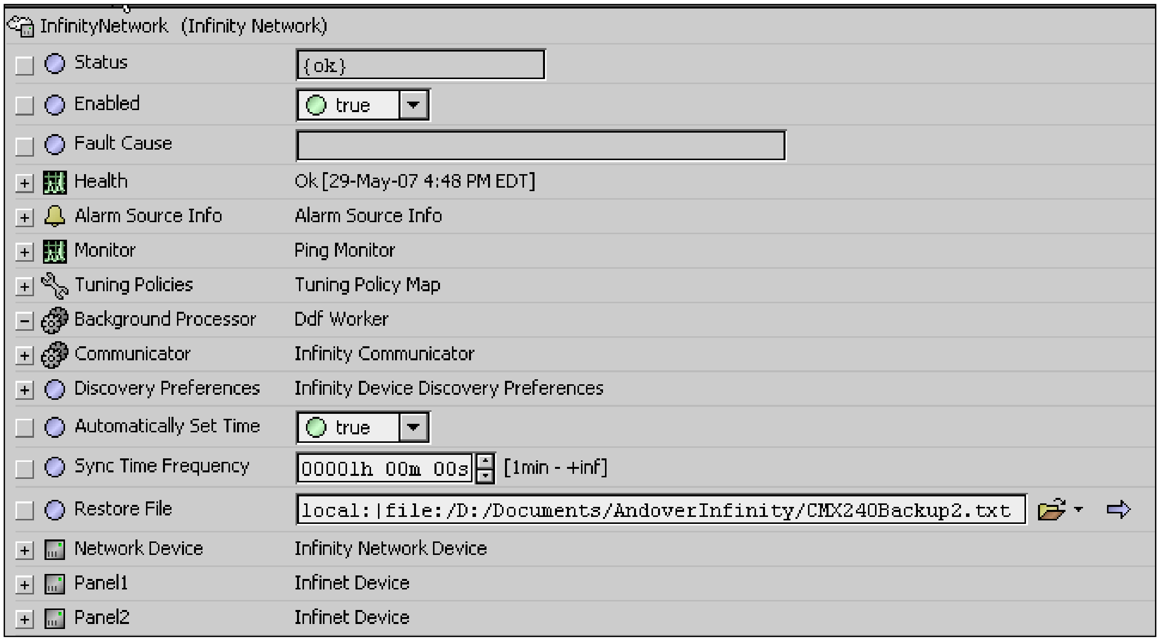
- Double click the InfinityNetwork in the Nav tree to display the Device Manager
    - Click Discover and wait for the discovery to complete.
    - Select one or more discovered devices in the top pane and click “Add”.
    - You now have one or more InfinityDevice(s) added.
- Navigate to the “Points” folder under the device and double click the points folder to display the “Andover Infinity Point Manager” view.
    - Click the “Discover” button and wait for discovery to complete
    - Select one or more discovered points in the top pane and click “Add”
    - You now have one or more Infinity points added, and they should be polling.

INFINITY COMPONENT GUIDES

InfinityNetwork

The InfinityNetwork provides all the configuration parameters necessary to allow the driver to communicate with a network of Andover Infinity devices.

The InfinityNetwork is the "network-level" component in the EC-Net^{AX} architecture. It has the standard network component properties such as status and enabled (see "Driver Architecture / Common network components" in the *EC-Net^{AX} Drivers Guide* for more information), as well as properties unique to configuring an Infinity network.



InfinityNetwork (Infinity Network)

Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [29-May-07 4:48 PM EDT]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Background Processor	Ddf Worker
Communicator	Infinity Communicator
Discovery Preferences	Infinity Device Discovery Preferences
Automatically Set Time	true
Sync Time Frequency	00001h 00m 00s [1min - +inf]
Restore File	local:|file:/D:/Documents/AndoverInfinity/CMX240Backup2.txt
Network Device	Infinity Network Device
Panel1	Infinet Device
Panel2	Infinet Device

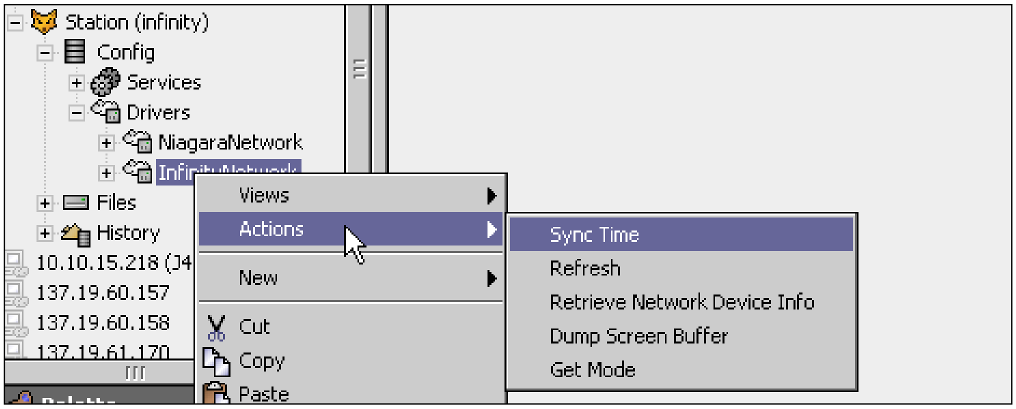
NOTE: In the following properties, the grayed out properties are inherited from the base EC-Net^{AX} driver classes, and as such are only touched on here. For a full explanation, refer to the “Common network components” section in the EC-Net^{AX} Drivers Guide.

- **Status** – The status of the network. Will normally be {ok}. See Fault Cause property for more information if not {ok}.
- **Enabled** – Enables or Disables the Andover Infinity Driver
- **Fault Cause** – if the “Status” is fault, the fault cause will appear here.
- **Health** – Communications health and fail statistics
- **Alarm Source Info** – configuration for routing alarms
- **Monitor** – container for monitor (ping) properties.
- **Tuning Policies** - A container for tuning policies which determines how and when proxy points are read and written.
- **Background Processor** – Frozen slot for a worker thread used in the driver. No configurable properties.
- **Communicator** – A container for all properties that pertain to the communications stack. *SEE “Communicator” section below.*

- **Discovery Preferences** – Communication properties specific to Infinity device discovery operations, which take much longer than normal ping and poll communications.
- **Automatically Set Time** – If set to true, then the date and time in the Infinity panel will be updated with the EC-BOS^{AX} date and time at a frequency determined by the next property, and will be updated at station start or whenever the Infinity network transitions from down to up.
- **Sync Time Frequency** – how often to send set time commands to the Infinity network panel, if "Automatically Set Time" is set true.
- **Restore File** – the file name used for the last backup or restore operation
- **Network Device** – a frozen slot that is identical to a dynamically added "Infinet Device", with the additional properties of "User Name" and "Password". *See the "InfinityNetworkDevice" section.*
- Zero or more "Infinet Device"s may appear as 'dynamic' slots. *See the "InfinetDevice" section.*

Network Action Menus

There are several actions available by right-clicking on the InfinityNetwork object in the nav tree (or right-click the InfinityNetwork on a property sheet).

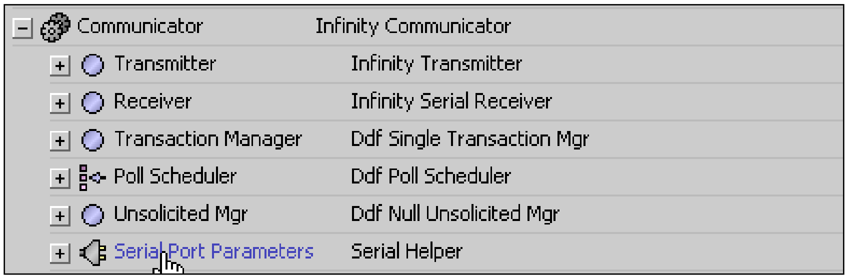


- **Sync Time** – causes the time in the Infinity network controller to be set to the time currently in the Infinity driver platform. This is done by switching the VT100 interface to menu mode, navigating the "Edit/System Date and Time" menus, and entering in the current time. Note that this action is also called periodically by the driver if the property "Automatically Set Time" is true.
- **Refresh** – causes a CTRL-Z (refresh) command to be sent to the Infinity network controller, which in turn causes the controller (most of the time) to re-send all of its VT100 lines. Note that CTRL-Z does not resend controller lines if the current state of the VT100 interface has the cursor inside of a "OK" or "Yes"/"No" or "Ok"/"Cancel" dialog box.
- **Retrieve Network Device Info** – This queries the Network Device for "device id" properties (See the property "Device Id" under the Network Device). Since the network device is a frozen slot on the network, it is not "learned" the way other (Infinet) devices are learned, and therefore the informational properties of Model, Serial Number, and Id are not current the first time the driver is started. To obtain these values, use this command. Note that these properties are persistent, so this command should only need to be issued whenever the driver is first configured or whenever the Infinity device configuration has changed.

- **Dump Screen Buffer** – this is a diagnostic action that dumps the contents of the driver's VT100 buffers to stdout (viewable using the platform tools). The screen buffer consists of actually two separate buffers, one that holds screen content, and one that holds screen formatting information. Both of these are printed to stdout.
- **Get Mode** – this is a diagnostic action that prints the current Screen Buffer mode to stdout. The driver makes extensive use of "modes" to track the state of the interface and to send the appropriate commands.

Communicator

When you expand the "Communicator" property under the InfinityNetwork, you get a list of properties that are used to view/monitor the communication activity of the network. The communicator consists of several modular components that are divided in functionality:

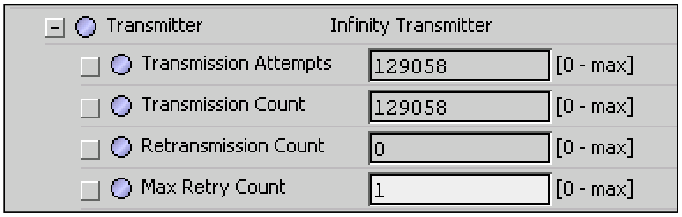


Communicator	Infinity Communicator
Transmitter	Infinity Transmitter
Receiver	Infinity Serial Receiver
Transaction Manager	Ddf Single Transaction Mgr
Poll Scheduler	Ddf Poll Scheduler
Unsolicited Mgr	Ddf Null Unsolicited Mgr
Serial Port Parameters	Serial Helper

The majority of Communicator properties are under the **Receiver** slot.

Transmitter

The transmitter component for the Infinity driver is an "InfinityTransmitter". This component holds properties that govern or track the transmitting of messages from the driver to the field panel



Transmitter	Infinity Transmitter	
Transmission Attempts	129058	[0 - max]
Transmission Count	129058	[0 - max]
Retransmission Count	0	[0 - max]
Max Retry Count	1	[0 - max]

- **Transmission Attempts** – the number of messages that the driver has attempted to send
- **Transmission Count** – the number of messages actually sent
- **Retransmission Count** – the number of messages which had to be re-sent
- **Max Retry Count** – This is the only configurable property on the transmitter component. This is the number of times the driver will attempt to retransmit a message if the initial attempt fails. For Infinity, a count of '1' is recommended.

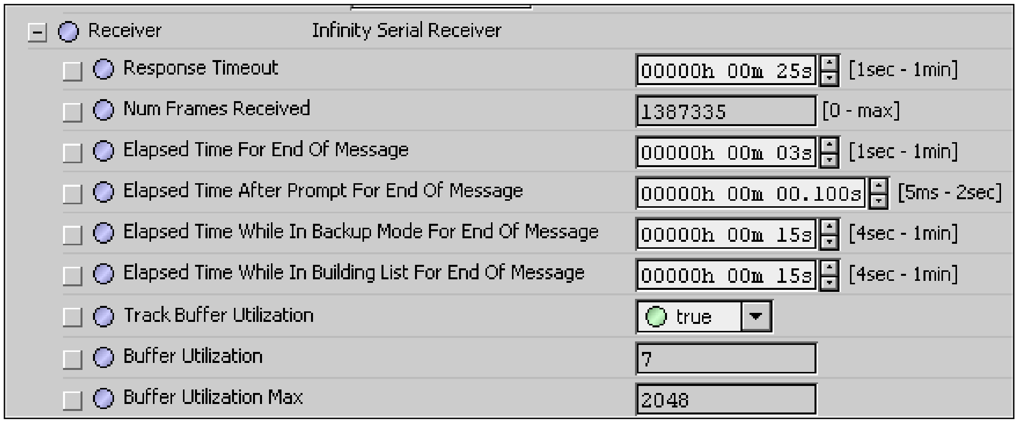
Receiver

The Receiver component is used to configure properties necessary for proper reception of messages from the Infinity network device. The Infinity driver differs from most drivers in that because it uses an interface meant for human interaction with a VT100 terminal, the only way to determine when a message is complete wait until characters stop being received. Depending on what "command" or "keystrokes" were sent to the panel, the timeout to wait for characters to stop arriving may be substantial.

For example, to do a point learn, keystrokes are sent to navigate the “view / points” menu, and the controllers spend a matter of seconds forming a point list for display. However, “normal” polling from the driver uses the command line mode “print” command, which has a much faster turn-around time. For this reason, there are always two timeout properties in effect for any given transaction: 1) a “Response Timeout”, and 2) an “Elapsed Time” timeout.

The “Response Timeout” property is the total duration that a response may take to send a response. For special operations such as point or device learns, this duration is specified in the popup that displays when you launch the learn. For all other operations, this timeout is specified here. See Table 1 on page 14 for recommended settings.

The “Elapsed Time” properties specify the elapsed time *since the last (or previous) character arrived* after which the message can be declared as complete. This is an optimization that is used on common/frequent transactions so that the driver does not have to wait for the entire “Response Timeout” time between every message. Note that the message still has to fit within the confines of the “Responses Timeout” time.



Receiver	Infinity Serial Receiver	
Response Timeout	00000h 00m 25s	[1sec - 1min]
Num Frames Received	1387335	[0 - max]
Elapsed Time For End Of Message	00000h 00m 03s	[1sec - 1min]
Elapsed Time After Prompt For End Of Message	00000h 00m 00.100s	[5ms - 2sec]
Elapsed Time While In Backup Mode For End Of Message	00000h 00m 15s	[4sec - 1min]
Elapsed Time While In Building List For End Of Message	00000h 00m 15s	[4sec - 1min]
Track Buffer Utilization	true	
Buffer Utilization	7	
Buffer Utilization Max	2048	

- **Response Timeout** – the total amount of time that the driver will wait after a transmission for the response to be complete. In the Infinity driver this is the amount of time the driver will wait if no characters are received before issuing a retry (or declaring a failure if retries are exhausted).
- **Num Frames Received** – the count of messages declared to have been received.
- **Elapsed Time For End Of Message** – This time should be set longer than the longest time between characters that occurs whenever the Infinity panel is sending characters to the driver. This is the fall-back inter-character timeout that signals that a message has been received in full, and is used if conditions for use of the other “Elapsed Time” properties are not in effect. If this value is too short, the symptoms are that you will see a “Raw Response” in the point manager display that corresponds to a different point. This value should be set as low as possible while keeping communication faults from occurring.
- **Elapsed Time After Prompt For End Of Message** – this is an optimization that allows for a very quick timeout whenever the cursor is positioned after an “R>” prompt. This optimization takes advantage of the fact that whenever the cursor is moved to directly after the “R>” prompt in command line mode, then the message is near complete and no large inter-character times are expected. Typically, this can be set in the range of 20 to 50 ms.
- **Elapsed Time While In Backup Mode For End Of Message** – Backup mode is entered by clicking the “Restore” button on the “Infinity Virtual Terminal” view. This sends the “SAVE INFINET” command to the controller, which in turn will cause the controller to start sending the controller backup to the driver. The controller backup lines and sections typically can have very large gaps of time when the controller is queuing up

additional lines to be sent. This property needs to be set longer than the longest time gap expected in the response. If incomplete backup files are being received, then most likely this property needs to be set longer. This is not a performance critical value, since backups are only performed by user invocation.

- **Elapsed Time While In Building List For End Of Message** – Device learns and point learns use the "View/Infinet Controllers" and "View/Points" and "View/System Variables" menu mode commands to retrieve lists of panels or lists of points. During the execution of these commands, the controller typically displays the message "Building list – please wait" on the status line of the VT100 terminal. If this message is detected, then the inter-character timeout specified here will be used to allow the controller time to generate the point list. If point learns or device learns are failing, this parameter might need to be set longer. You can test for the required timeout by using the "Infinity Virtual Terminal" to navigate to the View menus described above, and timing how long the "Building List – please wait" message appears on the status line.
- **Track Buffer Utilization –** If set to true, the following two properties will be calculated. Used for diagnostic purposes.
- **Buffer Utilization –** The current, or instantaneous, snapshot of bytes left in the serial port input buffer that are waiting to be processed.
- **Buffer Utilization Max –** The peak number of bytes that have been used in the serial port input buffer.

Table 1 Recommended Settings.

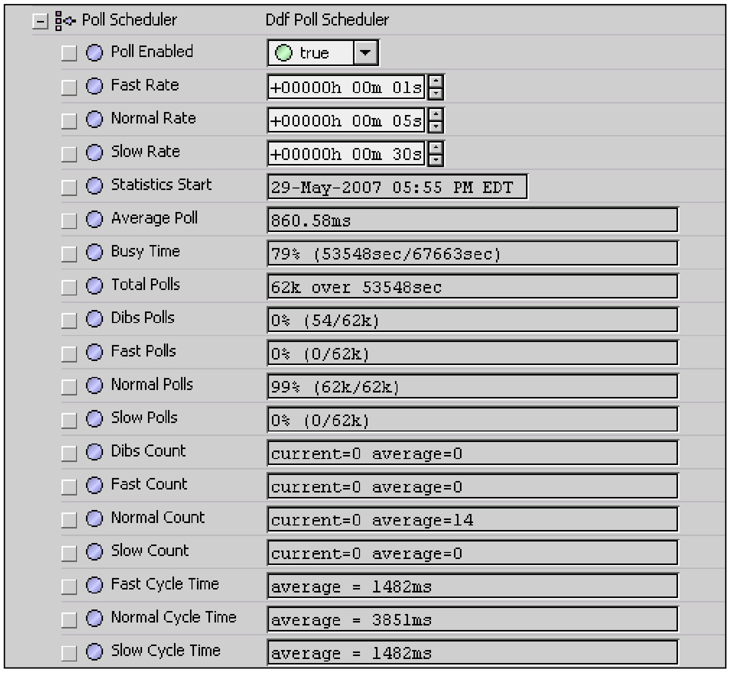
Communicator Receiver Property	**Recommended setting**
Response Timeout	25 sec
Elapsed Time For End Of Message	3 sec
Elapsed Time After Prompt For End Of Message	0.100 sec
Elapsed Time While In Backup Mode For End Of Message	15 sec
Elapsed Time While In Building List For End Of Message	15 sec

Transaction Manager

The transaction manager manages matching the response with the request. There are no configuration items for the Infinity driver.

Poll Scheduler

Sets the fast/normal/slow rates for polling.



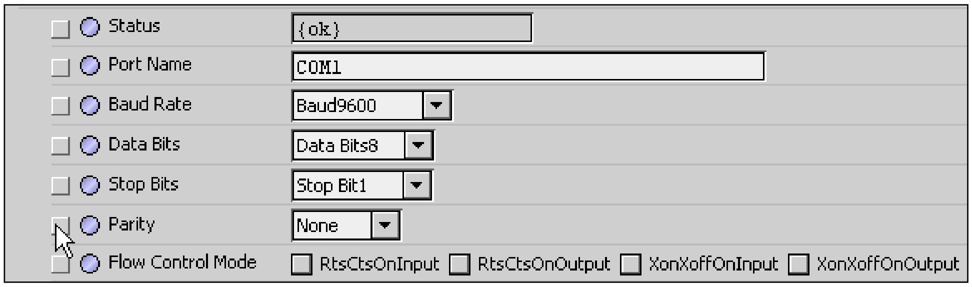
Poll Scheduler	Ddf Poll Scheduler
Poll Enabled	true
Fast Rate	+00000h 00m 01s
Normal Rate	+00000h 00m 05s
Slow Rate	+00000h 00m 30s
Statistics Start	29-May-2007 05:55 PM EDT
Average Poll	860.58ms
Busy Time	79% (53548sec/67663sec)
Total Polls	62k over 53548sec
Dibs Polls	0% (54/62k)
Fast Polls	0% (0/62k)
Normal Polls	99% (62k/62k)
Slow Polls	0% (0/62k)
Dibs Count	current=0 average=0
Fast Count	current=0 average=0
Normal Count	current=0 average=14
Slow Count	current=0 average=0
Fast Cycle Time	average = 1482ms
Normal Cycle Time	average = 3851ms
Slow Cycle Time	average = 1482ms

Unsolicited Mgr

Not implemented for Infinity driver.

Serial Port Parameters

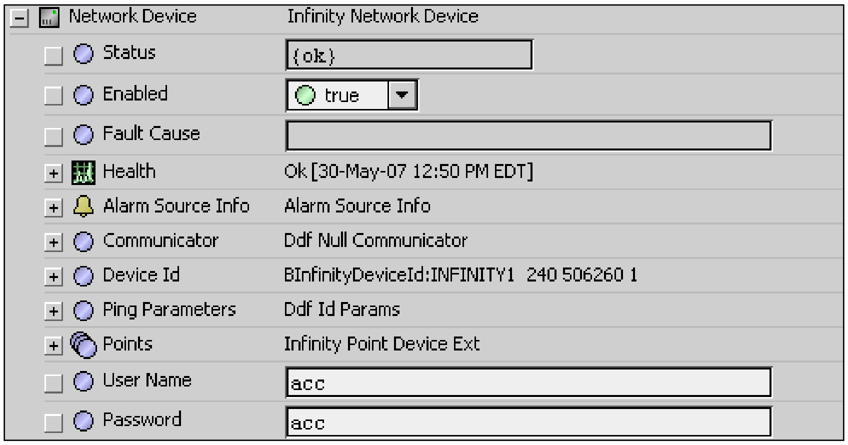
Serial port parameters should match the parameters configured in the Infinity com port that the driver is connected to.



Status	{ok}
Port Name	COM1
Baud Rate	Baud9600
Data Bits	Data Bits8
Stop Bits	Stop Bit1
Parity	None
Flow Control Mode	☐ RtsCtsOnInput ☐ RtsCtsOnOutput ☐ XonXoffOnInput ☐ XonXoffOnOutput

InfinityNetworkDevice

The InfinityNetworkDevice is a permanent or “frozen” device that is always included as a permanent slot on the Infinity Network. The InfinityNetworkDevice corresponds to the Infinity panel to which the RS-232 connection is made.

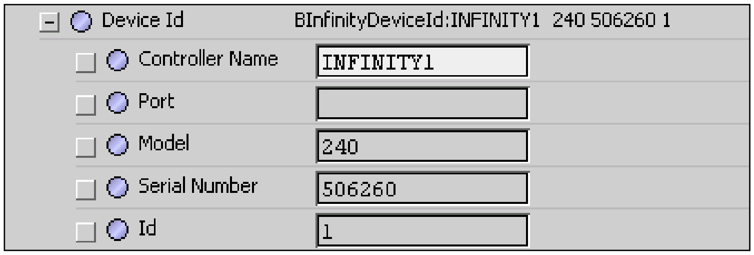


Network Device	Infinity Network Device
☐ Status	{ok}
☐ Enabled	true
☐ Fault Cause	
Health	Ok [30-May-07 12:50 PM EDT]
Alarm Source Info	Alarm Source Info
Communicator	Ddf Null Communicator
Device Id	BInfinityDeviceId:INFINITY1 240 506260 1
Ping Parameters	Ddf Id Params
Points	Infinity Point Device Ext
☐ User Name	acc
☐ Password	acc

NOTE: In the following properties, the grayed out properties are inherited from the base EC-Net^{AX} driver classes, and as such are only touched on here. For a full explanation, refer to the "Common device components" section in the *EC-Net^{AX} Drivers Guide*.

- **Status** – The status of the device. Will normally be {ok}. A value of {down} indicates that the last ping to the device was not answered (the “Print CommStatus” command is used as the ping message).
- **Enabled** – Enables or Disables communication to the associated device from Infinity Driver
- **Fault Cause** – if the “Status” property value is {fault}, the fault cause is displayed here.
- **Heath** – contains metadata about the health of this device on the network:
  - Down – indicates if this device is down – should be false under normal operation.
  - Alarm – indicates if this device is in alarm – should be false under normal operation
  - Last OK Time – the last time of successful communication to this device
  - Last Fail Time – the last time of unsuccessful communication to this device
  - Last Fail Cause – the reason of the last communication failure
- **Alarm Source Info** – configuration items for alarms generated from the ping process (device up and device down events)
- **Communicator** – not used in the Infinity driver

- **Device Id** – contains metadata about the Infinity device



Device Id BInfinityDeviceId:INFINITY1 240 506260 1
Controller Name INFINITY1
Port
Model 240
Serial Number 506260
Id 1

  - **Controller Name** - the controller ID. This should be set to match the controller ID of the main Infinity panel this driver connects to. The ping message and other messages require this property to be set correctly in order to succeed.
  - **Port** – Does not apply to the Network level device
  - **Model**, **Serial Number**, and **Id** – Values reported back from the "Edit/Controller" menu selection of the VT100 interface. These value can be requested and set by an action on the Infinity Network object called "Retrieve Network Device Info"
- **Ping Parameters** – Not used in the Infinity driver
- **Points** – A container object that contains proxy points corresponding to points or system variables on the parent InfinityNetworkDevice. These proxy points are the data items in this device which need to be polled for data.
- **User Name** – an "Administrate" level user name used to log into the Infinity controller by the driver
- **Password** – the password corresponding to the User Name selected above.

InfinetDevice

The InfinetDevice is a dynamic device that is added by a user or by the learn mechanism of the device manager display. This represents an Infinet panel that is connected under the Infinity controller panel that the driver connects to. In the Infinity driver, this object is identical to the InfinityNetworkDevice described above with the following exceptions:

a. The object is dynamically added instead of a frozen slot on the InfinityNetwork.

b. The write Device Id property "Controller Name" should match the name of the Infinet panel configured in the Infinity Network controller.

c. The read-only Device Id properties are populated only if the device is added from the device manager learn mechanism.

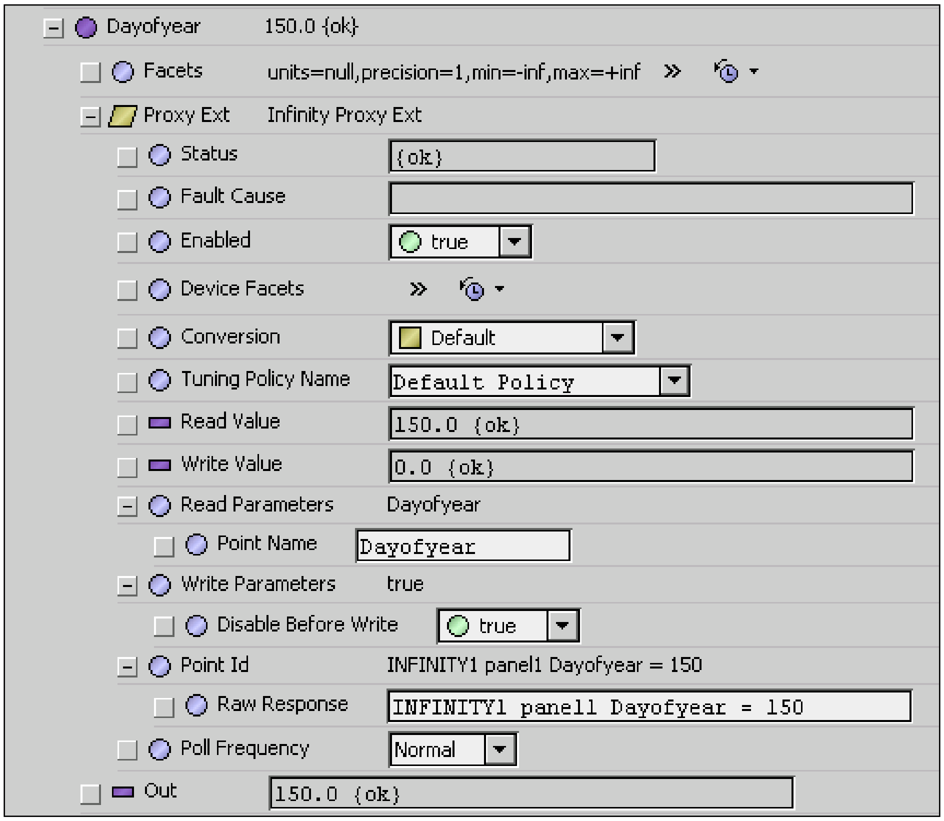
d. The User Name property is not present (you don't "log on" to an Infinet device).

e. The Password property is not present.

InfinityProxyExt

The Infinity proxy extension types take on the readable-writable personality of the control point they are attached to. For example, a InfinityProxyExt, when used as an extension on a NumericPoint has “read only” functionality, but when used on as an extension on a NumericWritable can read and write the point values.

The InfinityProxyExt is the "point-level" component in the EC-Net^{AX} architecture. The following shows an InfinityProxyExt on a NumericPoint control point:



Dayofyear	150.0 {ok}
Facets	units=null,precision=1,min=-inf,max=+inf
Proxy Ext	Infinity Proxy Ext
Status	{ok}
Fault Cause	
Enabled	true
Device Facets	
Conversion	Default
Tuning Policy Name	Default Policy
Read Value	150.0 {ok}
Write Value	0.0 {ok}
Read Parameters	Dayofyear
Point Name	Dayofyear
Write Parameters	true
Disable Before Write	true
Point Id	INFINITY1 panel1 Dayofyear = 150
Raw Response	INFINITY1 panel1 Dayofyear = 150
Poll Frequency	Normal
Out	150.0 {ok}

NOTE: In the following properties, the grayed out properties are inherited from the base EC-Net^{AX} driver classes, and as such are only touched on here. For a full explanation, refer to the “ProxyExt properties” section in the *EC-Net^{AX} Drivers Guide*.

- **Status** – The status of the proxy ext. Will normally be {ok}. Will transition to ‘fault’ if a response to a poll is detected to be an error. Will transition to ‘down’ if the parent device is down. Will indicate ‘stale’ if the point value has not been updated within the stale timeout period (see Tuning Policy property on the network).
- **Fault Cause** – if the “Status” property value is {fault}, the fault cause is displayed here.
- **Enabled** – Enables or Disables communication to the associated point from Infinity Driver
- **Device Facets** – Used to modify the value read to the value displayed. Typically not used in an Infinity integration.
- **Conversion** – Can apply custom conversion to the point values. Applies to both read and (inversely) to write values.

- **Tuning Policy Name** – selects the tuning policy to apply to this point. See "Tuning Policies" property of the InfinityNetwork
- **Read Value** – the value last read from the point in the panel
- **Write Value** – the value last written from this proxy ext (does not apply to read only point types)
- **Read Parameters** – contains metadata on how to address this point in the Infinity panel. For Infinity, the only parameter necessary to specify a unique point is the point name.
- **Write Parameters** – If the control point type is a "Writable" point, then this property governs whether or not an Infinity "Disable" command should be written to the point before the value is written. This allows the point value to be controlled by the NiagaraAX framework rather than any program or link which may be configured in the controller that also attempts to control that same point.
- **Poll Frequency** – Fast, Normal, or Slow. Rates are determined by the Poll Scheduler settings on the InfinityNetwork/Communicator/PollScheduler property.

ANDOVER INFINITY VIEWS

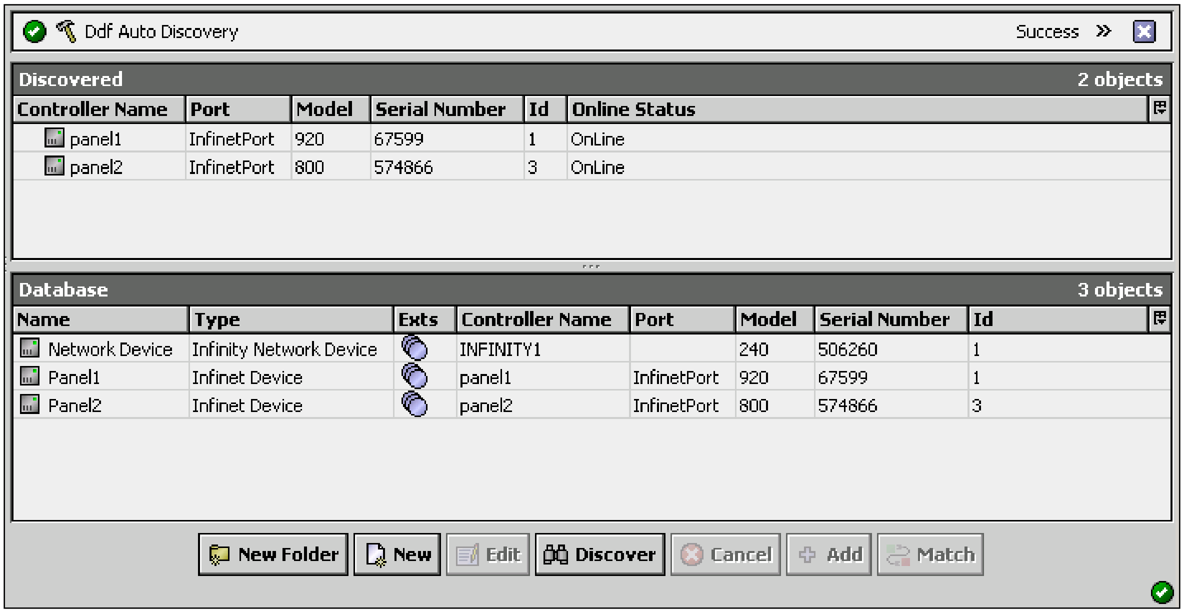
Andover Infinity views include the following:

- Device Manager
- Point Manager
- Infinity Virtual Terminal

Device Manager

The Device Manager is the default view when you double-click on an InfinityNetwork in the Nav tree. This manager view provides a quick and easy way to display and learn Infinity devices that are on the Infinet network.

The Andover Infinity Device Manager consists of either one or two main panes, depending on whether or not the "Discover" button has been clicked. The view below shows a typical Andover Infinity Device Manager view.



Ddf Auto Discovery Success

Discovered 2 objects

Controller Name	Port	Model	Serial Number	Id	Online Status
panel1	InfinetPort	920	67599	1	OnLine
panel2	InfinetPort	800	574866	3	OnLine

Database 3 objects

Name	Type	Exts	Controller Name	Port	Model	Serial Number	Id
Network Device	Infinity Network Device		INFINITY1		240	506260	1
Panel1	Infinet Device		panel1	InfinetPort	920	67599	1
Panel2	Infinet Device		panel2	InfinetPort	800	574866	3

The "**New Folder**", "**New**", and "**Edit**" buttons are not unique to the Andover Infinity Device Manager, and are explained in the *EC-Net*^{AX} *Drivers Guide* in the "About the Device Manager" section.

The "**Discover**" button is used to launch a device discovery process.

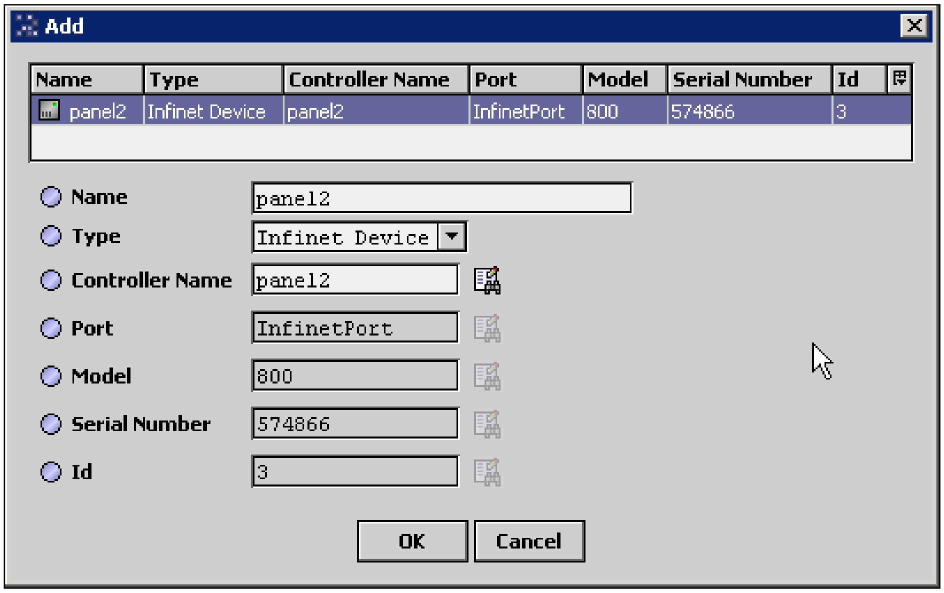
The "**Match**" button is used to match a device discovered in the top pane with a device that already exists in the database in the bottom pane. This is useful update panel metadata in the database device with data obtained from the learn. For Infinet devices, this is the only way to update values for "Port", "Model", "Serial Number", and "Id".

The "**Cancel**" button will end a discovery that is in process.

The "**Add**" button allows manual addition of a device object into the database (note that the Port", "Model", "Serial Number", and "Id" will not be set if the device is added in this manner, but can be set later using the "Match" button).

The "**Discover**" button implements functionality that is unique and tailored to discovering ANDOVER INFINET devices. By clicking the "Discover" button, the "learn" mode of the manager is invoked (the panes will be split, and a "discovery" table will be displayed in the top pane) and a discovery process will be launched. Any discovered devices will be displayed in the top pane.

If you highlight one or more rows in the top "Discovered" pane, the "Add" button becomes active. You can now add the selected devices to the station database by clicking the "Add" button. This will pop up the "**Add**" dialog box:



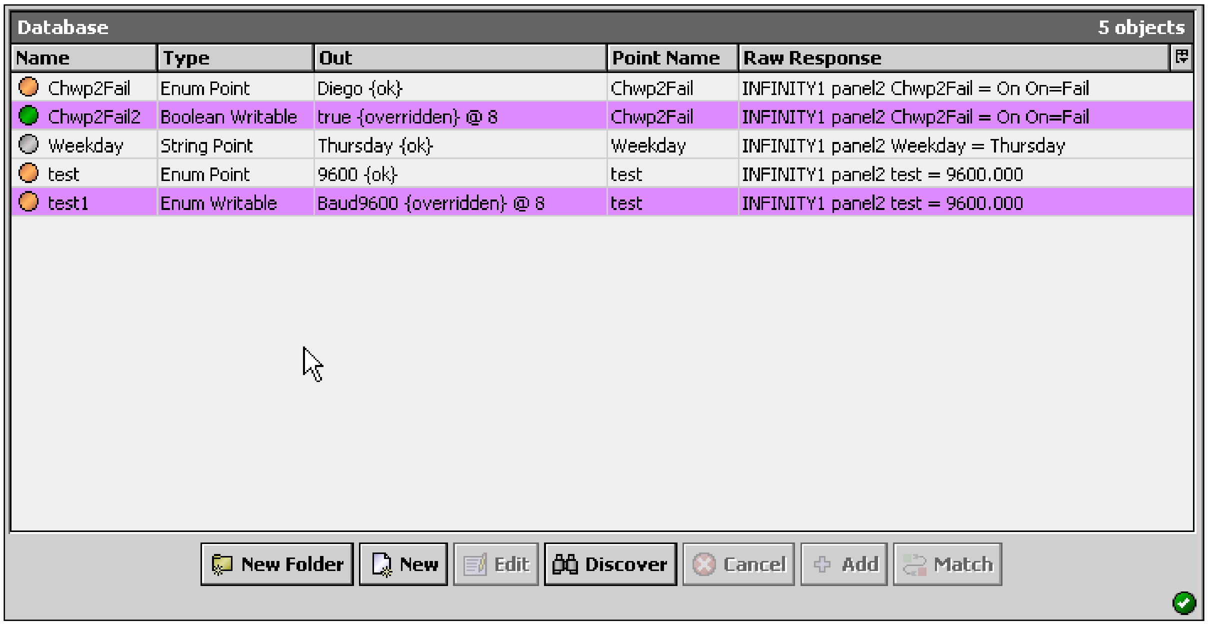
Name	Type	Controller Name	Port	Model	Serial Number	Id
panel2	Infinet Device	panel2	InfinetPort	800	574866	3

○ Name: panel2
○ Type: Infinet Device
○ Controller Name: panel2
○ Port: InfinetPort
○ Model: 800
○ Serial Number: 574866
○ Id: 3

The "Add" dialog box affords you the opportunity to change the display name and/or controller name of each of the selected devices. Click the "OK" button to add the devices to the database, or click "Cancel" to bail out.

Point Manager

The Point Manager is the default view when you double-click on a “points” folder under an InfinityDevice in the Nav tree. The Point Manager is a table-based view, where each row represents a unique point in an Infinity controller. Below is an example Point Manager view:

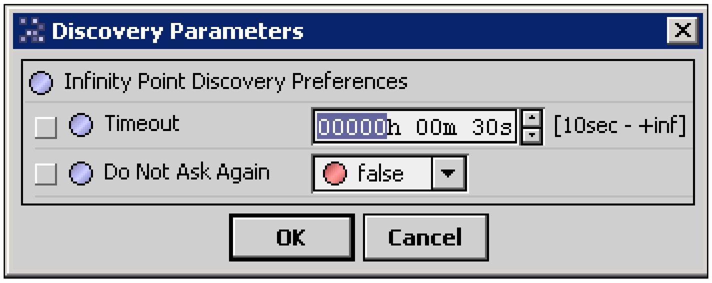


Database 5 objects

Name	Type	Out	Point Name	Raw Response
Chwp2Fail	Enum Point	Diego {ok}	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Chwp2Fail2	Boolean Writable	true {overridden} @ 8	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Weekday	String Point	Thursday {ok}	Weekday	INFINITY1 panel2 Weekday = Thursday
test	Enum Point	9600 {ok}	test	INFINITY1 panel2 test = 9600.000
test1	Enum Writable	Baud9600 {overridden} @ 8	test	INFINITY1 panel2 test = 9600.000

The “**New Folder**”, “**New**”, and “**Edit**” buttons are not unique to the Infinity point Manager, and are explained in the “About the Point Manager” chapter in the *EC-Net^{AX} Drivers Guide*.

By clicking the “**Discover**” button, the “learn” mode of the manager is invoked (the panes will be split, and a “discovery” table will be displayed in the top pane) and a dialog box will be popped up to prompt the user for some additional data to guide the discovery process, as shown here:



Discovery Parameters

Infinity Point Discovery Preferences

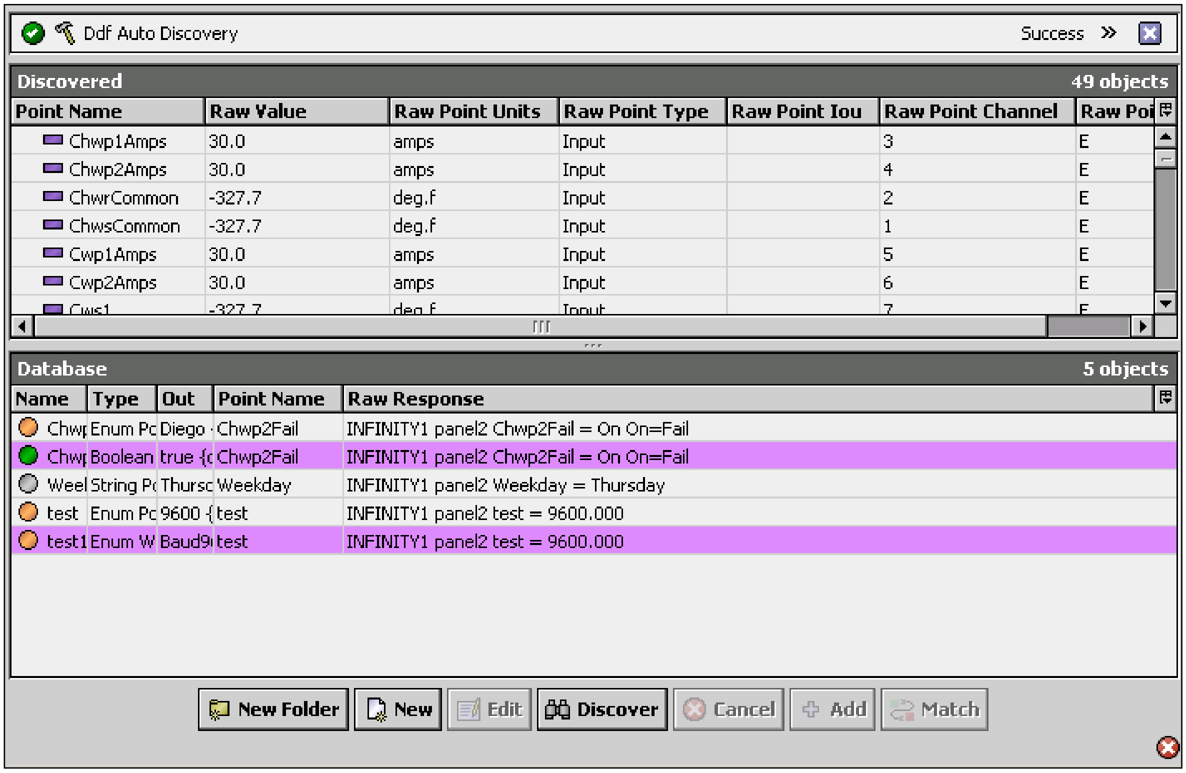
☐ Timeout 00000h 00m 30s [10sec - +inf]

☐ Do Not Ask Again false

This dialog box contains two parameters, the first is a “**Timeout**” value, and should be set for the entire duration that a point discover is expected to take. The parameter “**Do Not Ask Again**” will cause this dialog to not be displayed on subsequent learns.

(Note, the discovery parameters are actually a property on the “Points” folder, you can always reset the “Do Not Ask Again” property if you set it to “true” in the dialog).

Clicking the OK button (or, if the “Do Not Ask Again” property has previously been set to “true”, then simply clicking the Discover button) will cause a learn process to be kicked off, the progress of which will be displayed in the job bar at the top of the split screen. When the learn process is done, the results will be displayed in the “Discovered” pane on the top half of the point manager display:



Ddf Auto Discovery — Success

Discovered — 49 objects

Point Name	Raw Value	Raw Point Units	Raw Point Type	Raw Point Iou	Raw Point Channel	Raw Poi
Chwp1Amps	30.0	amps	Input		3	E
Chwp2Amps	30.0	amps	Input		4	E
ChwrCommon	-327.7	deg.f	Input		2	E
ChwsCommon	-327.7	deg.f	Input		1	E
Cwp1Amps	30.0	amps	Input		5	E
Cwp2Amps	30.0	amps	Input		6	E
Cws1	-327.7	deg.f	Input		7	E

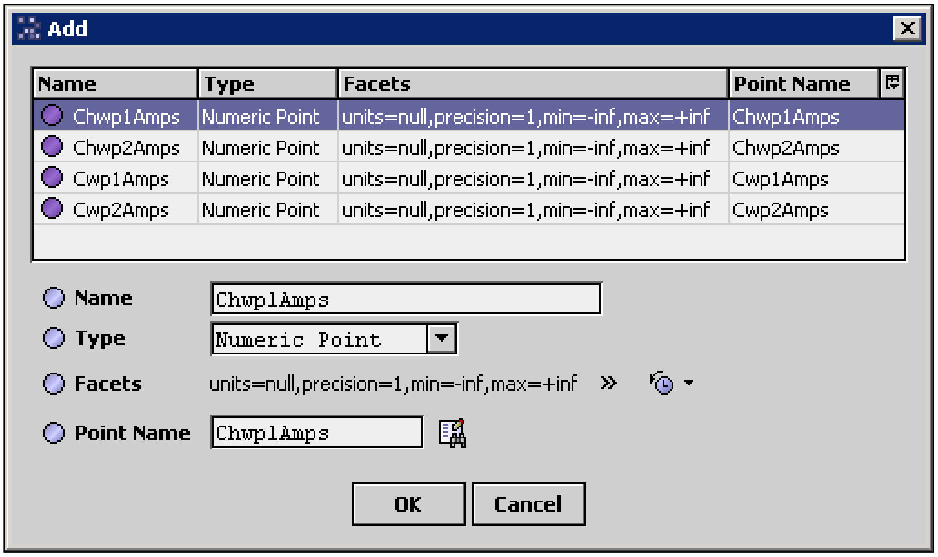
Database — 5 objects

Name	Type	Out	Point Name	Raw Response
Chwp	Enum Pc	Diego ·	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Chwp	Boolean	true {c	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Weel	String Pc	Thursc	Weekday	INFINITY1 panel2 Weekday = Thursday
test	Enum Pc	9600 {	test	INFINITY1 panel2 test = 9600.000
test1	Enum W	Baud9	test	INFINITY1 panel2 test = 9600.000

The display line for a learned point shows the point name and several columns that are the information returned from the "View/Points" and/or the "View/System Variables" menus of the Infinity VT100 interface. The "raw" columns are displayed to assist the integrator in selecting the proper points to be added to the database – the "raw" columns are not retained as properties of the point after it is added to the database.

Highlighting one or more rows in the top pane enables the "Add" and "Match" buttons. To add one or more points to the database, select them (multi-select by using CTRL-click or SHIFT-click operations with the mouse), and then select Add.

By clicking "Add" the point(s) are displayed in an "Add" dialog box where the added name, point name, and point type may be modified:



Add

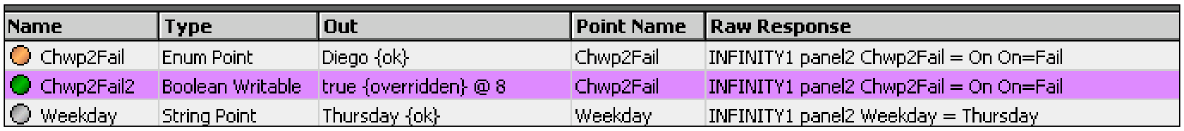
Name	Type	Facets	Point Name
Chwp1Amps	Numeric Point	units=null,precision=1,min=-inf,max=+inf	Chwp1Amps
Chwp2Amps	Numeric Point	units=null,precision=1,min=-inf,max=+inf	Chwp2Amps
Cwp1Amps	Numeric Point	units=null,precision=1,min=-inf,max=+inf	Cwp1Amps
Cwp2Amps	Numeric Point	units=null,precision=1,min=-inf,max=+inf	Cwp2Amps

Name: Chwp1Amps
Type: Numeric Point
Facets: units=null,precision=1,min=-inf,max=+inf
Point Name: Chwp1Amps

In this dialog, the points may be edited individually or in a batch. Note that any edits apply to ALL selected (highlighted) points in the Add dialog box. The “Type” determines which base point type will be created, and the “dataType” governs how the data retrieved from the Infinity “Print” command for each point are to be interpreted. ***There may be reasons for changing these at this point, but in general the most appropriate type of base point is already selected as the first type in the “Type” pull down selection box. You might want to verify that the Type does not need to be changed to a “writable” version of the same type.*** For a list of defaults and available selections for each of the point types, see the section “*Mapping of Infinity Points to Proxy Extensions*”.

Once the point(s) are satisfactorily edited, click “**OK**” to create the proxy points corresponding to the group addresses.

Once added to the database, the points appear in the lower pane (or only pane if not in learn mode) of the Point Manager display. See the following illustration:

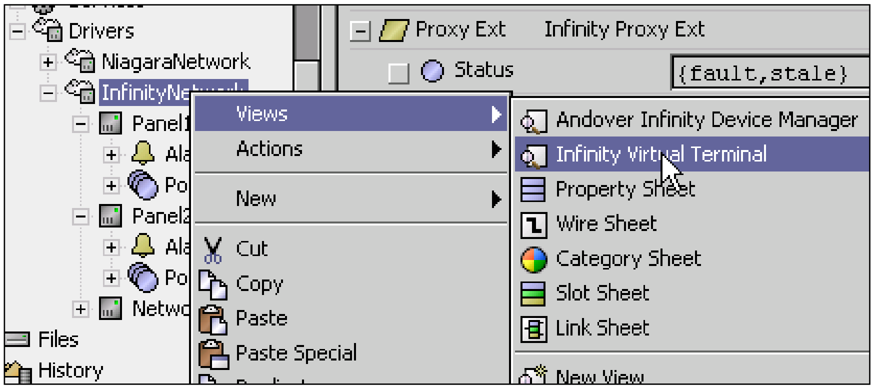


Name	Type	Out	Point Name	Raw Response
Chwp2Fail	Enum Point	Diego {ok}	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Chwp2Fail2	Boolean Writable	true {overridden} @ 8	Chwp2Fail	INFINITY1 panel2 Chwp2Fail = On On=Fail
Weekday	String Point	Thursday {ok}	Weekday	INFINITY1 panel2 Weekday = Thursday

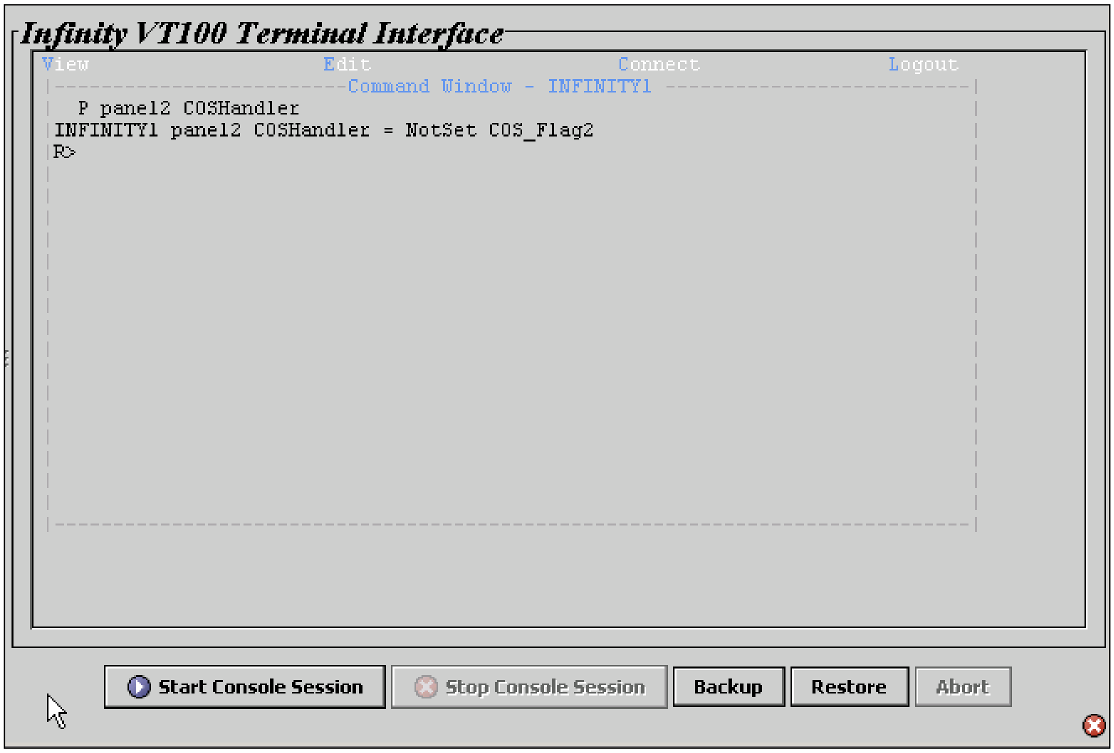
- **Name**: This is the name of the proxy ext in the EC-Net^{AX} database. It is, by default, derived from the “Point Name” from the point if the point was created using the driver’s learn capability, it may be changed to any valid name either during or after the learn. In the above illustration, you can see that Point Name “Chwp2Fail” has been added as two points – both a “Enum Point” and an “Boolean Writable”. This is perfectly valid, and in fact, in the Infinity driver, will still only result in a single poll to retrieve the values for both proxy points. Note that the system automatically assigned “Chwp2Fail2” to the second point to differentiate it from the first added point.
- **Type**: This is the point type, which defines how the point is handled internally within the NiagaraAX framework. The “Raw Response” is attempted to be parsed to a value which matches the “Type”. Note that in the above example, that the “Point Name” Chwp2Fail has a raw value of “On” that has been converted to a “true” value in the BooleanWritable proxy point “Chwp2Fail2”, but has been converted to an Enum value “Diego” in the enum point “Chwp2Fail”.
- **Out**: This is the “Raw Response” converted to an output value.
- **Point Name**: This is the point name that is polled using the Infinity “Print” command.
- **Raw Response**: This is the text string returned in the VT100 command line area in response to the Infinity “Print” command.

Infinity Virtual Terminal

The Infinity Virtual Terminal is a special view under the InfinityNetwork component. Access this view by right-clicking “Infinity Virtual Terminal” in the component Nav tree:


Drivers
NiagaraNetwork
InfinityNe
Panel1
Ala
Po
Panel2
Ala
Po
Netwo
Files
History
Proxy Ext
Infinity Proxy Ext
Status
{fault,stale}
Views
Actions
New
Cut
Copy
Paste
Paste Special
Andover Infinity Device Manager
Infinity Virtual Terminal
Property Sheet
Wire Sheet
Category Sheet
Slot Sheet
Link Sheet
New View

The virtual terminal view allows the user to monitor the driver activity as it is happening in real time. In addition, the user can interrupt the normal driver communications to perform interactive VT100 commands with the Infinity system, or to perform backups and restores of the system:


Infinity VT100 Terminal Interface
View
Edit
Connect
Logout
Command Window - INFINITY1
Start Console Session
Stop Console Session
Backup
Restore
Abort

```
 P panel2 COSHandler
INFINITY1 panel2 COSHandler = NotSet COS_Flag2
R>
```

The Infinity VT100 Terminal Interface view shows the VT100 interface of the Infinity device to which driver is connected. It shows the real-time communications that are taking place from the controller to the panel.

In addition, a user can enter an interactive session with the controller by clicking the "**Start Console Session**" button. The session will remain active until the user clicks "**Stop Console Session**" or exits the view. A console session will pre-empt all other communications for the duration of the session, causing the devices to be marked down and point to go stale. Exiting the console session should allow the driver to resume normal driver communications.

While in the console session, all keystrokes are from the user are sent to the Infinity controller just as if a terminal were connected directly to the controller. Therefore, the full range of Infinity configuration, programming, and file operations are available while in this "interactive mode."

CAUTION: Care should be used to leave the controller in command line mode when exiting the console session. While every attempt has been made to be able to detect the state of the VT100 interface and to be able to transition automatically from that state back to command line mode (the mode required for 99% of normal driver comm.), there are sure to be cases where the driver would not be able to recover.

See the following subsections for more details:

- Terminal Mode Usage Notes
- Useful keystrokes and commands in Terminal view
- Performing a Backup
- Performing a Restore

Terminal Mode Usage Notes

The next section "Useful keystrokes and commands in Terminal view" lists the most commonly used keystrokes and commands used for navigating in the Infinity menu system from within the Andover Infinity driver's virtual terminal view.

Interactive mode is a useful tool to use when first establishing communications to the panel during early project development, to make changes to the Infinity system, or to debug communications or driver issues. An interactive session sends keystrokes to the Infinity panel just as if the integrator were locally attached with a VT100 terminal.

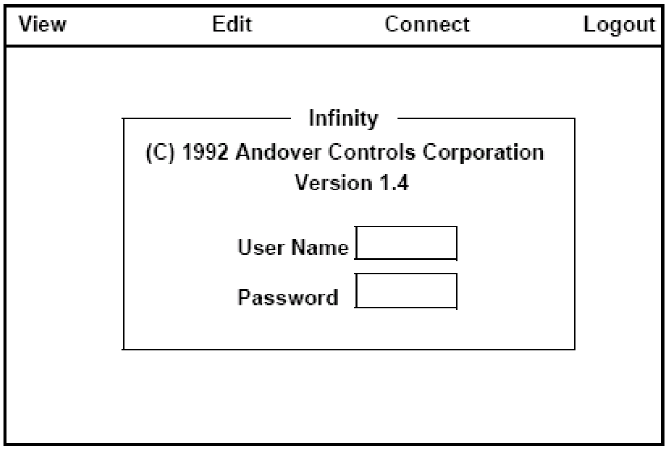
With that in mind, there are a couple of "scenarios" of which user should be aware of when entering terminal mode:

1) Scenario number 1: The Infinity controller was last left in a logged-on state. A simple **Ctrl-Z** should be sent to refresh the driver's VT100 buffer.
2) Scenario number 2: The Infinity controller was last left in a logged-off state. The "WINDOW" command should be entered to log on to the controller.
3) Scenario number 3: The Infinity controller was last left logged-on, but with an open dialog box which requires closing before the **Ctrl-Z** command will work. In this case, enter multiple **Esc** keys and multiple **Enter** keys to close any open dialog boxes, and then type **Ctrl-Z** to refresh the driver'sVT100 buffer.
4) If none of the above are successful in establishing a terminal session, then the next avenue to pursue would be to check the following:
   a. Make sure baud rate is configured in the driver matches the panel baud rate.
   b. Make sure the physical RS-232 cable is pinned out correctly (in general, on Rx/Tx/Gnd are required between the Infinity panel and EC-BOS^{AX}, but hardware handshaking lines may need to be "strapped" to the correct state at the Infinity panel (check RTS, CTS, DTR, DSR lines at the Infinity connector.
   c. Most Infinity/Continuum panels have multiple COM ports. Make sure you are connected to one configured for terminal access. Likewise, most EC-BOS^{AX}'s have

multiple COM ports. Make sure you are plugged into the one the driver is configured to use.

Useful keystrokes and commands in Terminal view

- **Ctrl-Z** – “REFRESH”. Refresh the terminal view – this requests an entire VT100 buffer be resent from the controller. This should always be the first command typed by a user when entering the interactive terminal mode session from the driver’s Terminal view. If nothing happens, the controller may not be in a logged-on state, in that case use the “WINDOW” command described below.
- **WINDOW** - Type the word “WINDOW” (no quotes). The word does not appear on the screen. Instead, a box appears that takes up most of the screen. The box is called the login “window”:



Infinity
(C) 1992 Andover Controls Corporation
Version 1.4

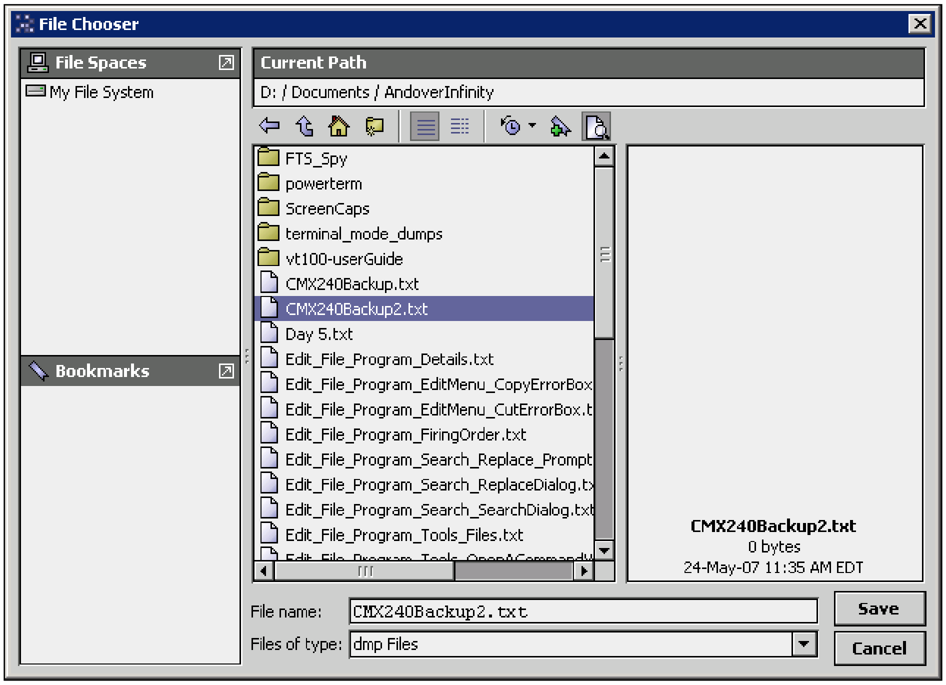
User Name
Password

- **Enter** key – Use to commit text entered into a text blank. Also used to close open “Error”, “Warning”, or “Confirm” dialog boxes.
- **Esc** key – Use to cancel an open dialog window and return to either the main window or the previous (parent) window. This command will close an open dialog box that is waiting for input. Sometimes upon driver startup, neither “Refresh” (Ctrl-Z) nor “WINDOW” will get a response. This can happen if the Infinity panel was last left in a state where an “Error”, “Warning”, or “Confirm” dialog box was last displayed, and the panel is waiting for an “ENTER” or “CANCEL” command to close the dialog.
- **Tab** key - Use to move to next text entry box or next attribute in the Infinity menus. Can also be used to navigate the menu bar.
- **F2** key – “Show List”. Some Infinity entry boxes have a defined set of correct entries, which may be obtained for selection using the F2 key. This is similar functionality to a pull-down “selection box” in a Windows application. For example, use this key when selecting COM ports in the menus that configure devices and printers in the Infinity menu systems, or when configuring point types.
- **F4** key – “Cancel Window”. Closes a window and erases changes, then returns to the menu bar. Use to toggle between menus at the top of the VT100 windows and the working area below the menus.
- **Pg Up**/**Pg Dn** keys – used to paint the working area of the screen with any information which may be prior or next. Useful when displaying point lists.
- **Esc-I** – (hold down Esc key and hit “i” key) - This key combination toggles insert mode, whereby typing overwrites or inserts. In general, the driver tries to keep the Infinity panel in overwrite mode, because insert mode consumes substantially more bandwidth than non-insert mode.

- **Esc-6** – (hold down Esc key and hit “6” key) – This key combination erases the current line and moves the cursor up one line. In general, the driver in normal operation uses this command to erase lines to minimize bandwidth usage. If a user is using the terminal in interactive mode, it is a good idea to erase all the lines before ending the terminal session and going back into driver mode. You will notice in non-interactive mode that the driver erases lines upon completion of a command.
- **Arrow keys** (up/down/left/right) – use to move the cursor one position in the respective direction. Can be useful for navigating point or device lists, or reviewing the interactive terminal command space. Also useful for navigating the menu bar.
- **Backspace** key – Erases a single character that appears before the cursor.

Performing a Backup

A “backup” of the Infinity system can be performed by clicking on the “**Backup**” button in the Infinity Virtual Terminal view. This causes a file chooser dialog box to appear:

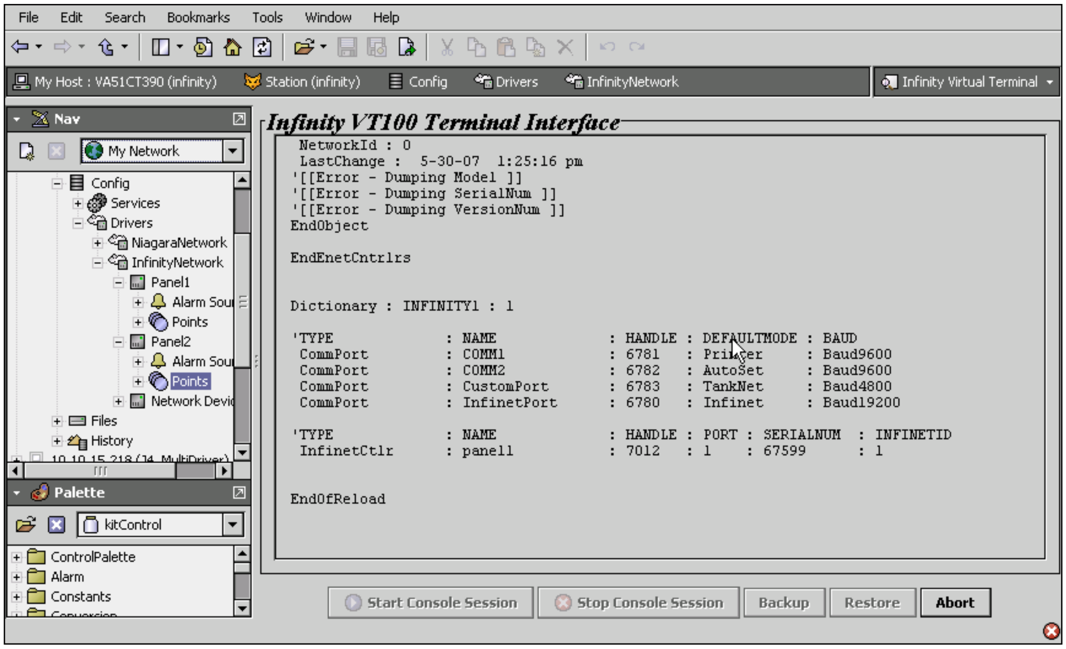

File Chooser
File Spaces
My File System
Bookmarks
Current Path
FTS_Spy
powerterm
ScreenCaps
terminal_mode_dumps
vt100-userGuide
CMX240Backup.txt
CMX240Backup2.txt
Day 5.txt
Edit_File_Program_Details.txt
Edit_File_Program_EditMenu_CopyErrorBox
Edit_File_Program_EditMenu_CutErrorBox.t
Edit_File_Program_FiringOrder.txt
Edit_File_Program_Search_Replace_Prompt
Edit_File_Program_Search_ReplaceDialog.tx
Edit_File_Program_Search_SearchDialog.txt
Edit_File_Program_Tools_Files.txt
CMX240Backup2.txt
0 bytes
24-May-07 11:35 AM EDT
Save
Cancel

Current Path: D: / Documents / AndoverInfinity

File name: CMX240Backup2.txt

Files of type: dmp Files

The user should navigate to a location on the EC-Net^{AX} Pro machine and name the backup file. The file should always have the extension “.txt”.   When the “Save” button is then pressed, the driver will send a “Save Infinet” command to the controller, and the text returned from the controller will be saved to the file specified here. In addition, the text will be written to the VT100 screen so the user can monitor the progress of the save operation. The “Abort” button will be the only button enabled at this point.

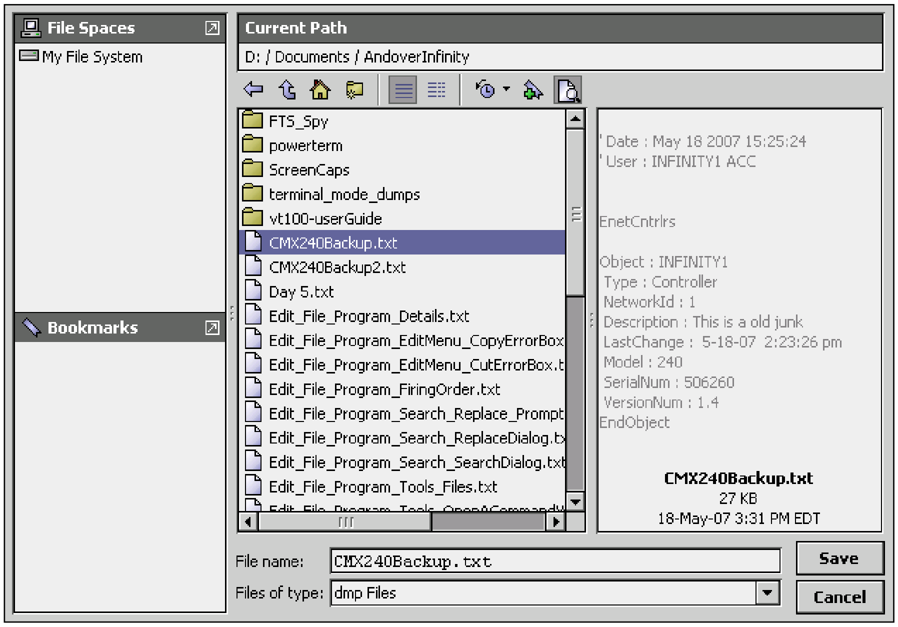

File Edit Search Bookmarks Tools Window Help
My Host : VA51CT390 (infinity)
Station (infinity)
Config
Drivers
InfinityNetwork
Infinity Virtual Terminal
Nav
My Network
Config
Services
Drivers
NiagaraNetwork
InfinityNetwork
Panel1
Alarm Sou
Points
Panel2
Alarm Sou
Points
Network Devi
Files
History
Palette
kitControl
ControlPalette
Alarm
Constants
Infinity VT100 Terminal Interface
Start Console Session
Stop Console Session
Backup
Restore
Abort

```
  NetworkId : 0
  LastChange :  5-30-07  1:25:16 pm
 '[[Error - Dumping Model ]]
 '[[Error - Dumping SerialNum ]]
 '[[Error - Dumping VersionNum ]]
 EndObject

 EndEnetCntrlrs


 Dictionary : INFINITY1 : 1

 'TYPE              : NAME                : HANDLE : DEFAULTMODE : BAUD
  CommPort          : COMM1               : 6781   : Printer     : Baud9600
  CommPort          : COMM2               : 6782   : AutoSet     : Baud9600
  CommPort          : CustomPort          : 6783   : TankNet     : Baud4800
  CommPort          : InfinetPort         : 6780   : Infinet     : Baud19200

 'TYPE              : NAME                : HANDLE : PORT : SERIALNUM  : INFINETID
  InfinetCtlr       : panel1              : 7012   : 1    : 67599      : 1


 EndOfReload
```

When the backup operation finishes, the button should return to the default state.

Performing a Restore

The "**Restore**" button in the Infinity Virtual Terminal view allows restoring a previous backup file. Just like in the "Backup" case, the "Restore" case prompts the user for a file:


File Spaces
My File System
Bookmarks
Current Path
FTS_Spy
powerterm
ScreenCaps
terminal_mode_dumps
vt100-userGuide
CMX240Backup.txt
CMX240Backup2.txt
Day 5.txt
Edit_File_Program_Details.txt
Edit_File_Program_EditMenu_CopyErrorBox
Edit_File_Program_EditMenu_CutErrorBox.t
Edit_File_Program_FiringOrder.txt
Edit_File_Program_Search_Replace_Prompt
Edit_File_Program_Search_ReplaceDialog.t
Edit_File_Program_Search_SearchDialog.txt
Edit_File_Program_Tools_Files.txt
Save
Cancel

D: / Documents / AndoverInfinity

```
' Date : May 18 2007 15:25:24
' User : INFINITY1 ACC


EnetCntrlrs

Object : INFINITY1
 Type : Controller
 NetworkId : 1
 Description : This is a old junk
 LastChange :  5-18-07  2:23:26 pm
 Model : 240
 SerialNum : 506260
 VersionNum : 1.4
EndObject
```

CMX240Backup.txt
27 KB
18-May-07 3:31 PM EDT

File name: CMX240Backup.txt
Files of type: dmp Files

Select the file and choose "Save". The driver will then send the "RELOAD –o" command and begin sending the contents of the selected file to the controller.

NOTE: Since the controller does not send any visible evidence that it is reloading, other than if an error occurs, there may be no feedback during a reload that the reload is progressing. Please be patient!!!

NOTE: Since all sections of the previously saved Infinet file may *not* be suitable/desirable to reload, a mechanism has been defined to specify which sections of the file, and possibly which entries within sections, are to be omitted from the reload. These are defined as follows:

1) Comment lines are not transmitted.
2) By default, dictionary sections are not transmitted.
3) By default, Import/Export sections are not transmitted.
4) Sections and entries are not transmitted if specifically excluded by entries in the modules lexicon file. The relevant entries in the lexicon file, as shipped, are shown below, and can be modified as needed:

```
# *** dump file lines to ignore on reload ***
#
# 1)  if an entry's value is "Exclude" it will
#     be ignored, even if included in the source
#     file, before being sent to the controller.
# 2)  If the value here is anything other than "Exclude",
#     the line will be sent to the controller.
# 3)  If the value is not listed here, then the
#     line will be sent to the controller.
# 4)  All comment lines (beginning with a "'") are
#     automatically "Exclude"
# 7)  Entire sections can be excluded by
#     Section.SectionName=Exclude
# 8)  All sections must have a corresponding end entry

Section.Dictionary=Exclude
Section.DictionaryEnd=EndDictionary
Section.ImportExport=Exclude
Section.ImportExportEnd=EndImportExport

#values found between "EnetCntrlrs" and "EndEnetCntrlrs"
Section.EnetCntrlrs=Include
Section.EnetCntrlrsEnd=EndEnetCntrlrs
EnetCntrlrs.Object=Include
EnetCntrlrs.Type=Include
EnetCntrlrs.NetworkId=Include
EnetCntrlrs.LastChange=Exclude
EnetCntrlrs.Model=Exclude
EnetCntrlrs.SerialNum=Exclude
EnetCntrlrs.VersionNum=Exclude
EnetCntrlrs.EthernetId=Exclude
EnetCntrlrs.IPAddress=Include
EnetCntrlrs.SubnetMask=Include
EnetCntrlrs.DefaultRouter=Include
EnetCntrlrs.MaxResponseTime=Exclude
EnetCntrlrs.Controller$20Options=Exclude
EnetCntrlrs.Xdriver$20Comm1=Exclude
EnetCntrlrs.Xdriver$20Comm2=Exclude
EnetCntrlrs.Xdriver$20Comm3=Exclude
EnetCntrlrs.Xdriver$20Comm4=Exclude
EnetCntrlrs.Max$20Infinet$20Controllers=Exclude
EnetCntrlrs.LAN=Exclude
EnetCntrlrs.PCB$20Revision=Exclude
EnetCntrlrs.Web$20Server=Exclude
EnetCntrlrs.Max$20IOUModules=Exclude
EnetCntrlrs.ACC_LON$20I$2fO$20Selected=Exclude
EnetCntrlrs.PPPIPAddr=Exclude
EnetCntrlrs.EndObject=Include

#values found between "BeginController" and "EndController"
Section.BeginController=Include
Section.BeginControllerEnd=EndController
BeginController.Object=Include
BeginController.Type=Include
BeginController.LastChange=Exclude
BeginController.Windows=Include
BeginController.DefaultMode=Include
BeginController.TrackCXD=Include
BeginController.EndObject=Include
BeginController.Code=Include
```

SPECIAL POINT VALUE CONVERSION CONSIDERATIONS

Conversion from Infinity point values to EC-Net^{AX} Boolean points

The Infinity driver uses the following sequence of conversions to convert Infinity point data into Boolean data for use within a BooleanPoint or BooleanWritable. Using this conversion technique it is possible to convert Infinity Boolean, Numeric, or String type points to Boolean values.

1) Try to convert the point string to a number. If this succeeds, then if the value is non-zero the EC-Net^{AX} value is true, else the value is false.

2) Else, see if the point string is "On" or "Off". If so, return true or false, respectively.

3) Else,

    3a) If there are any Boolean type facets on the EC-Net^{AX} proxy point, use the facets to convert the value. If the string point equals one of the facet value pair tags, then return the facet value pair value

    3b) If there are no Boolean facets to use, then attempt to lookup the value in the Infinity driver's lexicon file. The entries in the lexicon file should be of the form "point.boolean.define.<string>=<0 or 1>. The <string> value is used as the lookup key in the lexicon file. For example, if the lexicon file has the following entries:

    ```
    point.boolean.define.hokey=1
    point.boolean.define.pokey=0
    ```

    and the value of the response to a poll is "hokey", then the value would be converted to a "true" in the Boolean point.

4) Else, the value cannot be converted, and will display with a "fault" status.

Values can be added as needed to the lexicon file to convert string values to Boolean values using the lexicon editor. In addition, facets can be created on the Proxy point as needed. The preferred method would be to add facets to the point, but the lexicon file affords the opportunity to map several string values to Boolean values.

Conversion from Infinity point values to EC-Net^{AX} Numeric points

The Infinity driver uses the following sequence of conversions to convert Infinity point data into numeric data for use within a NumericPoint or NumericWritable. Using this conversion technique it is possible to convert Infinity Boolean, Numeric, or limited String type points to Numeric values.

1) Try to convert the point string to a number. If this succeeds, then if the value is non-zero the EC-Net^{AX} value is true, else the value is false.

2) Else, see if the point string is "On" or "Off". If so, return 1 or 0, respectively.

3) Else, the value cannot be converted, and will display with a "fault" status.=

Conversion from Infinity point values to EC-Net^{AX} Enum points

The Infinity driver uses the following sequence of conversions to convert Infinity point data into enum data for use within a EnumPoint or EnumWritable. Using this conversion technique it is possible to convert Infinity Boolean, Numeric, or limited String type points to Enum values.

1) Try to convert the point string to a number. If this succeeds, then use the resulting number (truncated to nearest whole number) as the ordinal into the enumeration

2) Else, try to use the point string directly as the “tag” of the enumeration. If it matches one of the pre-defined tags, then the enumeration value is the value that corresponds to that tag.

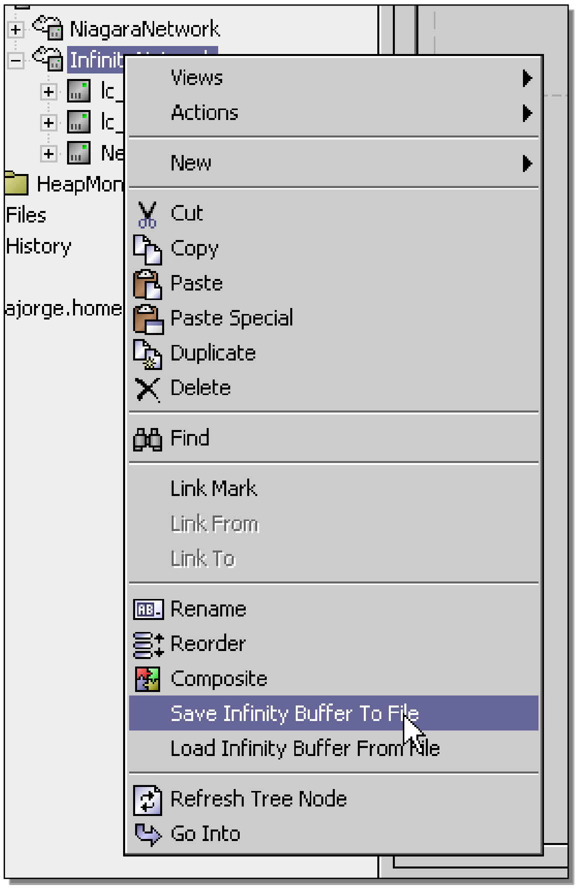
3) Else, the value cannot be converted, and will display with a “fault” status.

Conversion from Infinity point values to EC-Net^{AX} String points

No conversion is performed. The resultant value of the EC-Net^{AX} point is the string to the right of the “=” sign of the “Raw Response”.

MAINTENANCE SUPPORT FEATURES

The driver includes two commands available by right-clicking the InfinityNetwork object:


NiagaraNetwork
Infinit
Ic_
Ic_
Ne
HeapMon
Files
History
ajorge.home
Views
Actions
New
Cut
Copy
Paste
Paste Special
Duplicate
Delete
Find
Link Mark
Link From
Link To
Rename
Reorder
Composite
Save Infinity Buffer To File
Load Infinity Buffer From File
Refresh Tree Node
Go Into

The first command, “**Save Infinity Buffer To File**”, will capture a snapshot of the state of the driver’s screen buffer, and save the contents to a file. This information may be requested for support purposes if there are problems encountered in the field. The resulting text file is saved on a client machine running EC-Net^{AX} Pro, and can be copied and sent to technical support. The command can also be used for purely documentation purposes.

Step 1: Display the terminal view and click “Start Console Session”.

Step 2: Wait for the screen to stop updating.

Step 3: Right click the Infinity Network in the navigation tree, and select “Save Infinity Buffer To File”. Select a folder and file name and click save in the File Chooser dialog box.

Step 4: Click “Stop Console Session” in the terminal view to resume normal driver activity.

The second command, “**Load Infinity Buffer From File**” is able to take a file generated by the “Save Infinity Buffer From File” and reload it into the screen buffer. Typically, this would be performed by technical support personnel to recreate a field problem or event.

Step 1: Display the terminal view. Click “Start Console Session” in the terminal view.

Step 2: Wait for the screen to stop updating.

Step 2: Right-click the Infinity Network in the navigation tree, and select “Load Infinity Buffer From File”. In the file chooser dialog box, navigate to the desired file, and select “Open”.

Step 3: Verify in the Terminal Interface that the file loads and displays.

Step 4: Click “Stop Console Session” in the terminal view to resume driver operation.

DOCUMENT CHANGE LOG

Updates (changes/additions) to this *EC-Net^{AX} Andover Infinity Driver Guide* document are listed below.

- Updated: November 2011
  Additions made in "**Tested Versions**" section under Compatibility. References changed to the *EC-Net^{AX} Drivers Guide* instead of the *EC-Net^{AX} User Guide*.
- Updated: December 17, 2007
  The section on the **Infinity Virtual Terminal** view had two new subsections added about working in an interactive "console session": **Terminal Mode Usage Notes** and **Useful keystrokes and commands in Terminal view**. This document is no longer marked "Draft."
- Updated: September 12, 2007
  In section describing Communicator **Receiver** properties, the screen capture was updated to show buffer utilization properties, along with revised recommended settings for properties given in Table 1 Recommended Settings. (no longer vary by host platform, as previously documented). A new section **Maintenance Support Features** was added.
- Updated: August 21, 2007

  In section describing Communicator **Receiver** properties, three additional properties related to buffer utilization were added, along with Table 1 Recommended Settings.
- Initial Document: June 1, 2007

  Initial publication as PDF-only document.

Distech Controls Inc
4005-B Boul. Matte
Brossard, QC J4Y 2P4
Tel.: 1-800-404-0043
Tel.: (450) 444-9898

05DI-UGANDIN-10