

Setting up KA-Node network access

Jim Oberhofer KN6PE

January 2010

This Application Node has been updated for v2.4 and describes how to set up Outpost to take advantage of this features. Three examples are shown to describe how it all works together.

Introduction

The following are excerpts from the TAPR website on network schemes, originally published in "Packet Radio: What? Why? How? / Articles and Information on General Packet Radio Topics," TAPR, Publication #95-1. 1995, by Greg Jones, WD5IVD (http://www.tapr.org/pr_intro.html#NetSchemes).

"The first networking scheme with packet radio was Digipeaters. Digipeaters would simply look at a packet, and if its call was in the digipeater field, would resend the packet. Digipeaters allow the extension of range of a transmitter by retransmitting any packets addressed to the digipeater.

"Kantronics improved on the digipeater slightly and created KA-Nodes. As with digipeaters, KA-Nodes simply repeat AX.25 frames. However, a KA-Node acknowledges every transmission at each link (node) instead of over the entire route. Therefore, instead of an end-to-end acknowledgment, KA-Nodes allow for more reliable connections with fewer timeouts, because acknowledgments are only carried on one link.

"NET/ROM was one of the first networking schemes to try to address the problems with digipeaters. A user connects to a NET/ROM station as if connecting to any other packet station. From there, he can issue commands to instruct the station to connect to another user locally or connect to another NET/ROM station. This connect, then connect again, means that to a user's TNC, you are connected to a local station only and its transmissions do not have to be digipeated over the entire network and risk losing packets."

The key take-aways from the above is that using KA-Nodes and NET/ROM node results in:

- More reliable connections
- Fewer timeouts
- Less network traffic

However, with this new capability comes more complexity in configuring Outpost.

Not all Nodes are the same

While all nodes fundamentally do the same thing, that is, get you from here to there, they don't do it exactly in the same manner or with the same types of controls. For instance here's a description of SOME (not all) nodes:

- KPC3** Single port KA-Node, most fundamental. You can connect to it and then connect (**C <next_node>**) to somewhere else further down the path to the BBS.
- KPC9612, KAM4** Dual port KA-Node. With 2 ports, these can be configured to operate on 2 frequencies at the same time. The user can connect to the node on one port and either progress to the next node on the same port (**C <next_node>**) or cross-connect to the other port (**X <next_node>**).
- BPQ Switch** Multi-port NET ROM implementation. This node type is implemented in software and, as the name implies, lets the user connect to the next node either from the same port (**C <next_node>**) or switch to a different port (**C <port#> <next_node>**).
BPQ Switches can also be implemented on the same PC as a BBS thereby acting as the front door to the BBS. Once connected to a BPQ switch, all the user needs to enter is a **BBS** command to connect to the BBS.

Along with this kind of flexibility comes complexity. Getting the configuration right is key to a successful node path implementation.

What you need to know about each node

There are 6 things you need to know about every node that is in the path:

<p>1. Node connect name</p>	<p>This is the station name of a node in the chain. It could be a Call Sign, Call Sign with SSID, or Alias. For example, Station Node names can be:</p> <ul style="list-style-type: none"> ▪ K6FB-7: a KA-Node operated by the Las Cumbres Amateur Radio Club ▪ W6TDM-3: a NET/ROM emulation on a G8BPQ node switch ▪ DIA: In the San Francisco Bay Area, this is the alias name of a station up on Mount Diablo
<p>2. The message returned that indicates that a successful connect to this node occurred</p>	<p>This is some or all of the message that the KA or NET/ROM node will return when a successful connect to the node occurs. In Example 1 below, the KPC-3 node returns 2 lines of text indicating a successful connect. All we care about is the last line, and preferably the last couple of words on the line.</p> <pre>cmd:c k6kp-7 cmd:*** CONNECTED to K6KP-7 ###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A ENTER COMMAND: B,C,J,N, or <u>Help</u> ?</pre> <p>For the NET/ROM successful connect message, its fairly easy...</p> <pre>###<u>LINK MADE</u></pre> <p>I've underlined the portions of the 2 strings that Outpost can use. As with other aspects of Outpost, the program will be looking for these strings to know it can proceed with the next action.</p>
<p>3. The Connect Command needed to get to the <u>next node</u> in the path</p>	<p>To get to the next node, you typically will use the C or Connect command. However, for our dual-port TNCs, you may need to cross-connect to a different band or frequency to get to the next node.</p> <ul style="list-style-type: none"> ▪ When you come in on one port and go out on the <u>same</u> port, you use the "C" or Connect command. ▪ When you come in on one port and go out on the <u>other</u> port, you use the "X" or Xconnect command.
<p>4. Indicate if we connect with the Node or BBS name to the <u>next node</u> in the path</p>	<p>In almost every case, we will issue the next connect command with the name of the next node or BBS. However, the BPQ node switch could be installed with a BBS on the same PC, thereby allowing the user to enter the "BBS" command to connect to the BBS.</p> <p>In Outpost, the default setting is CHECKED: connect with the node/BBS name. See Example 3 below for details.</p>
<p>5. The port number needed to get to the <u>next node</u> in the path</p>	<p>Some Nodes (like the BPQ Switch) have the ability to access several radio ports (1, 2, 3, etc). This parameter indicates the port that Outpost should use to get to the <u>next node</u> or BBS. If you go in and out on the same port, the default is "0". See Example 2 below for details.</p>
<p>6. The Unsuccessful Connect message that is returned if the connect to the <u>next node</u> fails</p>	<p>Identify the message returned by the Node that tells Outpost that we failed to connect so that Outpost will disconnect from the path. The best way to discover this is to force an error. See the Example 1 Step 3 for details.</p>

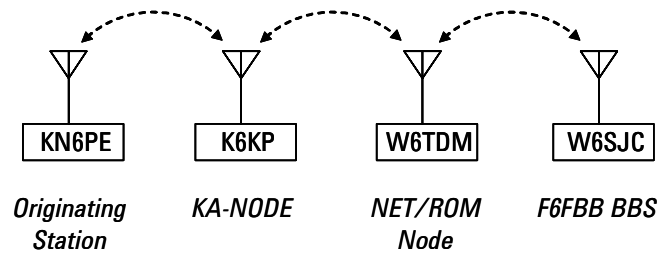
Let's look at how this works by reviewing 3 examples.

Example #1

The first example describes a basic configuration of two types of nodes in Outpost.

Suppose there is a F6FBB BBS named W6SJC out there that I want to access, but cannot get to it directly. However, I can reach it by going through 2 nodes that are in the path between my station and the BBS. These stations are:

- K6KP: a station that has a dedicated Kantronics KPC-3 with its KA-NODE capability turned on.
- W6TDM: a station with the G8BPQ Switch software loaded that acts as a Node in a NET/ROM compatible AX25 network.



To configure any node path, I do the following:

1. Collect information about the path to the BBS
2. Perform the connection manually before starting any Outpost setup
3. Record the relevant information for each node that you will need for the Outpost setup
4. In Outpost, set up the BBS with node access
5. Test the configuration

Step 1. – Collect information about the path to the BBS

In our example, I know the following:

KN6PE	My home station.	
K6KP	K6KP-7	the connect name to get to the KA-Node. While all Kantronics TNCs default to the “-7” SSID for their KA-Nodes, it can be overridden by the TNCs’ sysop.
W6TDM	W6TDM-3	the connect name to get to the NET/ROM Node. In this case, the sysop of the W6TDM node configured the node access name to have an “-3” SSID.
W6SJC	W6SJC-1	the connect name of the BBS

Step 2. – Manually connect to the BBS

This step confirms that you can really get to the BBS with this set of nodes.

I used Outpost’s serial terminal program – ipserial.exe – to execute these steps. However, other programs can be used, such as Hyperterm. Some spaces are introduced into the listing below to improve readability.

Power up the TNC

This is the power-on message I get when turning on the TNC.

On power-on, the TNC’s startup message is displayed followed by the “cmd:” prompt.

```
KANTRONICS KPC3P VERSION 8.2
(C) COPYRIGHT 1997 BY KANTRONICS INC. ALL RIGHTS RESERVED.
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.
cmd:
```

Connect to the first node

Enter the command to connect to the first node in the path to the BBS.

If successful, there are 2 connect messages displayed.

- The first “*** CONNECTED” message is produced by my local TNC.
- The second “###CONNECTED” message is from the K6KP KA-Node

KA-Nodes are pretty friendly, and offer a list of node commands that you could then use.

Connect to the second node

Enter the command to connect to the second node in the path to the BBS.

Similar to the above, the “###LINK MADE” message comes from the W6TDM-3 node on a successful connect.

Unlike the KA-Node, other than the above message, it doesn't really volunteer any additional information. Pressing “enter” only returns another blank line. Entering a “?” does produce the list of commands that the node does support. This is useful for manual interaction only.

Connect to the BBS

Finally, we enter the command to connect to the BBS; it all looks familiar from here.

When done, enter a “b” for bye, and the disconnect process occurs all the way back to my local TNC command prompt.

```
cmd:c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

```
c w6tdm-3
###LINK MADE
<cr>
?
CUPTNO:W6TDM-3} BBS CHAT CONNECT BYE INFO NODES PORTS
ROUTES USERS MHEARD
```

```
c w6sjc-1
CUPTNO:W6TDM-3} Connected to W6SJC-1
[FBB-7.04j-AB1FHMRX$]

Hello Jim,
Welcome to The San Jose EOC Mailbox

W6SJC >
lm
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

W6SJC >
b

73 Jim,
###DISCONNECTED BY W6TDM-3 AT NODE K6KP-7
*** DISCONNECTED
cmd:
```

Step 3. – Record the relevant information about the nodes

As stated, there are 6 things we need to tell Outpost about every node in the path:

1. What is the **node connect name**?
2. What message is returned that indicates that a **successful connect to this node** occurred?
3. What **Connect Command** is needed to get to the next node in the path?
4. Do we **connect with the Node or BBS name** to the next node in the path?
5. What **port number** is needed to get to the next node in the path?
6. What **“Unsuccessful Connect” message** is returned if the connect to the next node fails?

You can gather most of the information from the above example. However, you will need to force an error to get the **“Unsuccessful Connect” message** from the Node. For instance:

KA-NODE

You connect to the first node as usual

```
cmd:c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

You connect to the 2nd node. In this case, enter a station that does not exist.

```
c w6ztn-3
###RETRIED OUT AT NODE K6KP-7
ENTER COMMAND: B,C,J,N, or Help ?
```

The error message indicating a failed connect from the K6KP-7 node is the **“###RETRIED”** message.

NET/ROM Node

Connect to the 1st Node is successful

```
c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

Connect to the 2nd node is successful

```
c w6tdm-3
###LINK MADE
```

Attempt to connect to the BBS, but with a station name that does not exist.

```
c w6sjb-1
CUPTNO:W6TDM-3} Failure with W6SJB-1
```

In this case, an error message is returned.

I only care about any portion of the message that excludes specifics about what we were trying to do. The string **“Failure with”** is ideal as an error message.

Once you have completed the above, I usually summarize it so that I can easily recreate the configuration later. For instance, to get from my home station (KN6PE) to the F6FBB BBS (W6SJC-1), I recorded the following information that I will use to configure Outpost:

	Node Name	Successful connect message	Connect Cmd to next Node/BBS	Connect with the name to the next Node/BBS?	Port Num to the next Node/BBS	Unsuccessful connect message to the next Node/BBS
1	K6KP-7	Help ?	C	Yes [✓]	0	###RETRIED
2	W6TDM-3	###LINK MADE	C	Yes [✓]	0	Failure with

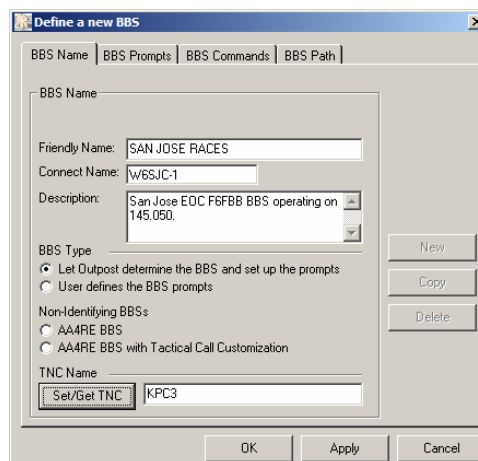
NOTES

1. For the K6KP-7 (KPC-3) Successful Connect Message, I'm only taking the last few characters since they are unique enough to indicate that we are connected.
2. For the W6TDM-3 (BPQ Switch) Unsuccessful Connect Message, I'm taking portions of the message that will be common enough so that this error message should work for other BBSs as well.
3. In all cases, be careful with spaces and capitalization... Outpost is looking for the literal strings.

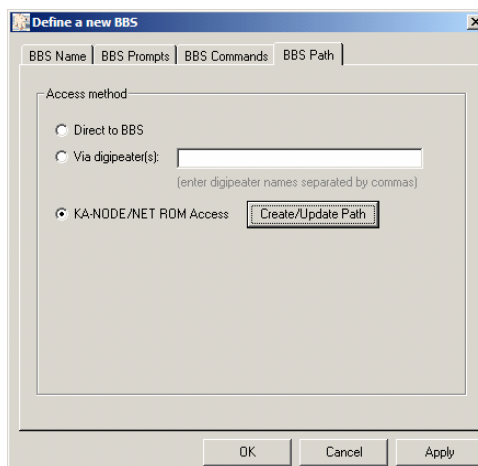
Step 4. – Configure Outpost (and your BBS Setup) for Node Access

Finally, its time to set up the BBS in Outpost for node access.

1. From Outpost's main form, **Setups > BBS**. On the 1st Tab press **New** (if not previously setup). Enter the Friendly name, the connect name, and any description (optional).



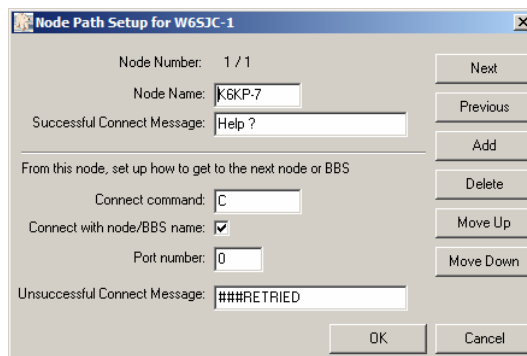
2. Select the Path Type, 4th Tab. Pick the last option for KA-Node/NET ROM Access. Once checked, the **Create/Update Path** button is enabled.



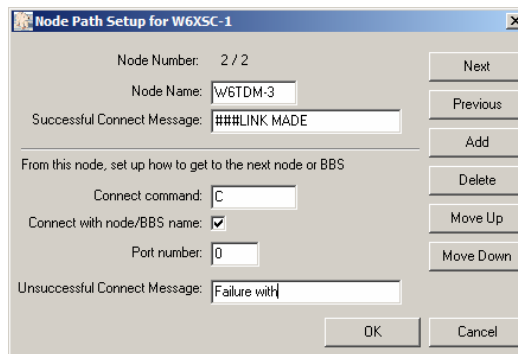
3. Press the **Create/Update Path** button to begin entering the information on the 2 nodes.

The Node Path Setup form allows you to add, delete, move up, and move down a node in the network path order as necessary.

4. Enter the information for the 1st node... K6KP-7. When done, press **Add** to store this entry, clear the form, and get ready to add another node.



- Enter the information for the 2nd node... W6TDM-3.
When done, press **Next** or **Previous** to review the node data that you entered.
NOTE: if the path became marginal and you needed to add another node somewhere in the path, you would add the node (it gets put at the end of the list) and then use the **Move Up** or **Move Down** buttons to position the new node in the right order.



When done, press **OK** to exit Node Path Setup, then press **OK** on the BBS setup form to save the changes.

How it works

- Using the BBS Setup form, we define the BBS named **W6SJC-1**, and the 2 nodes to get to this BBS.
- Using the TNC Setup form, we use the default TNC **Connect** command.
- Both nodes – **K6KP-7** and **W6TDM-3** – are on the same frequency in the path the BBS, so all connect commands will be the standard “**C**” or “**connect**”.
- When we press **Send/Receive**, we connect from our station to **K6KP-7**, then connect to **W6TDM-3**.
- Finally, Outpost sends the last connect command to connects us to the **W6SJC-1** BBS.

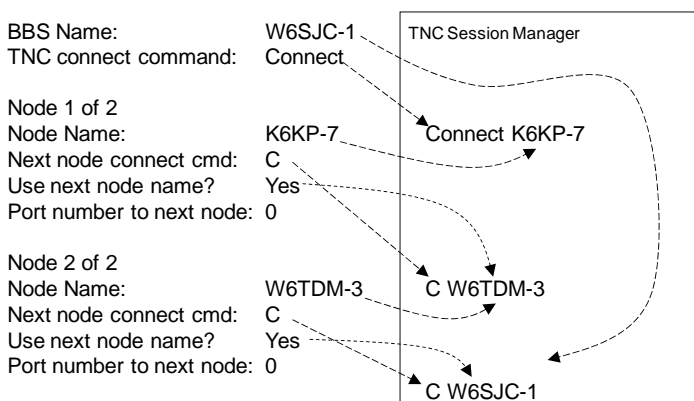
What you set up in Outpost

BBS Name: W6SJC-1
 TNC connect command: Connect

Node 1 of 2
 Node Name: K6KP-7
 Next node connect cmd: C
 Use next node name? Yes
 Port number to next node: 0

Node 2 of 2
 Node Name: W6TDM-3
 Next node connect cmd: C
 Use next node name? Yes
 Port number to next node: 0

How Outpost uses these settings



Step 5. – Testing the Configuration

With all configurations entered, running Outpost will look exactly the same as if you entered all the commands manually. Here’s the output of the run I did with the above configuration, with **Retrieve Private Messages**, **Retrieve New Bulletins**, and **Session Logging** turned on.

```
-----
15-Oct 12:37:22: Outpost v2.1.0 c175: TNC Session #
575
-----
cmd:b
BEACON EVERY 10 min
cmd:Echo on
ECHO was ON
cmd:my KN6PE
cmd:Mon off
MONITOR was OFF
cmd:daytime 061015123723
cmd:connect K6KP-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
connect W6TDM-3
###LINK MADE
connect W6SJC-1
CUPTNO:W6TDM-3} Connected to W6SJC-1
[FBB-7.04j-AB1FHMRX$]
```

```
Hello Jim,
Welcome to The San Jose EOC Mailbox

W6SJC >
LM
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

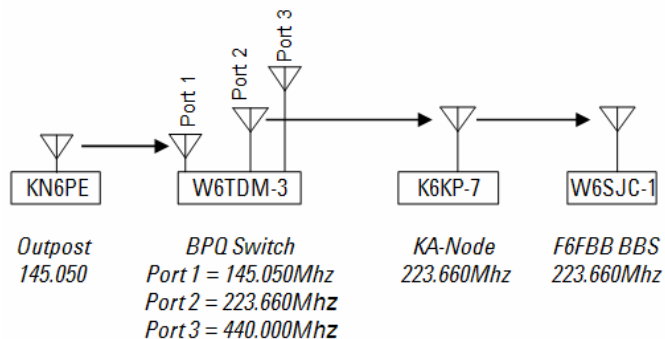
W6SJC >
LB
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

W6SJC >
B

73 Jim,
###DISCONNECTED BY W6TDM-3 AT NODE K6KP-7
*** DISCONNECTED
cmd:
```

Example #2

Let's assume a different configuration of nodes and frequencies to get to the BBS. In this example, the nodes are different (same names, different implementation), and we will switch from one frequency to another by using the "next Port Number" to navigate through a BPQ Switch.



Collect information about the path to the BBS

In this example, I know the following:

KN6PE	My home station; operates on 2 meters, primarily on 145.050.	
W6TDM	W6TDM-3	A BPQ Switch. This station is operating on 3 different frequencies. Once connected to BPQ, you can connect to a different node by issuing the connect "C" command either with or without the Port Number. In this example, we will connect to the BPQ on the 2 meter port, and exit on the 220 port (Port 2).
K6KP	K6KP-7	A KPC3 TNC. This is a standard KPC3 with its KA-Node capabilities enabled.
W6SJC	W6SJC-1	The BBS; it could be any type. For this example, we connect to W6SJC-1.

Here is the Node Path setup that Outpost needs.

	Node Name	Successful connect message	Connect Cmd to next Node/BBS	Connect with the name to the next Node/BBS?	Port Num to the next Node/BBS	Unsuccessful connect message to the next Node/BBS
1	W6TDM-3	###LINK MADE	C	Yes [✓]	2	Failure with
2	K6KP-7	Help ?	C	Yes [✓]	0	###RETIED

How it works

- Using the BBS Setup form, we define the BBS named **W6SJC-1**, and the 2 nodes to get to this BBS.
- Using the TNC Setup form, we use the standard TNC **Connect** command.
- From my station, I press **Send/Receive** and connect to the first node **W6TDM-3** on the 2 meters frequency.
- Because **K6KP-7** is on a 220 frequency and is accessible by **W6TDM-3**, I configured **W6TDM-3** to go out on Port 2. Note that the port number is inserted in the connect command to **K6KP-7**.
- Once connected to **W6TDM-3**, Outpost issues the usual connect command to get to the BBS as usual.

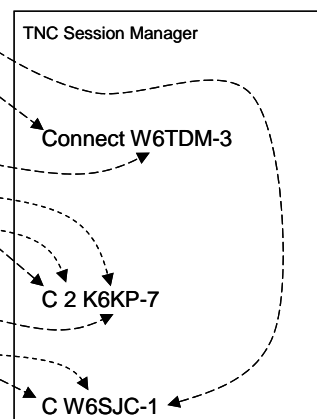
What you set up in Outpost

BBS Name: W6SJC-1
 TNC connect command: Connect

Node 1 of 2
 Node Name: W6TDM-3
 Next node connect cmd: C
 Use next node name? Yes
 Port number to next node: 2

Node 2 of 2
 Node name: K6KP-7
 Next node connect cmd: C
 Use next node name? Yes
 Port number to next node: 0

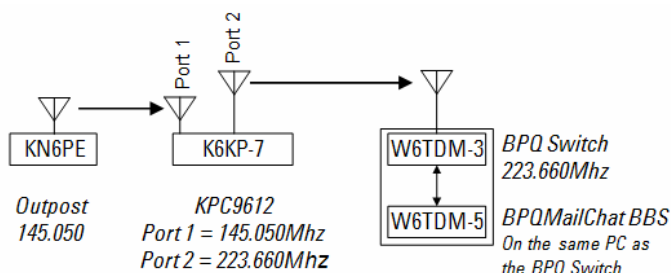
How Outpost uses these settings



Example #3

Let's assume yet another configuration of nodes and frequencies to get to the BBS, but different node hardware. We still need to switch from one frequency to another, but will use a different connect command because of different node hardware.

Additionally, the last node is installed on a PC that is hosting both a BPQ Switch and a BPQMailChat BBS.



Collect information about the path to the BBS

In this example, I know the following:

KN6PE	My home station. Operates on 2 meters, specifically 145.050.	
K6KP	K6KP-7	A KPC9612 dual port TNC. It supports the X or Xconnect command that “cross-connects” you from one port to the other port. In this example, we will connect to this node on the 2 meter port, and want to continue to the next node on the 220 port.
W6TDM	W6TDM-3	A BPQ Switch that operates on the 220 frequency. In this example, the PC that hosts the BPQ Switch also hosts the BBS. Once connected to the BPQ Switch, the “ BBS ” command is all that is needed to connect to the BBS.
W6TDM	W6TDM-5	the name of the BBS.

Here are the Node Path setup entries that Outpost needs:

	Node Name	Successful connect message	Connect Cmd to next Node/BBS	Connect with the name to the next Node/BBS?	Port Num to the next Node/BBS	Unsuccessful connect message to the next Node/BBS
1	K6KP-7	Help ?	X	Yes [✓]	0	###RETRIED
2	W6TDM-3	###LINK MADE	BBS	No []	0	Failure with

How it works

- Using the BBS Setup form, we define the BBS named **W6TDM-5**, and the 2 nodes to get to this BBS.
- Using the TNC Setup form, we use the standard TNC **Connect** command.
- From my station, I press **Send/Receive** and connect to the first node **W6TDM-3** on the 2 meters frequency.
- Because **W6TDM-3** is on a 220 frequency and is accessible by **K6KP-7**, I need to go out **K6KP-7**'s Port 2. The **X** or **Xconnect** command performs the cross-connect that we need.
- Once at **W6TDM-3**, Outpost sends the **BBS** command and we will connect to the BBS that resides on this PC.

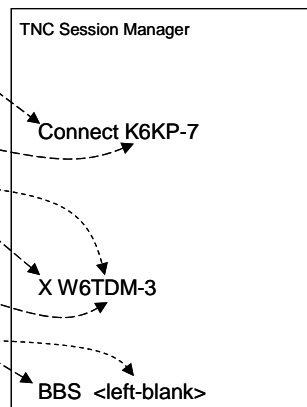
What you set up in Outpost

BBS Name: W6TDM-5
 TNC connect command: Connect

Node 1 of 2
 Node Name: K6KP-7
 Next node connect cmd: X
 Use next node name? Yes
 Port number to next node: 0

Node 2 of 2
 Node name: W6TDM-3
 Next node connect cmd: BBS
 Use next node name? No
 Port number to next node: 0

How Outpost uses these settings



Notes, Considerations, Limitations, etc.

The performance of the KA-Node feature depends on a couple of things:

1. **Quality of the configuration.** Getting the path characteristics defined correctly will go a long way to ensure that Outpost will behave the way you expect. I encourage you to take the time to work through the configuration and confirm it all works for your environment before committing it within your operations.
2. **Quality of the path.** Users will always try to balance the number of nodes in a path to minimize excessive channel traffic with the need to have a guaranteed connection. Some of the best paths to BBSs that I experience during the day literally disappear during the night. Balancing path integrity with channel utilization is something you should consider when using KA-Nodes (or even Digipeaters) in your operations.
3. **Connection Management.** As you can see from the above examples, there is a lot of capability implemented into hardware and software node systems that give the greater packet environment a lot of operational flexibility in terms of connection management. It is important to understand the entire connection path before you start configuring Outpost to use it. My assumption has been: if you can do it manually, you should be able to do it through Outpost (see Limitations below).
4. **Trading off robustness with complexity.** I have had Outpost running KA-Nodes from home for weeks prior to releasing v2.1. By keeping the configuration relatively simple to implement, it does open the door for unexpected behavior and unhandled situations if a node along the path does not behave as anticipated. While I am hoping that the trade-offs between robustness and complexity were the right ones, I will defer to the user community to confirm that the mix is right.
5. **Limitations.** There are 2 known limitations with the Node Path implementation:
 - a. The KA-Node implementation is with AGWPE connecting through a NET/ROM Node. For some unknown reason, the KA-Node I've used for test does not like the way AGWPE sends the subsequent connect request (works fine for serial access). This is an open issue and under investigation. If you are using AGWPE and this configuration is working for you, please let me know.
 - b. Outpost currently does not support a Telnet connection to a BPQ switch (which apparently is allowed) then performing a **BBS** command to connect to a locally resident BBS. Obviously, bringing Telnet into the mix opens the door for someone Telnet'ing to a BPQ Switch node, then wanting to progress by RF to some even further BBS down some RF path. All of this is currently not supported.