

Router Integration

RLS and Telco modules can be set up as

controllers for the SAS 64000 or SAS 32KD router

systems when the *BMXdigital* Server is running

Release 3.24 software. Universal Input modules

connected to the router—although they cannot

control the router, also show the current source

routed to the module in their input source display.

Note that starting with software Release 4.00

SAS routers are no longer supported since the system

is optimized for VistaMax system integration.

To integrate an SAS router with a *BMXdigital* console requires a working knowledge of SAS router systems and the *BMXdigital* Session capabilities. Refer to the appropriate SAS router documentation and to Chapter 4 of this manual for *BMXdigital* Server information.

One complication to integrating an SAS 64000 router is there are two versions: older 64000 routers (those shipped prior to 2002) run SASTerm software (a DOS version of the router control software); newer 64000 routers run RCS software (a Windows version of the router control software). Even though the SASTerm version will work with the *BMXdigital*, SAS recommends that older 64000 routers be updated to run the SAS RCS

software, which all SAS 32KD routers and newer 64000 routers (shipped from early 2002) already run. Contact SAS (818.840.6749) about updating the firmware and software on older 64000 routers. Connection information is presented in this section for units using either SASTerm software or RCS software.

USING THE *BMXDIGITAL* WITH SAS ROUTERS

Each *BMXdigital* input module must first be set as a destination on an SAS router (one of the router's outputs is physically connected to the module's input and the ROUTERS.INI file in the *BMXdigital* Server has an line entry listing that module's SAS router destination).

The *BMXdigital* Server software can handle connections to multiple SAS routers on a single console or a single SAS router tied to multiple consoles. These connections are discussed in detail later in this section.

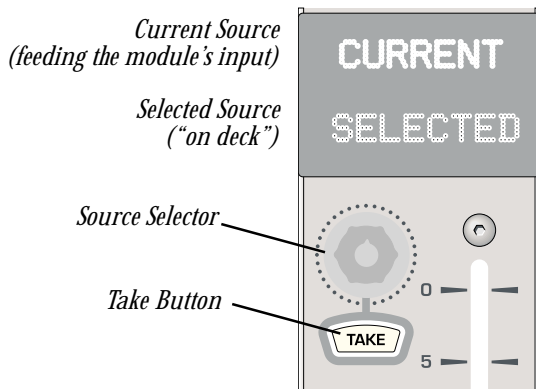
SAS router names are limited to eight characters, which are typically decided upon during the router installation. The source names available to each module are shown in an alphabetical source list as the Source Selector knob is rotated on the RLS or Telco modules.

In terms of their router control functions, the RLS and Telco modules are essentially identical. The RLS and Telco module's Source Selector and Take button function like an SAS remote controller, allowing a user to "dial-up and take" any available signal source from the router.

MODULE SOURCE NAME DISPLAYS

On the next page is a detail view of the RLS and Telco modules' router controls. The ten-character display in the top row shows the current source (CURRENT) that has been "taken" (that router input is now connected to the module). The bottom row display shows the selected source (SELECTED), as chosen by rotating the Source Selector control in order to find the next desired router source.

Telco & RLS Module Router Controls



Source Selector: A rotary encoder that allows any available router source to be shown "on deck." Rotating this control clockwise moves alphabetically up through the source names. Rotating it counterclockwise moves down alphabetically through the names. The selected source is then "taken" by pressing the Take button.

Take Button: When pressed, "takes" the selected or On Deck source (e.g., the input to the module is switched from the current source to the selected source). The selected source name then becomes the current source, and is thus shown in both the top and bottom displays.

The Source Selector and Take button are only active when the module is set as a switcher (refer to the RLS and Telco Quick Guides in Chapter 2 for the DIP switch settings that set the module as a switcher).

Using the TAKE button

Source changes are not permitted while the module is On. This keeps inadvertent source changes from occurring while on-air. When a module is On, and the Take button is pressed, the route change request is not sent until the channel is turned Off. However, the current source name will change immediately while the On button light winks, indicating that the module is pending a change (just as when a session change is done while a channel is On). The eight character source label in the upper display will be bracketed ([SOURCE_2]) during this time, highlighting the pending nature of the current selection.

When the module is Off, pressing the Take button immediately sends a route change request to the SAS router. Depending upon the number of sources and destinations configured in the router, it could take a few seconds for the SAS router to confirm the Take request. During this time, the brackets surround the current source name ([SOURCE_2]), again indicating that the route has not yet occurred. Once the router confirms the route, the brackets are removed from the current source name display.

INHIBITED ROUTES

Sources can be inhibited to specific destinations in the SAS router setup. Inhibits are typically set up to prevent a potential feedback path. If a source is inhibited, the brackets around the current selection display will never clear. This serves as an indication of an inappropriate source selection. The next section describes how to restrict the source selection list so inhibited sources cannot be accidentally selected.

Restricting Source Selections

All 256 names contained in an SAS 64000 system, or the first 256 possible names in the 32KD, can be scrolled through using the Source Selector. However, this is not always a practical way to use a module specific source selector. In many applications only a small fraction of the sources may be required, thus restricting the complete list of sources is desirable. This is done through a session file setting.

Each session file can set a different list of source selections for each router-controller module. For example, a morn-

ing show session file could set the router modules to only show the sources specific to that show, or even to dedicate a source (like a traffic report service) to one module. Then, when the session file for the midday show is loaded, the same router modules could show completely different lists of sources, or any module could always show the same list. The source lists can overlap or differ in any way.

For example, one RLS module may be set to see just the commonly used news sources, a Telco module may be set to only see an ISDN or other remote service requiring a mix-minus feed. Additional RLS modules may be set to select between the available outside feed sources or just the inter-studio sources. One module could also be left unrestricted, for maximum flexibility.

Although Telco modules typically use Salvo commands (detailed later in this section) in order to change both the module input source and the mix-minus return path, they can also function just like an RLS module—as a simple router selector for changing the input source.

ROUTER SOURCE SELECTION NOTES

Anytime a label change is made to the router using the SAS configuration software, the *BMXdigital* Server automatically downloads the new label list. Although this is not a common occurrence, it can cause consternation with the board operator since every router controller module will show UPDATING RTR LABELS in their displays until the new list download is complete.

It's best to warn operators of this change, even though any router signals that are on-air will not be affected by this label updating.

Stereo Linking

Router sources may be stereo linked within the SAS configuration setup. When linked, the labels for the right channel label are typically "masked." Masked sources are not displayed when using the Source Selector, so stereo linking is transparent to the board operator—provided that the SAS router has been configured correctly.

Session Changes

When a new session file is loaded, the current source on all router controller modules is appended to the new session file's source list for each module. This way, the last current source is still available during the next session. If the module is On when a new session is taken, the current source does not change immediately. Instead, the module goes into pending (the On button blinks). Once the module is turned Off, the new list of source names is loaded into the module and the current source changes to that set in the session file.

If an SAS XY controller makes a source selection that affects a module, that externally invoked source name is automatically added to the selected source list as well (for that active session only, it does not affect the stored settings in the session file).

Session Saving

When a session is saved, the current route (or salvo) is saved in the new session file. All session-based list restrictions and other router based settings from the current session are also saved.

ADVANCED ROUTER OPERATIONS

Salvos

SAS routers have a function called Salvo, which is a way to trigger multiple events from a single request. The *BMXdigital* supports this function.

Salvos are particularly useful with a hybrid or codec. By designating a salvo—rather than a standard source selection, for a specific hybrid or codec, not only is that hybrid or codec output routed to the module, but the appropriate mix-minus output from the console can be routed to the return input on the hybrid or codec.

Salvos are specified within the session file, but must be constructed with care and in coordination with the configuration setup of the router.

Refer to the SAS documentation for additional information on creating salvos.

Module Independent Routing Events

So far, the way that modules interact with a single router destination have been discussed. A route change to a module can be made directly, through the source selector and take button, through an SAS XY panel or software panel, or through a change in sessions.

But, it is also desirable to be able to define a particular set of routes or salvos required for a particular show or daypart that is independent of the specific RLS or Telco modules in the console. These are called Module Independent Routing (MIR) events. They are specified in the session file. When the console configuration is changed via a session for a particular daypart, MIR routes can be invoked.

Multiple Routers / Multiple Consoles

Multiple routers can connect to a single BMXdigital console through separate instances of the SAS RCS application (contact SAS about upgrading TCP Server on the older 64000 routers).

Multiple BMXdigital consoles can share a single common router through a common RCS application. When doing this it is important to configure one console in the group to be the first entity to draw the task of requesting labels. A line entry in the ROUTERS.INI file called Labelmaster is used to specify this configuration. This keeps multiple consoles from simultaneously requesting the same data, especially when a network connection is reestablished or when a label change occurs.

CONNECTING A CONSOLE & ROUTER

Pages A-6 and A-7 show the interconnections between the BMXdigital console and the various components in the SAS 64000 and 32KD systems using the SASTerm and RCS applications.

But, before physically connecting anything to the SAS system, the DIP switches on the RLS and Telco modules that will be router controllers must be set for router control. The reason is that their displays show system status messages as well as source selections. To set the modules for router control, set these DIP switches as follows.

On an RLS Module:

DIP switch	#	Setting
DS1	4	ON
DS1	5	ON

On a Telco Module:

DIP switch	#	Setting
DS2	4	ON
DS2	5	ON

The next step to integrating the console with an SAS router is to connect the SAS server computer and the BMXdigital console together. This is done through the BMXdigital LAN connection (the DATA connector on the Session module, as shown on pages A-6 or A-7).

Use a standard CAT-5 cable to tie the console to a hub, or a crossover cable to tie the console directly to the SAS server computer.

Communications Initialization

When first connecting to a router, the BMXdigital goes through a four step initialization process. If everything is configured correctly, the first three happen in quick succession. If things are not configured correctly, then one of these steps will be continuously shown.

Refer to the following section on Router Integration for the specific model router being installed for additional configuration information.

Step 1 - Finding the host computer

As this occurs, each router module displays:

A rectangular display screen with a grey background and a thin black border. The text "IP ADDRESS" is on the top line and "SEARCH ..." is on the bottom line, both in a white, pixelated, monospace font.

If this display persists longer than a few seconds, it indicates that the *BMXdigital* Server can't find the Router's host computer. Check that the DATA cable is plugged in properly (and is of the right type—straight thru CAT-5 for a hub, cross-over CAT-5 for direct connection to the SAS server computer), and that the IP address setting of the computer matches the IP address setting in the ROUTERS.INI file on the *BMXdigital* Server.

Step 2 - Finding the SAS TCP Server Application

As this occurs, each router module displays:

A rectangular display screen with a grey background and a thin black border. The text "SAS SERVER" is on the top line and "SEARCH ..." is on the bottom line, both in a white, pixelated, monospace font.

If this display persists, it indicates that the *BMXdigital* Server cannot find the SAS TCP Server application (for older 64000 running SASTerm) or the RCS application (for 32KD and newer 64000). Make sure that SAS TCP Server or the RCS application is running and that it is correctly configured. Since some networks block unauthorized TCP access, consult with the local IS staff to see if that could be causing the problem.

Step 3 - Finding the SAS Router

As this occurs, each router module displays:

A rectangular display screen with a grey background and a thin black border. The text "ROUTER" is on the top line and "SEARCH ..." is on the bottom line, both in a white, pixelated, monospace font.

If this display persists, then the SASTCP Server

or RCS application cannot find the router on the computer's serial port. Check the serial connection. An incompatible revision of MCU firmware can also cause this search to fail (see the Installation Note below).

Step 4 - Downloading Source Labels

As this occurs, each router module displays:

A rectangular display screen with a grey background and a thin black border. The text "UPDATING" is on the top line and "RTR LABELS" is on the bottom line, both in a white, pixelated, monospace font.

This could take up to 20 seconds to complete, depending upon the number of sources listed in the router. When any label on the router is changed, it causes the *BMXdigital* Server to automatically update the labels to each module again. Fortunately this does not typically happen very often once the system is completely configured.

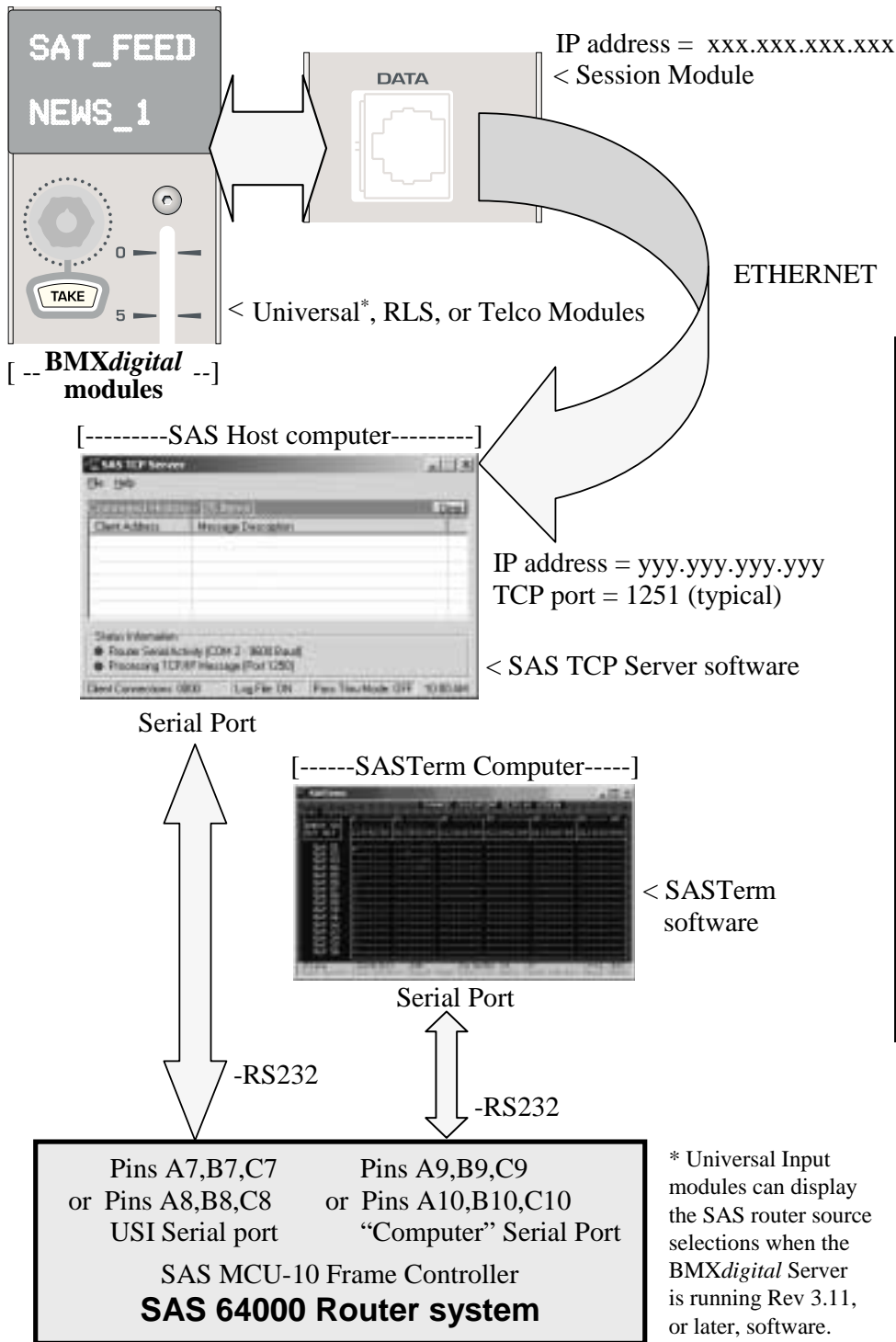
SAS ROUTER INTEGRATION—USING

INSTALLATION NOTE:

This integration was designed and tested using Rev 3.00 of the SAS MCU firmware, and Rev 2.02 of the SAS TCP Server software.

Contact SAS regarding firmware revisions that are compatible with this integration.

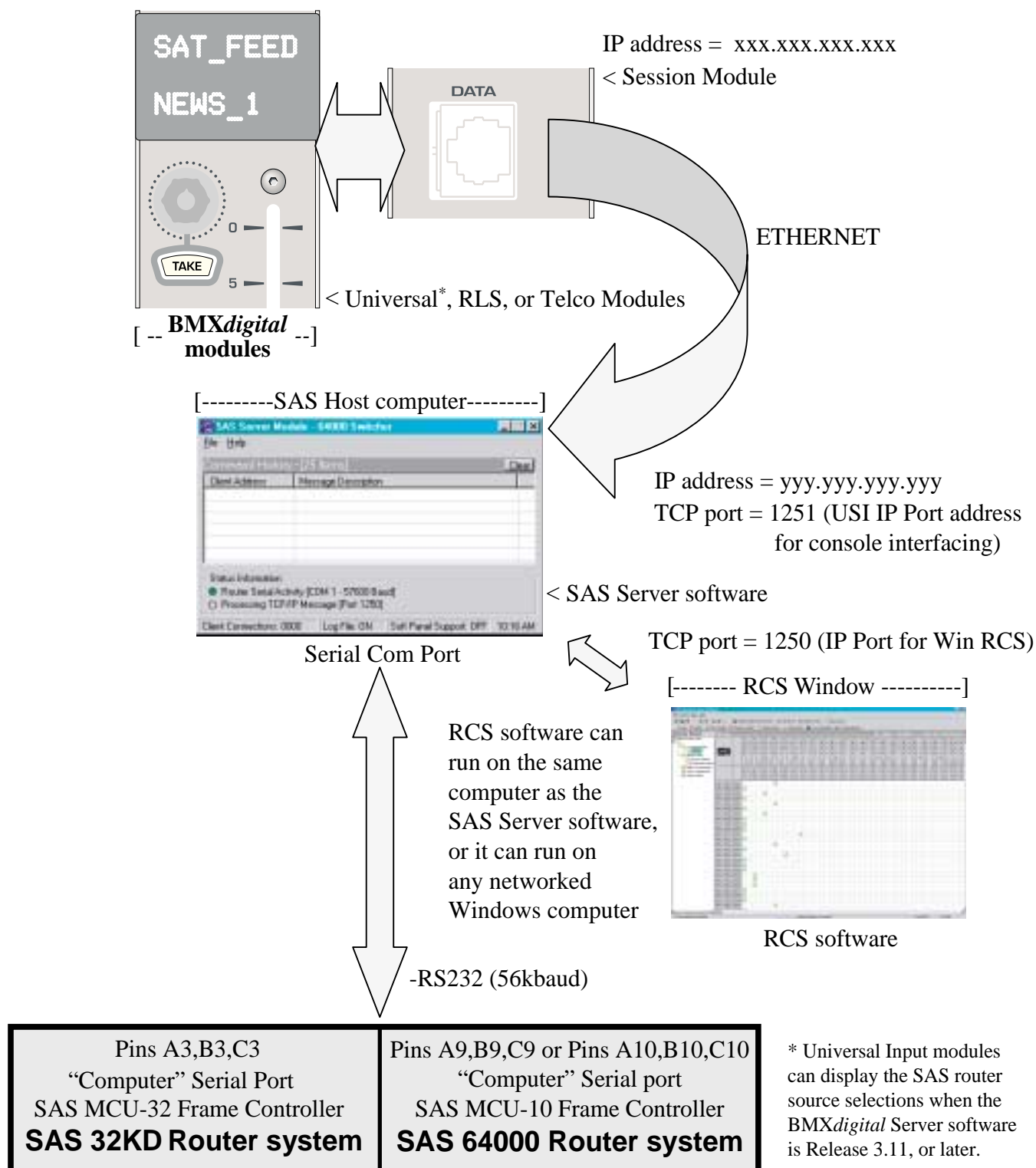
When configuring the SAS TCP Server Software, make sure that Pass Thru Mode is set to OFF, and that Log File creation is also OFF.



Hint: A quick test of the P C > Network > S A S Server > Router connection can be done by typing: "Telnet <ComputerName or IP_address> 1250" at a command prompt. This returns the message: "Login Successful" when the connections are correct. Pressing the ENTER key will get the response "SAS READY" from the router when it is connected correctly.

Be sure to close the Telnet session before connecting the BMXdigital to the SAS Server.

Communications Connections Between the BMXdigital Console and an SAS 64000 Router Running the SAS Term Software



Communications Connections Between the BMXdigital Console and an SAS 64000 or 32KD Router Running the RCS Software

Appendix A-7

HARRIS CORPORATION

SASTERM SOFTWARE

Required Materials

- SAS 64000 Manual
- SAS SASTerm Software
- SASTCP Server Software
- Spare serial port on the SAS computer

Installation consists of four elements:

1. Connectivity. Confirm the following are working or set correctly: the network connection between the BMXdigital and the SAS computer; the serial connection between the SAS computer and the MCU-10 port on the 64000 (as shown on page A-6); the DIP switches on the router modules are set correctly (see page A-4).

2. Software/Firmware Versions. Confirm that the BMXdigital Server software is Version 3.10 or later; that the SAS MCU firmware revision is Version 3.00 or later; that the SAS TCP Server Software is Version 2.02 or later and that the Pass Thru Mode and Log file settings are both Off.

3. ROUTERS.INI Configuration. This file must be edited as shown on pages A-9 to A-12. After editing this file, the BMXdigital Server must be reset by power cycling the console for these changes to be recognized. After this is done, observe the four-step communications initialization process, as previously shown.

4. Session File Configuration. Each Session File will need to be configured for the router control modules. Details on doing this are also on the following pages.

SAS ROUTER INTEGRATION—USING RCS SOFTWARE

Required Materials

- SAS 64000 or 32KD Manual
- SAS RCS Application Software
- SAS Server Module Software
- Spare serial port on an the SAS computer

Installation consists of four elements:

1. Connectivity. Confirm that the following are working or set correctly: the network connection between the BMXdigital and the SAS computer; the serial connection between the SAS computer and the MCU-10 port on the 64000, or the MCU-32 port on the 32KD Frame (as shown on page A-7); the DIP switches on the router modules are set correctly (see page A-4).

2. Software/Firmware Versions. Confirm that the BMXdigital Server software is Version 3.10 or later; that the SAS MCU firmware is Revision 3.02 or later; that the SAS RCS Software is Version 1.41 or later.

3. ROUTERS.INI Configuration. This file must be edited as shown on pages A-9 to A-12. After editing this file, the BMXdigital Server must be reset by power cycling the console for these changes to be recognized. After this is done, observe the four-step communications initialization process, as previously shown.

4. Session File Configuration. Each Session File will need to be configured for the router control modules. Details on doing this are also on the following pages.

ROUTERS.INI file

This file identifies each router associated with a BMXdigital console and defines which router destination is connected to which Universal, RLS, or Telco module. The ROUTERS.INI file is stored in the BMXdigital Server storage card (Storage Card/Data/SysFiles/).

ROUTERS.INI File Sections

[Routers]

This section (required) identifies each router connected to the console by a number (Type_1, Type_2, Type_3, etc.). The entries in this section list the router's host name, IP address and port, and whether it is the label master. Here's a typical entry for the [Routers] section:

```
[Routers]
Type_1=sas32kd
Hostname_1=Sserver
HostIP_1=137.237.207.101
Port_1=1251
labelmaster_1=0

Type_2=sas64k
Hostname_2=sc019753
HostIP_2=137.237.207.102
Port_2=1251
```

Entries within the [Routers] section:

Type_n=router_id

Type_n, with *n* being a positive integer that always starts with 1, simply numbers each router. **router_id**, at this time, will be either "sas64k" (for a 64000 Router) or "sas32kd" (for a 32KD Router). Other Router ID codes will be added in the future as additional routers are supported. Currently, SAS Router systems are supported through the SAS TCP Server or SAS Server Module. Routing matrices of up to 256 x 256 are supported.

Hostname_n=hostname

Hostname_n, where *n* is the same positive integer (starting with *n=1*) used in the line:

Type_n=router_id.

hostname is the computer name of the networked computer hosting the SAS TCP Server or SAS Server Module. In the example, the name of the servers are: Sserver and sc019753. If the host name is not found, the BMXdigital Server looks for the IP address in the next line.

HostIP_n=yyyy.yyyy.yyyy.yyyy

HostIP_n, where *n* is the same positive integer (starting with *n=1*) used in the line:

Type_n=router_id

yyyy.yyyy.yyyy.yyyy is the computer IP address of the networked computer hosting the SAS TCP Server or SAS Server Module. In the example, the IP address for the servers are: 137.237.207.101 and 137.237.207.102.

Port_n=m

Port_n, where **n** is the same positive integer (starting with **n=1**) used in the line:

Type_n=router_id

m is the IP port address for the SAS TCP Server (typically 1250) or the USI IP port address for the SAS Server Module (typically 1251). The default value is 1251. Refer to the documentation for the SAS TCP Server or Server Module for additional information on setting this variable.

LabelMaster_n=b

LabelMaster_n, where **n** is the same positive integer (starting with **n=1**) used in the line:

Type_n=router_id

b is either 1 (true) or 0 (false). If this entry is missing, the default value is 1, true. When this entry is 0 (false), this entry delays the BMXdigital from requesting label data from the router for 30 seconds. Only one BMXdigital per router should be the Labelmaster. All others should be set to 0.

[Router Control]

This section (required) associates each router-designated console module to a particular router, and to a specific output (destination) on that router. Modules are identified, in Session files and in the ROUTER.INI file, by a channel number rather than as a Universal Input, Telco, RLS, etc. Channel numbers identify a specific module by either its physical location in the mainframe (Universal Inputs and RLS modules) or by its ID number (Telco modules). Telco and RLS modules are further specified by the type of input signal (Direct, External RLS, Router). A table listing of the Channel Number Assignments is on page 4-7.

With that in mind, here's a typical entry for the Router Control section:

```
[Router Control]
```

```
Router_129=1
```

```
Output_129=11
```

```
Router_130=1
```

```
Output_130=13
```

```
Router_131=1
```

```
Output_131=21
```

```
Router_132=1
```

```
Output_132=15
```

```
Router_81=2
```

```
Output_81=15
```

```
Router_82=1
```

```
Output_82=17
```

```
Universal_10=1,1
```

```
Univoutput_10=25,27
```

Entries within the [Router Control] section:

Router_n=m

Router_n is a positive integer indicating the type and the number of a specific module. For an RLS module used with a router, **n** would be between 129 and 143, indexed as installed, left to right, in the console. For Telco modules used with a router, **n** is 81 for Telco 1, 82 for Telco 2, and so on up to 86 for Telco 6. The Telco ID is set via DIP switches DS1-1 thru DS1-6 on the Telco modules. Again, a table listing these Channel Number Assignments is on page 4-7.

m is the Router Type number, given under the Routers section, where this module is connected.

Output_n=m

Output_n is the module identification number. It is the same number **n** as the **Router_n=m** number.

m identifies the specific audio output port on the router connected to the associated RLS or Telco module. **m** can be numbered from 1 to 256. When a stereo pair is connected, always enter the number for the left channel (the odd number) of the stereo linked pair.

When using SASTerm software, the Alpha Names for the right channels are typically masked using a left hand brace symbol"{" as the first character in the label name. Stereo links are defined through the configuration of the SASTerm software. Consult an SAS manual for additional details on stereo linking and naming.

Universal_n=a,b

Universal_n indicates the number of the Universal Input module that connects to a router output. **n** must be between 1 and 63 (see Channel Number Assignments Table on page 4-7).

a,b indicate the Router Type numbers (listed under the Routers section) for that module's A and/or B inputs.

Note: The comma is a vital part of this statement. If a router only connects to the A input, then the statement must be written as: **Universal_n=a,** If a router only connects to the B input, then the statement must be written as: **Universal_n=,b**

Univoutput_n=a,b

Univoutput_n is the same module identification number used in **Universal_n=a,b**.

a,b identifies the audio output ports on the respective router(s) that connect to the associated Universal input module's A and/or B inputs.

Note: The comma is a vital part of this statement, even if only one input is connected to a router. When only the A input connects to a router then the statement must be written as: **Univoutput_n=a,** When only the B input connects to a router then the statement must be written as: **Univoutput_n=,b**

ROUTERS.INI File Content Example:

The following example shows three routers;

- 2 SAS 64000 routers
- 1 SAS 32KD router

used with 4 console modules integrated with the router;

- 2 RLS modules
- 1 Telco module
- 1 Universal Input module

This is a rather intricate arrangement of three routers. The first two routers use a common host computer, running two instances of the SAS TCP Server application software, each instance is set to separate network port numbers. (Separate serial communication ports would be needed too.) The third router uses a different host computer, multiple consoles are connected to this router, this console is not the labelmaster.

Example, ROUTERS.INI File

```
[Routers]
Type_1=sas64k
Hostname_1=Sserver1
HostIP_1=137.237.207.101
Port_1=1251
```

```
Type_2=sas64k
Hostname_2=Sserver1
HostIP_2=137.237.207.101
Port_2=1251
```

```
Type_3=sas32kd
Hostname_3=Sserver2
HostIP_3=137.237.207.102
Port_3=1251
Labelmaster_3=0
```

```
[Router Control]
Router_129=1
Output_129=31
```

```
Router_130=2
Output_130=33
```

```
Router_81=3
Output_81=55
```

```
Universal_1=3,
Univoutput_1=57,
```

Line-by-line Explanation:

All routers connected to the console are identified in this section.

The first Router (Type_1) is an SAS 64000 (sas64k).

Its Host PC's name is Sserver1.

Its Host PC's IP address is 137.237.207.101.

Its Network Port uses the default value (1251).

The second router (Type_2) is also an SAS 64000.

It uses the same PC as router 1, hence the Host PC name and IP address are the same as for the first router.

It uses Network Port 1251.

The third router (Type_3) is an SAS 32KD (sas32kd).

Its Host PC's name is Sserver2.

Its Host PC's IP address is 137.237.207.102.

Its Network Port uses the default value (1251).

This console does not request labels immediately from this router.

All router-controller modules are identified in this section.

This module is the first router/switcher RLS module (Router_129). It connects to router 1 (Router_129=1), output 31 (Output_129=31).

This module is the next router/switcher RLS module (Router_130). It connects to router 2 (Router_130=2), output 33 (Output_130=33).

This module is Telco 1 (Router_81). It connects to router 3 (Router_81=3), output 55 (Output_81=55).

This module is Universal Input 1 (Universal_1). Its A input connects to router 3 (Universal_1=3), output 57 (Univoutput_1=57).

SESSION FILE ENTRIES SPECIFIC TO ROUTER OPERATION

Session files can have entries to configure router operations. There are two general classes of router entries:

• Module-Specific Entries

These entries establish module specific selection specifications, routes, or salvos to be established when a particular session is invoked. This is the most common type of router control entry, as it directly associates an RLS or Telco module to a router and a specific output.

• Module-Independent Entries

These entries establish routes and salvos that are not specific to a module, but are established when a particular session is invoked. The use of this type of entry is far less common than the Module Specific entry, as it deals with route changes that are not associated with any specific RLS or Telco modules.

Module-specific entries for linking a module to a router and a specific output were covered in the previous section. Here are additional entries used within the [Router_n] lines, under the [RouterControl] section:

Include=a-b,c-d,e,f, . . .

a-b and **c-d** represent ranges of SAS router sources which are allowed to be seen on this module's input source display. The range can span both even and odd numbered sources, even when linked as stereo pairs. Masked labels are always excluded from the include list. If the Include entry is missing, then all 256 unmasked sources will be alphabetically displayed as the source selector is turned.

e,f, . . . represent individual SAS router sources which are to be shown on this module's input source display.

Example:

```
Include=17-32,37,39
```

This entry says: include inputs 17 up to 32, 37, and 39. Note that any, or all of these inputs could be stereo-paired. The stereo linkage information is defined through the SAS software configuration. Please consult the SAS manual for additional details.

Salvo_n=salvname,a

Salvo_n is a positive integer, starting with 1, to number the salvo command specific to this module.

salvname is the display name for this salvo command.

a selects the Salvo number as defined in the SAS configuration SALVO NUM.

Example:

```
Salvo_1=Atlanta,2  
Salvo_2=Boston,3  
Salvo_3=Chicago,4
```

These three salvos could be set up to route three different telco or codec sources to a specific Telco module in coordination with routing its mix minus output back to the selected telco or codec. Salvo names are limited to eight characters.

Salvo labels are not provided for in the SAS communication protocols. This limitation is actually a blessing in disguise. Setting the label here allows the label to clearly reflect the source's name at the module, regardless of any otherwise confusing actual salvo label.

When a salvo is invoked, its name appears in the upper display (enclosed in [], brackets) until the router responds with the input source name connected to the module. When the router responds, confirming a route change, the current display changes to the source name. Thus, using the same name as the actual source for the salvo would be an intuitive choice.

Caution: This entry limits the module display to show salvo labels only. Any "include" specifications are ignored.

This entry also prohibits the carry-over of source selections from an earlier session to the current selection list. Likewise, it prohibits the names of sources invoked from external XY controllers to be appended to the selection list. Modules that are specified to execute salvos within a particular session file do so exclusively.

Take=*n* or Take=Salvo_*n*

n is the SAS output source number which is to be provided on this module when this session loads.

Salvo_*n* is the requested salvo as previously numbered and labeled.

Example:

Take=15 (defines source 15)

Take=Salvo_3 (defines Salvo_3)

Note: These entries are automatically created when a session is saved, so they rarely need to be edited by hand. When saving a new session, this entry is updated to reflect the current source (or Salvo) at the time the session is saved.

Module Independent entries

These entries establish routes and salvos that are not specific to a module, but are established when a particular session is loaded. The use of this type of entry is far less common than the Module Specific entry, as it deals with route changes that are not associated with a particular RLS or Telco module.

[RouterCommand_*n*]

This section defines a non-module specific entry section where ***n*** is a positive integer starting with *n*=1, identifying a specific router. Entries within the [RouterCommand_*n*] section include:

Salvo_*n*=*a*

where ***n*** is a positive integer, starting at 1, to number the salvo commands issued when this session is loaded.

a is the Salvo number, as defined in the SAS configuration setting, SALVO NUM.

Example:

Salvo_1=2

Salvo_2=3

Salvo_3=4

Note: This structure should only be used for salvos that do not have a direct impact on the routing of an RLS or Telco module.

Take_*n*=*a*,*b*

where ***a*** is the router input number and ***b*** is the router output number

Example:

Take_1=10,1

Take_2=20,2

Note: This structure should only be used for routes that do not have a direct impact on the routing of an RLS or Telco module.

Here's an excerpt from a session file, to show how these additional sections are used:

Example, excerpt from Session file	Line-by-line Explanation:
<pre>[RouterCommand_1] Salvo_1=2 Salvo_2=5 Take_1=11,3 [RouterCommand_2] Salvo_1=1 Take_1=21,3 Take_2=23,5 Take_3=35,7 [Router_129] Take=16 [Router_130] Include=17-32,55,57,59 Take=19 [Router_131] Include=17-32 Salvo_1=Denver,1 Salvo_2=Encino,2 Take= Salvo_2 [Router_81] Salvo_1=Atlanta,1 Salvo_2=Boston,2 Salvo_3=Chicago,7 Take=Salvo_3</pre>	<p>Line-by-line Explanation:</p> <p>This is a module independent event for Router 1. Perform actions as set as SALVO NUM 2 in the SAS software. Perform actions as set as SALVO NUM 5 in the SAS software. Route source 11 to destination 3.</p> <p>This is a module independent event for Router 2. Perform actions set as SALVO NUM 1 in the SAS software. Route source 21 to destination 2. Route source 23 to destination 5. Route source 35 to destination 7.</p> <p>First RLS router module ([Router_129]) Set the current source to output 16 and list all labels provided from router (no include statement).</p> <p>Second RLS router module ([Router_130]) Restrict the source selections to inputs 17-32, 55, 57 and 59. Set the current source to output 19.</p> <p>Third RLS router module ([Router_131]) <i>This include list is ignored</i> <small>see note 1</small> Label SALVO NUM 1 as "Denver" Label SALVO NUM 2 as "Encino" <i>When session loads, set Salvo_2 (Encino) as current source.</i></p> <p>Telco module 1 ([Router_81]) <i>Restrict to salvo list only</i> <small>see note 2</small> Label SALVO NUM 1 as "Atlanta" Label SALVO NUM 2 as "Boston" Label SALVO NUM 7 as "Chicago" <i>When session loads, set Salvo_3 (Chicago) as current source.</i></p> <p><small>note 1</small> <i>Salvo operations ignore include statements.</i></p> <p><small>note 2</small> <i>Normally, the absence of an include statement causes every source to be listed, however, Salvo operations restrict the selection to the salvo list only.</i></p>

SOFTWARE NOTES

In addition to the "software hooks" for a Universal Input module to function in conjunction with a router output, version 3.11 software adds access control and proprietary FTP commands. These are detailed in this section.

ACCESS CONTROL

Username & Password—Even though FTP access can be controlled using a Username and password, we suggest that password security NOT be used—unless there is a real need to do so. BMX-

digital is reasonably secure, provided that the BMX*digital* is not exposed to the Internet through a gateway and that an adequate firewall is in place.

While this is typically enough security, this practice does not stop an inquisitive or malicious user inside the firewall. Thus, one username and password may be specified in the NQX.INI file. If neither username or password is specified in the file, then FTP access is open.

If the username and password are specified, then only those with the correct username and password have access to the BMX*digital's* FTP server.

Caution: If the username or password is lost, one will have a great deal of trouble recovering access. See page A-18 for recovery instructions.

To set up password security, add a **USER** entry into the [BMXdigital] section of the NQX.INI file. Please note that neither the username nor password are case sensitive. Here is an excerpt from the NQX.INI file:

```
[BMXDigital]
; A Unique device name here.
NAME = BMXdigital
;-----
; FTP Security, Username,password
; If absent, no password is
; required.
USER = Scotty,BeamMeUp
;-----
```

The BMXdigital has a Telnet access for setting system time, date, and other special functions. But, this service is disabled at system start up. Activating Telnet service can only be done after access to the FTP server has been achieved.

PROPRIETARY BMXdigital FTP COMMANDS

The FTP subcommand language is extendable with the FTP “literal” command that sends arguments, verbatim, to the BMXdigital FTP server.

Four specific arguments have been established to extend proprietary BMXdigital control via FTP protocol. These proprietary commands are:

- **Reset**
- **TelnetON**
- **TelnetOFF**
- **<SessionName.ses>**

Reset: This command resets the BMXdigital. This can be used to remotely restart the BMXdigital Server in advance of loading new software.

TelnetON: This command enables access to the telnet control protocol.

TelnetOFF: This command disables access to the telnet control protocol.

<SessionName.ses>: This commands loads a session file (any session file in the Sesfiles folder on the BMXdigital Server).

Note: These commands are not case sensitive. Session names must not have embedded spaces.

When a session file is loaded remotely, the current session line will display <<REMOTE>> while the session is loading. Once the session has loaded, the top line changes to display the current session file name.

Shown below is an example of a command line-initiated ftp access. This example loads a session file named **DriveTime.ses**.

```
ftp> open bmxdigital
Connected to bmxdigital.
220 WINCE FTPServer (Version 1.04).
User (bmxdigital:(none)): scotty
331 Password required for SCOTTY.
Password:*****
230 User SCOTTY logged in.
ftp> literal drivetime.ses
250 DRIVETIME.SES: Loading ...
ftp>
```

Use of the “literal” extension allows the following commands to be entered:

```
ftp> literal Reset
ftp> literal TelnetON
ftp> literal TelnetOFF
ftp> literal SesName.ses
```

Windows 2000 provides an easy FTP command line access (other operating systems have FTP access methods as well). Consult the Windows 2000 help for more information on FTP commands.

Through command line access, these and other ftp commands can be easily be activated through batch files. The batch files can be activated through icons, and called by other programs. Through the use of links (shortcuts), the appearance or transparency of these batch files can be customized.

FTPCMD.BAT

To aid in the process of logging on to a secured BMXdigital Server, a batch file utility program, FTPCMD.BAT, is listed below. To use the FTPCMD.BAT utility, a command line entry, or a batch file may be used.

Adjacent are four examples of invoking this batch file using a command line entry with a user named "Scotty" and a password of "BeamMeUp." Note that even if FTP security is not enabled, the username and password are

still required to run the batch file.

In these examples, the BMXdigital Server is using the default bmxdigital name, its IP address is 192.168.100.22, and the session file to load is DriveTime.ses.

```
Ftpcmd Scotty BeamMeUp Reset bmxdigital
-----
Ftpcmd Scotty BeamMeUp TelnetOn bmxdigital
-----
Ftpcmd Scotty BeamMeUp TelnetOFF 192.168.100.22
-----
Ftpcmd Scotty BeamMeUp DriveTime.ses bmxdigital
```

FTPCMD.BAT Batch File

```
@echo off
REM *****
REM   FTPCMD.BAT
REM   This batch file logs on to a BMXdigital Server
REM   and executes a session file, performs a reset,
REM   enables or disables telnet access.
REM   This utility is crafted for use under Windows2000,
REM   as this mechanism of passing variables to the ftp client
REM   is specific to the Windows 2000 ftp command processor.
REM *****
echo   FTP Command Launcher System Utility, Version 1.00
echo   Copyright (c) 2001 Harris Corp / Pacific Design Center

rem 1st arg is Username
rem 2nd arg is password
rem 3rd arg is session file name or special command
rem 4th arg is device name or IP address

if "%1" == "" goto usage
if "%2" == "" goto usage
if "%3" == "" goto usage
if "%4" == "" goto usage

:runit
rem === build temp cmd file =====
echo open %4 >tempfile.txt
echo user %1 %2 >> tempfile.txt
echo literal %3 >> tempfile.txt
echo quit >> tempfile.txt
rem =====
ftp -i -n -s:tempfile.txt
rem =====
del tempfile.txt

if "%5" == "" goto end
rem if %5 exists the batch file pauses for debugging purposes

pause

goto end

:usage
echo =====
echo Argument Usage was incorrect. Please see below:
echo =====
echo   "FTPCMD [Username] [Password] [argument_a] [Devicename]"
echo   Where argument_a is:
echo   "[Session_file_name |Reset |TelnetON |TelnetOFF]"
echo   Caution: no spaces are allowed in Session_file_name
echo =====
pause
goto end

:end
```

To recover the username, password, or IP address from a *BMXdigital* Server, the following hardware items are required in order to connect to the Single Board Computer (SBC) on the Session module: a Windows computer running HyperTerminal, a DB9 Port Connection Cable, and a Null Modem Cable.

1. Remove the Session Module from the console.
The console may remain turned on, as hot swapping of modules is permitted.
2. Connect a DB9 Port Connection Cable (male DB9 to 2x5 header) to the HDR4 COM1 header on the SBC. The SBC is the PCA toward the board operator end of the module that does not have a motherboard connector. The HDR4 header is right below the Timer Control label on the Session module faceplate. Orient the flat-cable so that its red stripe is toward pin 1 on the header (which is toward the board operator end of the module).
3. Connect a DB9 female to female null modem cable to the Port Connection Cable and to a DB9 serial port on a Windows computer.
4. Start HyperTerminal (a Windows Communications Accessories program) and establish a new connection using these Port Settings:



5. Reinstall the Session module into the console with the flat cable feeding up between the Session module and the Control room module.
6. As the *BMXdigital* Server starts up, the terminal will show multiple screens of information. The IP address will be listed within the NQXcompare section of the data being fed to the serial port as the SBC starts up. Use the HyperTerminal scroll feature to re-display this information as required.
7. Once the IP address is known, use Internet Explorer or other FTP program, on a computer networked with the *BMXdigital*, to view the Storage Card folder contents using this address: ftp://<IP address>/storage card/.

DB9 female to female null modem cable
(available from Cables N Mor,
<http://www.cablesnmor.com/null-modem-cable.html>)