Advanced Simulation Technology inc.
441-A Carlisle Drive
Herndon, Virginia 20170 U.S.A.
Tel. (703)471-2104 • Fax. (703)471-2108
www.asti-usa.com

ASTi
Telestra Quick Start
User Guide

Document: DOC-01-TELS-UG-1
Telestra v1.4

Advanced Simulation Technology inc. 441-A Carlisle Drive, Herndon, Virginia, 20170 USA
Revision D (April 2001)

TABLE OF CONTENTS

Overview

For domestic (U.S.) customers, Telestra comes with all required software pre-installed. This includes the Telestra federate, the DMSO NG RTI, the rtiexec (to be used when testing Telestra without another machine running the rtiexec), and certain support and debug facilities. For foreign (non-U.S.) customers, the Telestra system is shipped without the DMSO RTI. Please refer to the section entitled “Installing the DMSO RTI” for instructions and other information regarding this topic. It is also important to note that significant information can also be determined from the DACS node associated with the Telestra unit and hence this guide will reference both units as appropriate.

This guide assumes the user has a basic understanding of the High Level Architecture (HLA), and should be familiar with such terms as federate, federation, RTI, etc. It also assumes a basic familiarity with UNIX, such as accounts, passwords, basic commands, etc. If you are familiar with UNIX in general but not Linux in particular, there are some useful commands and tools specific to Linux described at the end of this document.

This guide is intended to assist the user in initial start-up of the Telestra unit and provides an overview of the basic commands and facilities of the system.

Telestra documentation is also available on ASTi’s website http://www.asti-usa.com/hla.

System Accounts

The Telestra system is shipped with two accounts.The administration/super user account name is “root” with a default password of “abcd1234”. This account is used for system maintenance, and will be used to configure the system.

The second account is a normal user account. The account name is “hlauser” with a default password of “HLA!now!”. This account is used to run the Telestra federate.

To change the password of either account, log into the account and type:

```
passwd
```

and follow the directions printed on the screen. ***DO NOT*** lose the passwords, particularly the root password. If the root password is lost, the system hard drive may need to be completely rebuilt!

Preparing Telestra for Use

Standard networking hardware should be used when making connections to Telestra and the DACS units (hubs, bridges, etc.). There are three ethernet interfaces on each Telestra system (HLA, DACS, and Host) used for communicating with Telestra. The location of each interface varies with chassis type. The figures below show the location of each interface.

Figure 1a. Ethernet Interfaces on a Telestra System without built-in CD-ROM

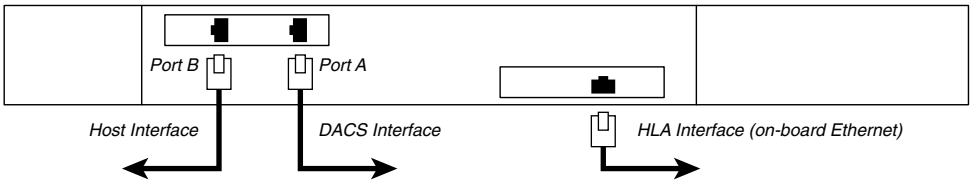


Figure 1b. Ethernet Interfaces on a Telestra System with built-in CD-ROM

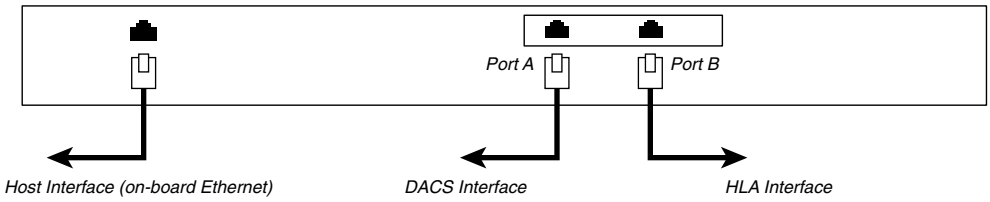
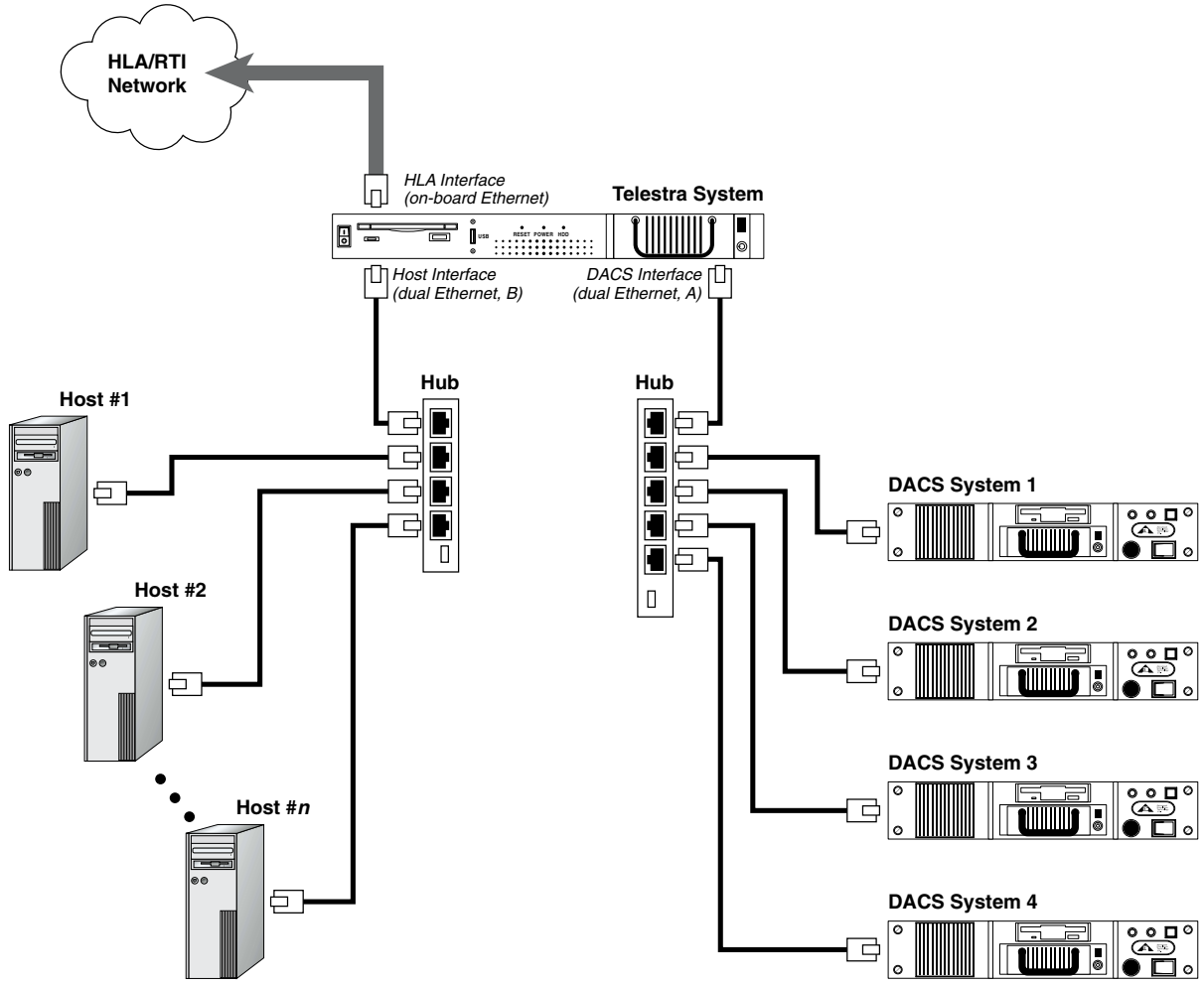


Figure 2 (below)shows an example network topology for the Telestra system. Notice that an individual Telestra system contains support for a maximum of four DACS boxes.

Figure 2: Example Network Topology for the Telestra System



Please note that in order to integrate Telestra into your network, certain configuration set-up procedures are required to set the IP address of the HLA Ethernet adapter and Host Interface adapter. The second Ethernet adapter, used to connect to the DACS node, is pre-configured on shipment, and should not be modified unless under ASTi instruction. Instructions for setting the IP address and hostname of the Telestra system are given in the “Telestra Setup Procedure” section.

A mouse (3 button PS/2 - supplied), keyboard (supplied) and monitor (not supplied - available to order separately) must be connected to the unit in order to configure it.

Once the system is configured, it can be run and shutdown remotely. Instructions for this are described in the document Telestra Remote Control Interface User Guide.

Starting and Stopping the Telestra System

The Telestra system runs the Linux operating system and therefore must be started and more importantly shutdown in the correct manner in order to maintain file system integrity. To start the unit, apply power via the front mounted on/off switch. The unit will boot-up and run the Telestra federate on the first console. To log into the system and edit files, you must switch to the second console by pressing the 'Alt-F2' key combination. The normal user account is "hlauser" with associated password "HLA!now!". The root account is "root" and the password is "abcd1234". Note that these passwords may be changed by the user if required.

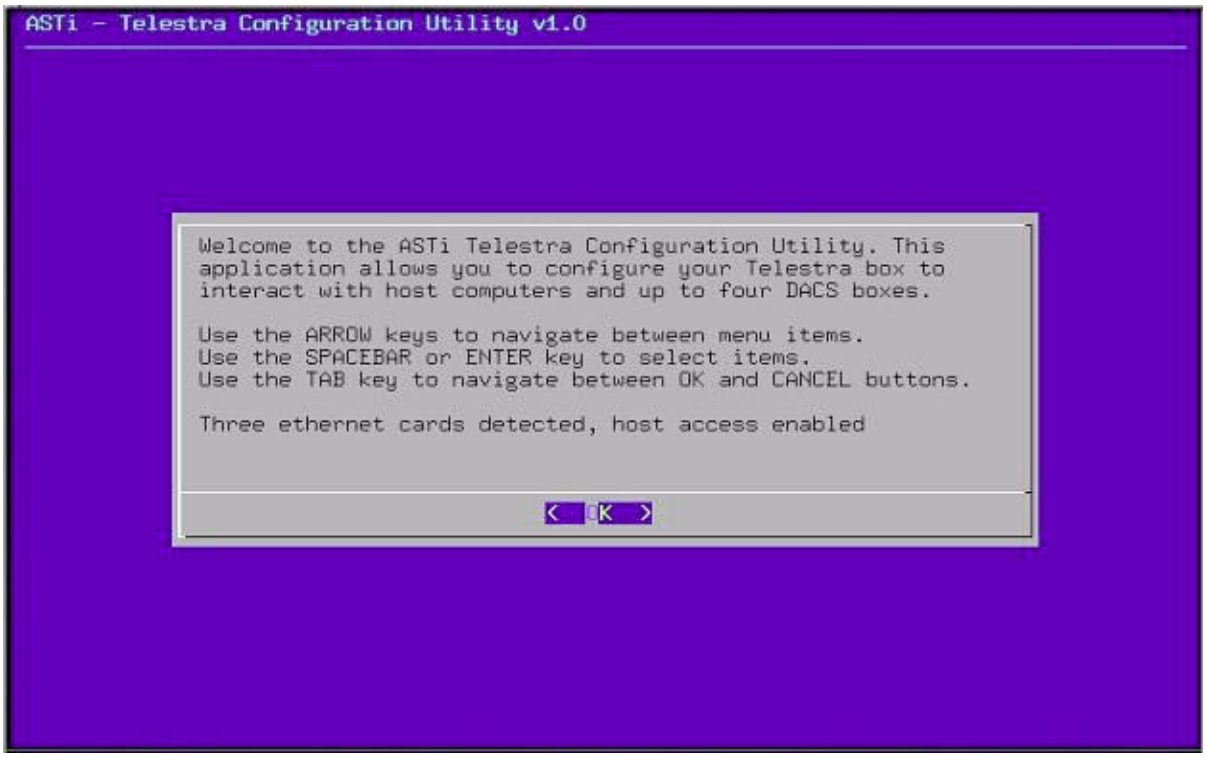
The Linux command "shutdown -h now" may be entered from a console screen to power down the system. Wait until the screen display reads "Power down" before turning the power switch off. It is also possible to shutdown and reboot the system via remote control over the network - see the document Telestra Remote Control Interface User Guide for more details.

Telestra Setup Procedure

Telestra Configuration Utility

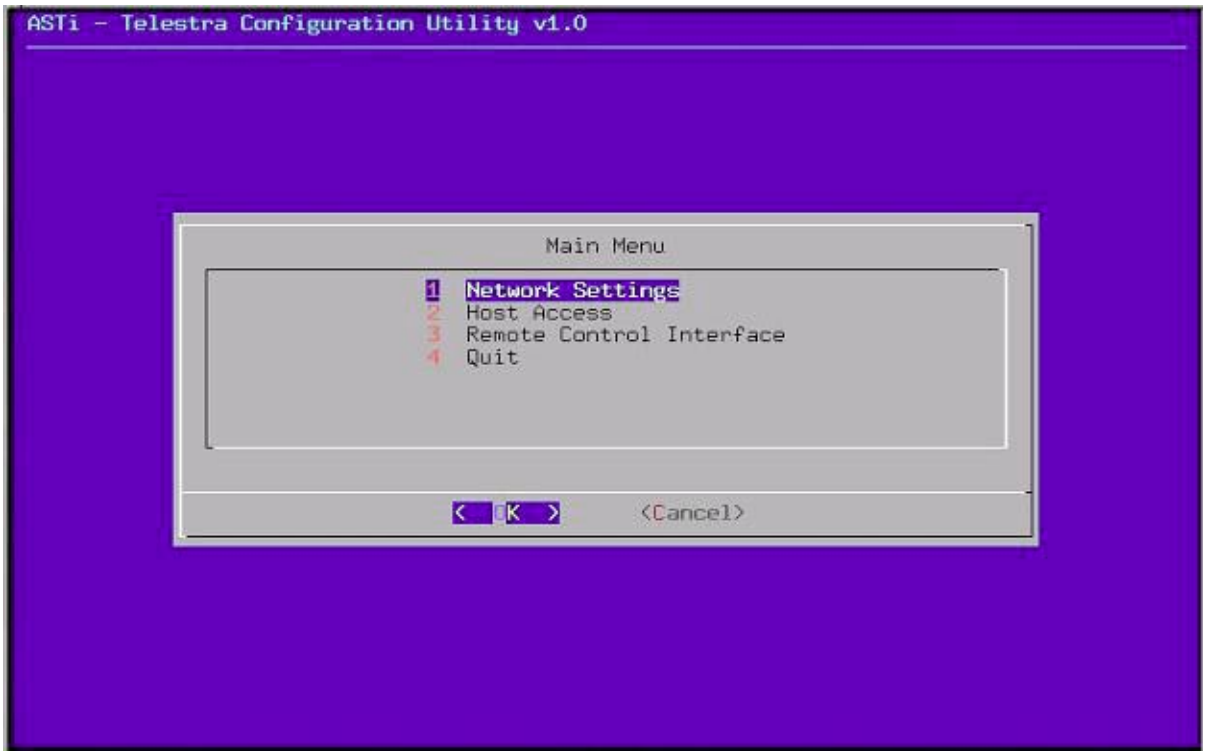
The Telestra system ships with a console GUI configuration tool that supports configuring all network interfaces, host access, terrain queries, tactical data links, and access to the federate remote control interface. To start the configuration utility, login as root and run the command configTelestra.pyc. Figure 3 shows the welcome screen for the Telestra Configuration Utility. The remainder of this section will describe configuring general settings. For instructions on setting up specific features such as tactical data links or terrain queries, please see the corresponding application note at http://www.asti-usa.com/ for more details.

Figure 3: Telestra Configuration Utility Welcome Screen



When the configuration utility first starts, the welcome screens displays general information about using the application. After selecting the OK button, the Main Menu screen shown in Figure 4 appears. This menu has four options, Network Settings, Host Access, Remote Control Interface, and Quit. The Quit option causes the configuration utility to exit and prompts the user about saving configuration changes. Any changes to the system configuration will require a system reboot in order for the changes to take affect. The Network Settings submenu allows configuration of ethernet interfaces as well as general settings such as Hostname, Domain, and Nameserver. The Host Access choice allows users to configure the additional features of the Telestra system (i.e. host interface and tactical data links). Finally, the Remote Control Interface option allows users to enable or disable access to the Remote Control Interface for the Telestra federate software.

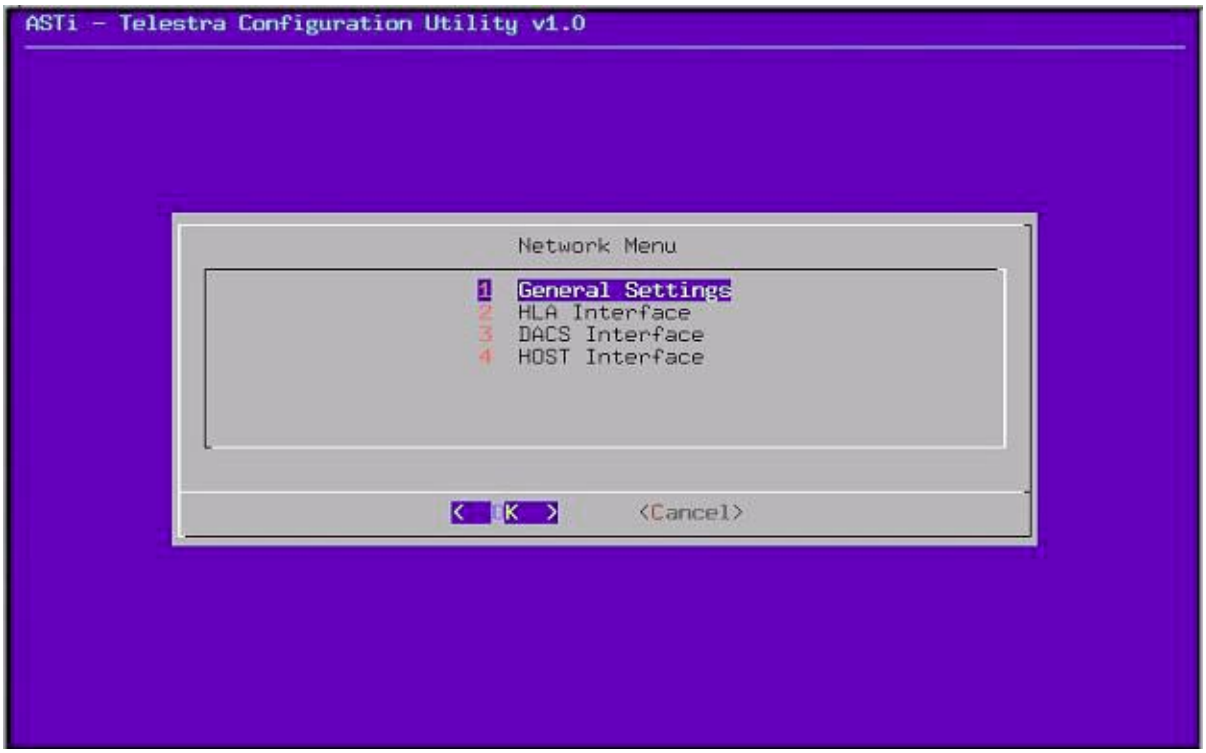
Figure 4: Telestra Configuration Utility Main Menu



Network Configuration

Figure 5 shows the Network Settings Menu, where users configure network information for all three ethernet interfaces. Under the General Settings menu, users specify the Hostname, Domain, and Nameserver for the system. For each of the remaining interfaces, the Telestra Configuration Utility contains options for the IP address, Netmask, and Gateway. The Telestra system ships with default values for all three interfaces. Users will need to modify the settings for the HLA and Host interfaces to match their specific network configurations. Note that the DACS interface does not require configuration unless under direction from ASTi. Please see Appendix A for more information on how to configure the HLA interface in order to join a federation.

Figure 5: Network Settings Menu



Installing a DMSO RTI (Non-U.S. Customers only)

The Telestra system normally comes with some version of the DMSO RTI installed. However, due to legal restrictions, it is difficult to export a system with the DMSO RTI installed. To accommodate our overseas customers, we have installed a script which makes it easy to install a DMSO RTI on the Telestra system. Using this script, a system can be exported and the RTI can be downloaded and installed by the end customer.

To download the DMSO RTI, go to the DMSO web site at http://hla.dmso.mil and register with the software distribution center. You will need the latest RTI compiled for Red Hat Linux 6.1. Please refer to the table below for the RTI version required for your system.

Telestra Version	DMSO RTI Version
Telestra-1.1-1	RTI-NG-1.3v2
Telestra-1.2-1 Telestra-1.4-1	RTI-NG-1.3v3.1

DMSO distributes the RTI in the form of self extracting shell files, which have an extension ".sh".Take this downloaded file and copy it to the Telestra system. The easiest way to do this is via ftp.

Then, log in as root and execute the command 'sh <RTIFilename>' in the directory where the shell file was downloaded. This will unpack the RTI into the following subdirectories: <RTI-Version>/<platform build type>. Upon successful completion, continue to the next section. If you have any problems extracting the RTI, contact DMSO RTI Support via their website at http://helpdesk.dctd.saic.com .

NOTE: ASTi can only guarantee that the system will work with an RTI that has been tested and approved for use with the Telestra federate. Switching to an untested RTI may cause the system to stop functioning.

Run the moveRTI.sh script (Non-U.S. only)

To complete the RTI installation, change directory (cd command) into the <RTI-Version>/<platform build type> subdirectory. Then list the directory contents (ls command) to confirm the following subdirectories exist:

```
apps
bin
doc
etc
config
include
lib
```

Next, execute the 'moveRTI.sh' command to move the RTI components into their ASTi specific locations.

To test if the RTI has been properly installed, log out and log in as hlauser. Then type "telestra -t" at the prompt. If the last line printed reads "TEST OK", then the installation was successful.

How to Run the Software

The software that is run on the Telestra box is an HLA federate. This software, when combined with the DACS audio routing and processing node, implements a full radio simulation environment, based upon the data structures defined in the ASTi Radio SOM. In order to function, the federate must join a federation.

In the normal mode of use, the rtiexec will be running on another computer system resident on the HLA network. However, to allow local only testing or stand-alone tests to be performed, it is possible to run the rtiexec on the Telestra box. The rtiexec must be running somewhere in order for the system to function. The rtiexec can be run by logging in as hlauser and typing rtiexec from consoles 2 through 6 (see below).

From power on, the Telestra system will start the Telestra federate application automatically on console 1. Linux allows access to multiple consoles, which are accessed by pressing “Alt-F1” for console 1, “Alt-F2” for console 2, and so on up to “Alt-F6” for console 6. The Telestra system automatically starts on console 1, consoles 2 through 6 should be used to administer files on the system.

At boot up, the “telestra” script runs the Telestra federate, but in its default configuration does not join a federation. In order to join a federation the “join” command must be used. A list of all possible commands may be seen by typing “help” at the command line.

At this point it is necessary to identify the significance of another file on the system: the configuration file. Configuration files end in the “.cfg” extension and contain basic information, such as the name of the federation to join and the name of .fed file to be used. The format of this file is similar to the configuration file found on the DACS system; however, the commands used are completely different. The Telestra federate will attempt to read in the file default.cfg for configuration settings. If no configuration file named default.cfg exists on the system, then internal default values are used that are coded into the Telestra software.

When the “join” command is used somehow we must tell the software which federation we would like to join. We can use “join” with an argument to specify the federation name. If no argument is supplied with “join” then the name from the configuration file is used. If no name is specified there, the join command will fail. Two federates must use the same federation name to pass information between them.

The “join” command may be used with an argument which overrides the default value. The argument supplied must be the name of the required federation and will be in the form “join <federation_name>”. The system will then issue a series of messages, ending with either “JOIN <federation_name> OK” or “JOIN <federation_name> FAIL”. If the system does not join successfully, the most likely causes are:

- No rtiexec was running
- No federation name was specified
- The .fed file did not contain the names in the ASTi Radio SOM or was not found.

If “JOIN <federation_name> OK” is returned, the system is now running and can be interrogated to determine the federate status and activity. This is achieved by simply pressing the Enter key. A

status report is then generated on-screen that includes a “Life Counter” value. This counter is generated internally and increments at one second intervals (although it only reports when the Enter key is pressed, and a new status report is displayed), and provides a confidence indicator that the system is operational.

Other information reported includes the number of attribute updates for radio receivers (rx) and radio transmitters (tx), the number of interactions (both rx and tx). The values in these fields are useful when verifying the operation of the associated local DACS processor and for remote sites. Attribute updates (such as radio power status, frequency, etc) are sent when there are changes in the transmitter or receiver attributes. Interactions are audio packets. Both types are held as incremental counters, and provide an indication to the system user of the amount of data processed.

Additionally, a current count value is displayed for the “Number of transmitters”, both for the local DACS node(s), and identified remotely on the HLA network from the RTI. A count value is also displayed for “Entities” identified as existing from the RTI. The Telestra federate does not generate any entities itself - it just uses this information to attach radios to entities.

The information displayed as part of the status report may be supplemented using additional command line instructions. The “object” command lists the transmitter, receiver, and entity objects that the federate has discovered. These are displayed for both local DACS objects and objects discovered from the RTI. The name, object type, source of the object (local/RTI) and object handle are displayed.

The “resign” command can be entered at any time in order to leave the current federation. To join a different federation, it is not necessary to resign first: issuing a new join command will cause the Telestra federate to resign from its current federation before attempting to join the new one. If the Telestra federate is instructed to join a federation it is already joined to, it will resign and re-join that federation.

In order to terminate the HLA federate application the “quit” command can by used at any time. If joined, the Telestra federate will resign before exiting.

In summary:

- When the Telestra system powers up, the software attempts to read parameters in the default.cfg file. From this, it gets values for the federation name, the federate name, the fed file, and the conversion file (.conf). These values can be overridden by user commands send directly to the telestra software.
- When the “join” command is issued, the .fed file and the .conf files are read. The telestra federate joins the federation specified (either in the .cfg file or from a command-line input). It reads in the .conf file, and attempts to get handles for all of the attribute names and object names for the RTI.
- If it fails to get any of the handles, it will resign from the federation and return a “JOIN <FEDERATION NAME> FAIL BAD_FED_FILE” message.If not, it will return a “JOIN <Federation Name> OK” message, and begins sending and receiving information to and from the RTI.
- At any time, it can receive a command to join another federation, resign, quit, shutdown, or reboot.

Using the Remote Control Interface

The commands discussed above are part of a complete set of commands that the Telestra federate recognizes. These commands can either be issued from the console or through a TCP/IP connection with the Telestra Remote Control Interface. We suggest that you use the console interface when first setting up and integrating the system. The next step is to set up the remote control interface if you desire to integrate the Telestra unit into a broader simulator control architecture. Once the remote control interface is in place, you can completely control the system from a remote computer. You can even shutdown the system through the Remote Control Interface, eliminating the need for a keyboard and monitor on the Telestra System.

For more information on the Telestra commands and the remote control interface, see the document Telestra Remote Control Interface User Guide.

The Configuration File

This file contains a list of parameters used to configure the Telestra federate when it is started. Unless specified otherwise by a command line argument, the file “default.cfg” is used. The values defined in the file are read when the application federate is started. Note that any text following a semi-colon (;) is treated as a comment and ignored. This file may be edited using the ‘pico’ editor (or perhaps those with a more self-flagilating persuasion may prefer the Unix standard ‘vi’ editor). The following commands are available:

`Federate_name = <Federate name>`

This command allows the user to specify the federate name. Note this name must be unique for a given federation. Each system will automatically select a unique name based upon the local IP address unless an entry is made in this file. Typically, this does not need to be set.

`Federation_name = <Federation name>`

This defines the federation that is joined, either automatically upon start-up (see Join_at_startup), or when a “join” command is used without an argument.

`Fed_file = <Fed file name>`

This defines the .fedfile that the system will use. On shipment the fed file will be “ASTi_SOM.fed”, but will require updating once the customer fed file is created that includes the ASTi Radio SOM.

`Convert_table = <Data representation definitions file>`

This file defines the data representations used by the Telestra Federate. The default file utilized will be “convert.conf” unless an entry is made in this file. It may be necessary to modify the data representations file, dependent on the specific FOM being used. To use a non-standard data format please contact ASTi.

`Join_at_startup = <Y or N>`

This entry determines whether the federate will attempt to join the federation upon start-up. The default value is N and therefore requires the user to perform a manual “join”. If this is set to Y, and the Telestra federate is run before an rtiexec is started, the federate will be unable to join, and may crash on some RTIs.

`Control_Port = <port number>`

Set the TCP/IP port number for the Remote Control Interface for the telestra software. When given, commands are not taken from the console. Instead, a TCP/IP socket is opened and waits for connections on the specified port number. Control of the federate is done through this port. For more information, see Using The Telestra Remote Control Interface. The default value is to use the console interface.

`Debug_Host = <ON or OFF>`

This is used to debug the TCP/IP connection that is set up with the Control_Port command. For more information, see the document Using the Telestra Remote Control Interface. The default value is OFF.


```
Audio_Backchannel = <ON or OFF>
```

If this is set, the Telestra Federate sends the audio over the HLA physical network, but bypasses the RTI. The Telestra federate will still create transmitter and receiver HLA objects. This is intended for cases when the RTI is unable to keep up with the amount of audio traffic being sent. In this mode the audio will not be seen on any of the standard HLA tools. The default value is OFF.

System File Structure

When you log onto the Telestra system as hlauser, your are positioned into the “hlauser” directory (/home/hlauser/). This directory contains all the files that are needed to configure and run the Telestra Federate. The following files are resident in this directory (or sub directory as indicated).

`telestra`

This file starts the Telestra federate. Use of the argument “-t” forces the system to run a system confidence test. When simply used on the command line, this starts the application federate, and uses the commands found in the configuration file.

`example.cfg`

This file is an example configuration file. It includes start-up configuration values for system file names and values.

`convert.conf`

This file contains the ASTi data format representation definitions. Modification to some of the names used within this file may be required, depending on the FOM being used for the system. This should be only done under direction from ASTi.

`ASTi_SOM.fed`

This file is a .fed file that only contains the ASTi radio SOM. In normal system operation a customer generated .fed file will exist in this directory, that will include the ASTi radio SOM. It will be necessary for the configuration file to point to the correct .fed file.

`RTI.rid`

This file is part of the RTI configuration, and has been modified from the standard file sent with the DMSO RTI to optimize the performance of the Telestra Federate. It should only be modified under the direction of ASTi.

`handle_1.val`

This is a debug file generated upon execution of the “join” command. It includes the name of the federation joined, time of joining, and a list of all the attributes and interactions published/subscribed to and their handles.If the RTI cannot get handles for any of the attributes, classes, or parameters that the telestra federate requests, it will resign from the federation, return a “JOIN FAIL”, and print the offending name in this file. As long as the system is functioning, this file can be safely ignored.

`~/docs/ASTi_SOM.omt`
`~/docs/ASTi_SOM.omd`

These files specify the ASTi SOM , and should be used to incorporate the ASTi SOM into the customer’s FOM. This can be done with the OMDT tool, which is available from the DMSO web site, or from another commercially available tool. The resultant .fed file will need to be copied to the /home/hlauserdirectory and the configuration file updated.

`~/docs`

Includes copies of user documentation. All documentation is in html format, and can be viewed with any web browser.

`~/bin`

Contains scripts helpful for using and maintaining the system. Currently, there is only one for customer use: hostemu. hostemu is used to debug the remote Control Interface, and is described more fully in the document Telestra Remote Control Interface User Guide.

The RTI Software

As stated previously the rtiexec application must be running somewhere on the network to which Telestra is connected. However there may be cases where the system is down or some standalone testing is required. In order to support this Telestra includes a local copy of rtiexec.

In order to run the RTI, a second console window should be opened. Log in as hlauser and enter “rtiexec” to run the RTI. This rtiexec will service every system on the network.

To stop the RTI, type “Ctrl-C”.

DACS Operational Differences running HLA

Note: This guide is written with the assumption that the user is familiar with DACS operation, and DIS Radios in the DACS.

The changes to the DACS Model Builder application are relatively straight forward in order to allow it to function in HLA mode. Note that all previous functionality in non networked and DIS operational modes are retained and are available in HLA mode.

For HLA operation the DACS system must be configured to run in HLA mode. This is done via a command in the DACS configuration (".cfg") file. The required command line is:

```
hla = on
```

Note: It is important to note that the "default.cfg" on the DACS has a different functionality to the "default.cfg" file used on the Telestra node.

When Model Builder is started with the "hla = on" command you will notice that the main menu now includes an selection titled "hlA network".

IMPORTANT: Any configuration file commands that begin with "DIS" should be commented out or removed.

Changes at the object level in Model Builder are restricted to the "Radio", "Receiver" and "Transmitter" objects in the "Signals" list and to "World Position" objects in the "Controls" list.

The changes to the "Radio", "Receiver" and "Transmitter" objects are all common to each other. The most obvious and fundamental change is that these objects are no longer identified using the "Site/Host/Entity/Radio ID" numbers of DIS. The identification is now via a concatenation of the World Position name used by the object to define it's location, and the Radio object name. These names are the user created names entered as part of the Model development (note that if the user does not enter a name, the system will chose a default name for each object). Therefore assuming we have a world position control object identified "Aircraft_Posn" and a radio object identified "Cockpit_UHF1", then the object name displayed (and created on the RTI) will be "Aircraft_Posn.Cockpit_UHF1.rx". (A radio object actually consists of two separate objects which are created on the RTI. These are Transmitter objects, whose names the ASTi System ends in '.tx', and receiver objects, whose names will end in '.rx').

Note that it is no longer necessary to manually set the Radio ID number to anything other than the Model Builder supplied default, since this is not published or utilized outside of the DACS/ Telestra pair.

World Position objects have been changed to include the option of setting the Entity ID type to "HLA". This is achieved using the "Entity ID" field and incrementing through the options. On selecting the "HLA" type, the line above changes from "Exercise No" to become "Federation No". Federation numbers 1-8 are reserved for RTI communication. The "Federation No" must be set to "1" in the current implementation since Telestra only supports one Federation. Please see the application note Using Telestra Backchannels for more information on assigning Federation numbers. The other numbers displayed for the site, host and entity fields should be left at the default values, since Model Builder defaults these to be unique, and they are not reported as part of the HLA data.

The “Entity” object used for entity attach now functions in a different way. It is now only necessary to enter the HLA name of the required attach entity in the name field of the Entity object in Model Builder. The system then interrogates the RTI and extracts the required position information. It is no longer required to set-up site/host/entity numbers. The “Federation No” number must be set to “1”. This will be changed in future versions to allow multiple federation support. A method of dynamically attaching radios to an entity across the network will be added in an upcoming release.

The network intercom object is currently represented as a Radio object on the HLA network.

System Utilities

Accessing a Floppy Disk

One issue when using the Linux operating system is the difficulty in accessing peripherals. Traditionally peripherals must be mounted and unmounted in the correct fashion. Red Hat Linux (and hence the Telestra System) includes a set of utilities that greatly simplifies access to the 3.5" floppy drive fitted to the unit. These tools are known as "mtools" and allow DOS-like commands to be used for all normal floppy operation, without the need to mount/unmount the drive.

Useful commands are:

`mdir a:`
Provides a directory listing of the floppy disk contents.

`mcopy <filename> a:`
Copies a named file to the floppy.

`mcopy a:<filename> .`
Copies a named file from the floppy into the current directory

`mdel a:/<filename>`
Deletes a specified file from the floppy disk. Note the use of the "/" slash in place of the DOS standard "\".

For a full synopsis of the utility, enter "man mtools" from the console.

Editing Text Files

There are several text editors available on Red Hat Linux. A very quick and simple text editor to use is pico. To run it simply enter

```
pico <filename>
```

from the command line. For the Unix pro, vi is available, as well as emacs and jed. If you are in X-windows, kedit provides an intuitive, simple to use interface.

For more information about these tools, or any other Linux command, type man <command name> for a full description.

Updating the Telestra software

The Telestra software updates are distributed as a Red Hat Package file, which have a file extension ".rpm". These are small enough to fit on a floppy disk, so the mtools mentioned above can be used to get them from a Windows machine onto the Telestra system.

To update the telestra software, copy the new software rpm file onto a DOS or Windows floppy. Log in to the Telestra system as root and type:

```
mount -t msdos /dev/fd0 /mnt/floppy
cd /mnt/floppy
./upgrade
```

or to use a linux based floppy type:

```
mount /mnt/floppy
cd /mnt/floppy
./upgrade
```

(One rpm file name will be telestra-x.x-x.i386.rpm, where x.x-x represents the release version.) This will install the new Telestra software.

Note that this will only let you move up in the revision levels. To install an earlier version or reinstall the current version, type instead:

```
rpm -e telestra
```

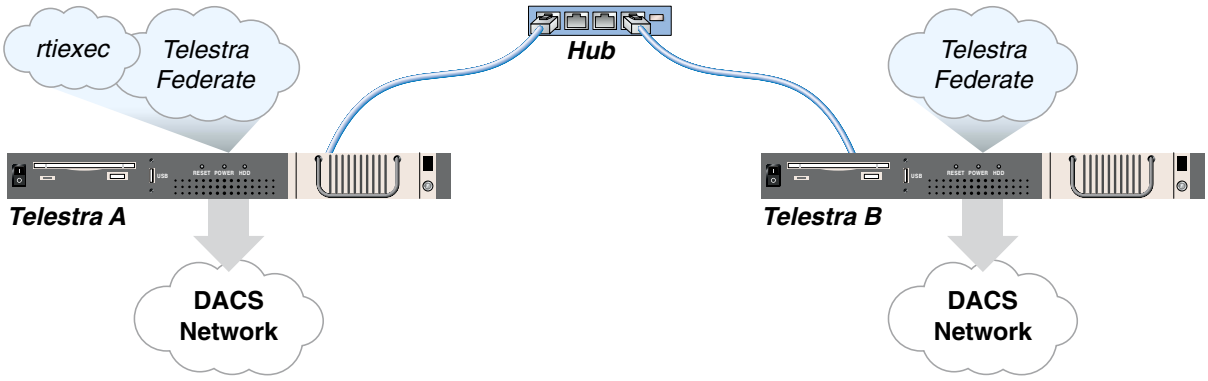
Then follow the instructions above to install the older version of Telestra.

Once the new software is installed, you can test it by logging out, logging in again as hlauser, and typing "telestra -t".

Appendix A: Joining a Federation

This section assumes that the RTI and Telestra federate software are properly installed on both Telestra systems. To test the installation, log in as hlauser and type “telestra -t”. For the purposes of this scenario, the rtiexec process will run on one of the two Telestra systems. Only one rtiexec process should exist on the network. For performance reasons, ASTi recommends running this process on a separate machine during an actual exercise and NOT on one of the Telestra systems. Figure 1 shows a picture of the network topology for this example. A hub connects the two Telestra systems together to form an isolated network. For the purposes of this example, we will ignore the host interface and DACS interface on both Telestra systems.

Figure 1: Network Topology for Example Scenario



The table below displays the network settings for both Telestra systems. Following this table are step-by-step instructions on using the Telestra Configuration Utility to setup both Telestra systems.

NOTE: Users will need to change the network settings below to match their own network configuration.

Hostname	Telestra_A	Telestra_B
Domain	asti-usa.com	asti-usa.com
IP Address	192.168.1.100	192.168.1.200
Netmask	255.255.255.0	255.255.255.0
Gateway	192.168.1.254	192.168.1.254

Telestra System Set-Up

These instructions walk through using the configuration utility to setup the Telestra_A system. To setup another system, follow these instructions on that system, using the network settings for that system.

Step 1: Make sure the Telestra system is powered ON.

Step 2: Switch to the second console using 'ALT-F2' and login as root.

Step 3: Start the Telestra Configuration Utility using 'configTelestra.pyc'. The network settings will reflect the default values.

Figure 2: Starting the Configuration Utility

Step 4: Under the Network Settings -> General Settings menu, change the default hostname.

Figure 3: Default General Network Settings

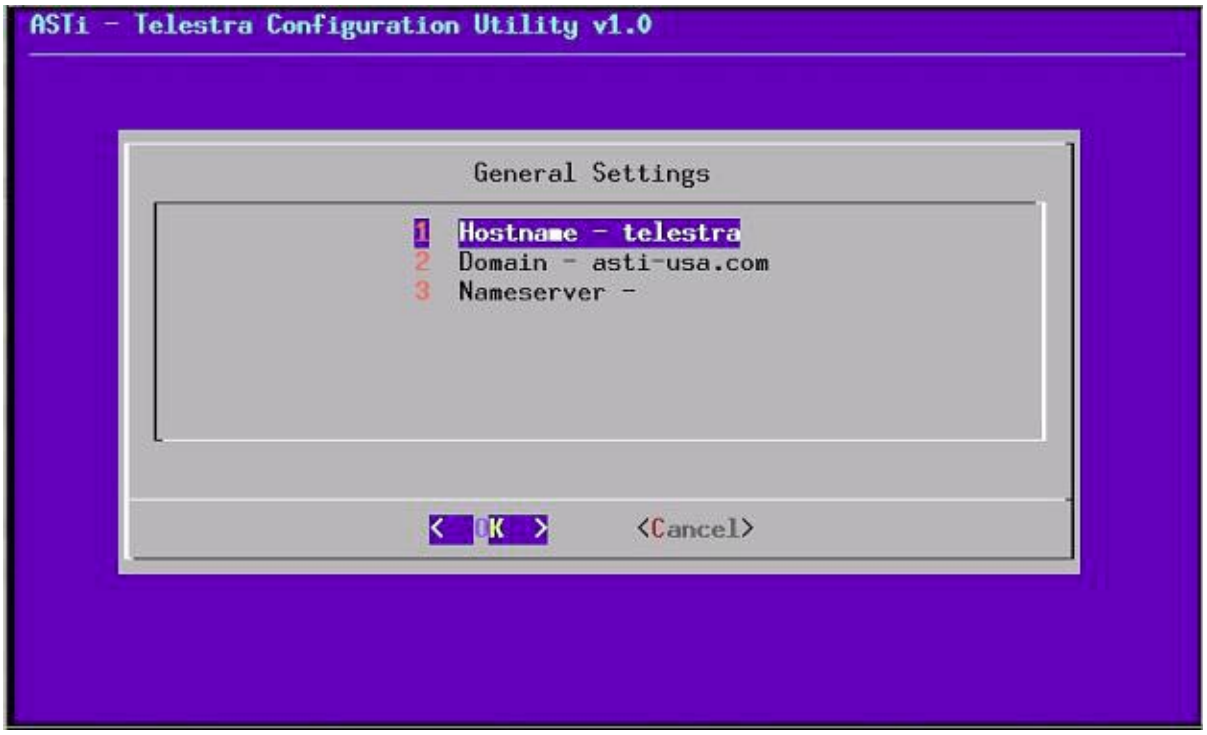


Figure 4: Changing the Default Hostname

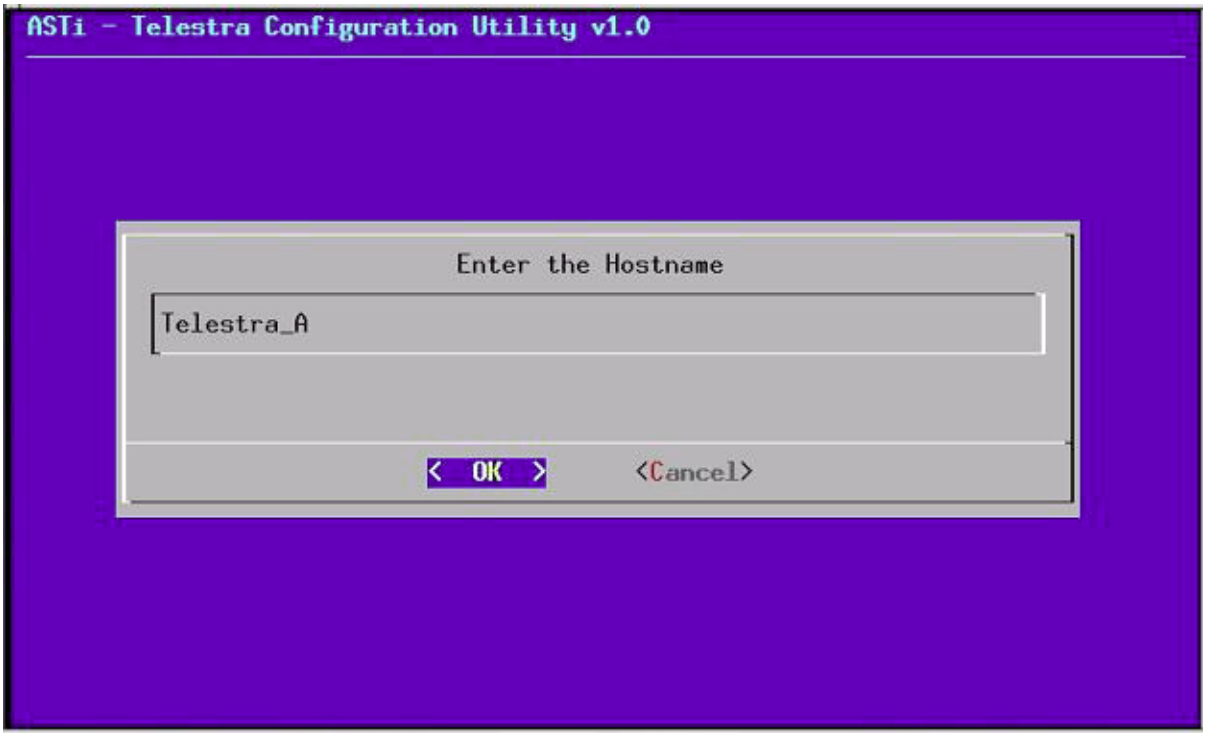
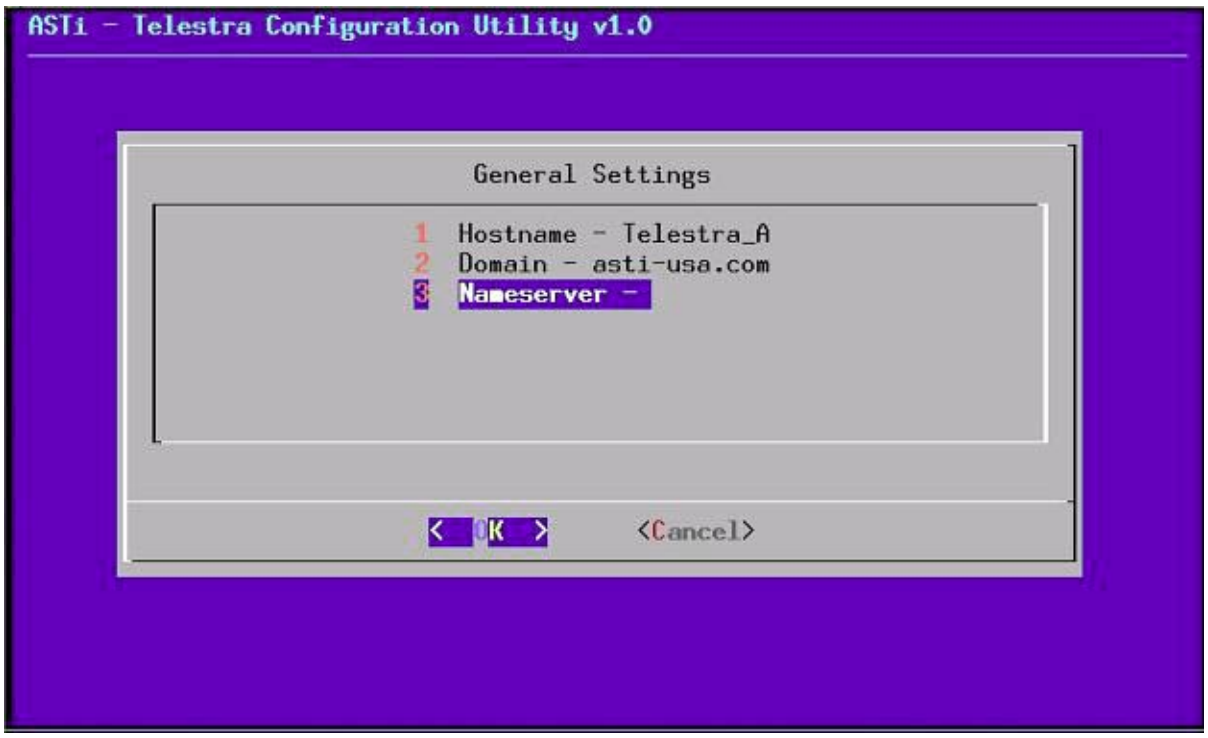


Figure 5: Modified General Network Settings



Step 5: Under Network Settings -> HLA Interface, change the IP address.

Figure 6: Default HLA Interface Settings

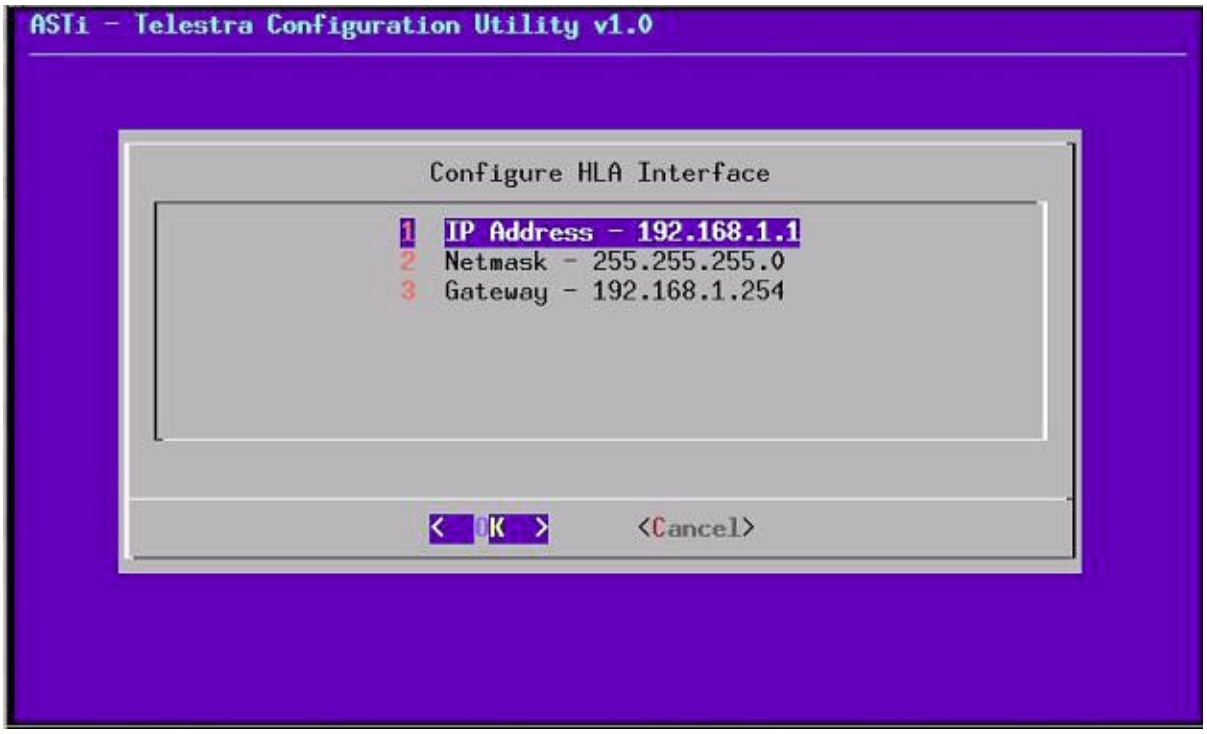


Figure 7: Changing the Default IP Address

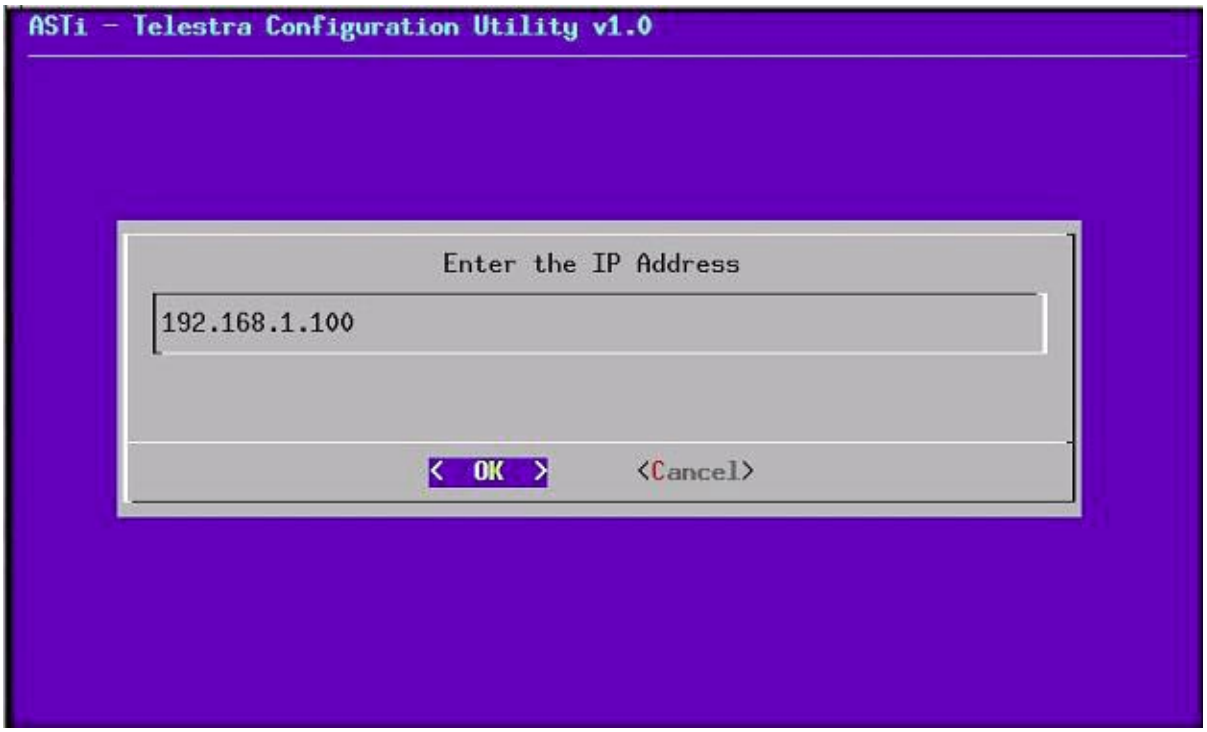
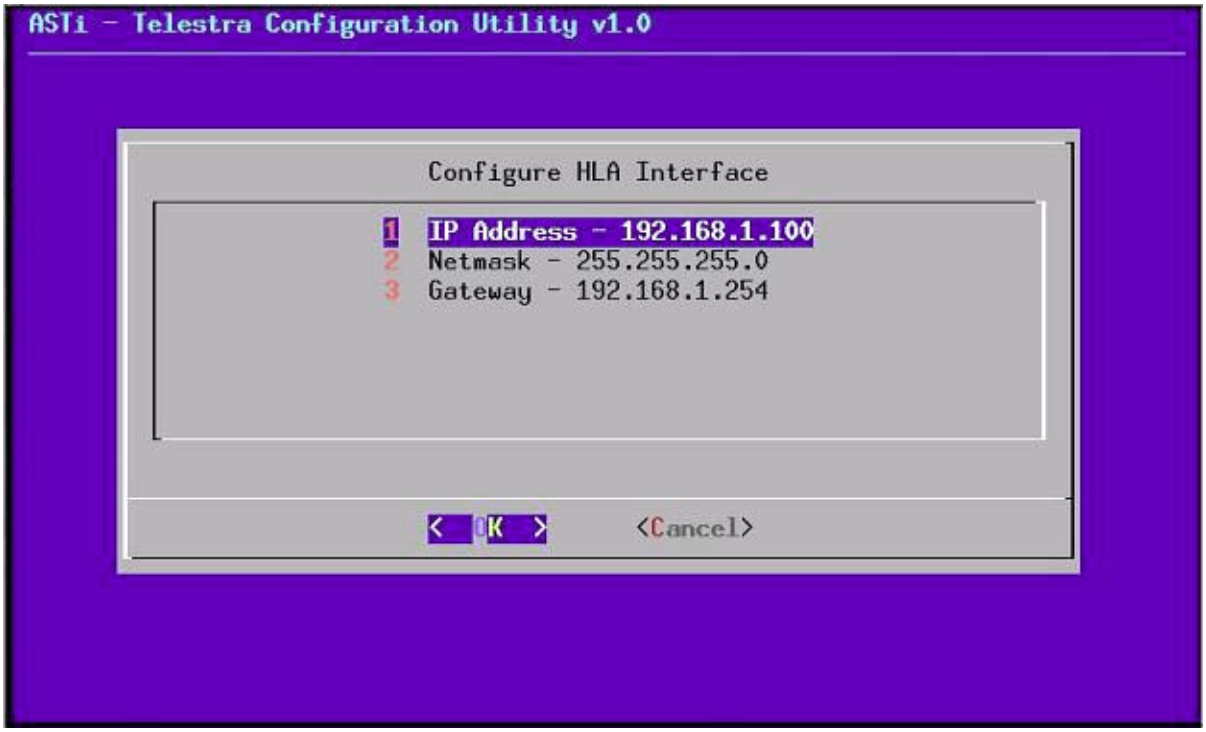


Figure 8: Modified HLA Interface Settings



Step 6: Reboot the system to activate the changes.

Step 7: Repeat steps 1 through 6 for additional machines, using the specific settings for each machine.

Setting Up the Federation

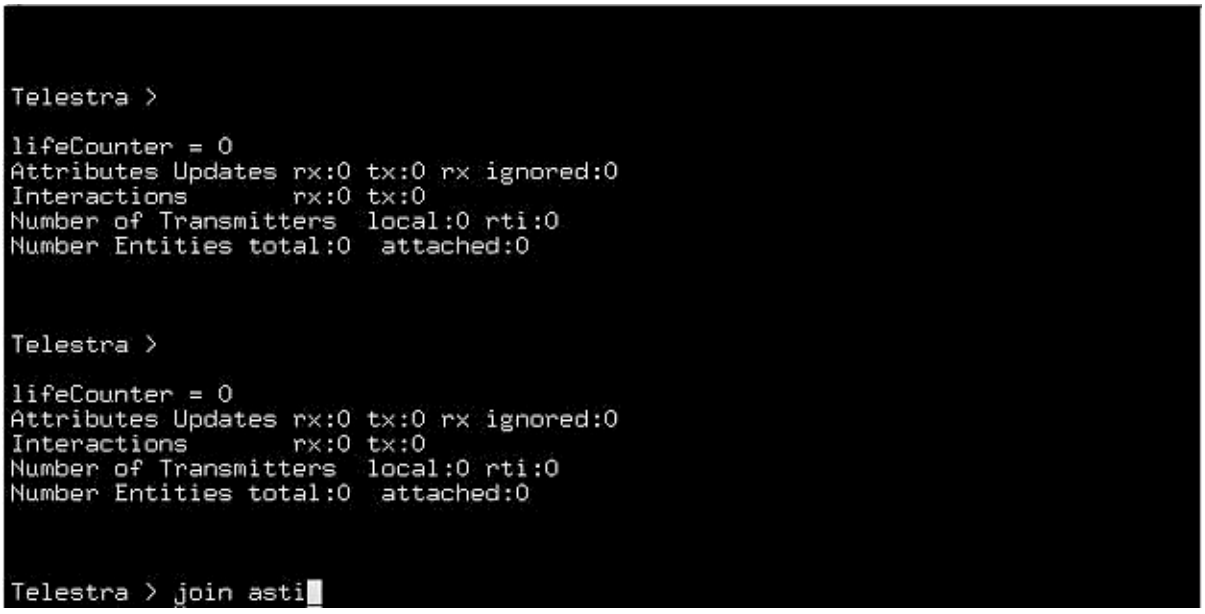
After configuring and rebooting all machines participating in the exercise (two in this example), you are ready to start setting up the Federation.

Step 1: On Telestra_A, switch the the second console and login as hlauser. Then start the rtiexec process by running the command ‘rtiexec’.

Figure 9: Running rtiexec on Telestra_A

Step 2: On each machine switch to the first console (You will see the ‘Telestra >’ prompt) and enter the command ‘join asti’.

Figure 10: Telestra Federate before the Join Command



Step 3: Wait for the federate software to return a 'JOIN OK' to indicate a successful join operation. If the software returns a 'JOIN FAIL', double check the network configuration on each machine.

Figure 11: Telestra Federate after the Join Command

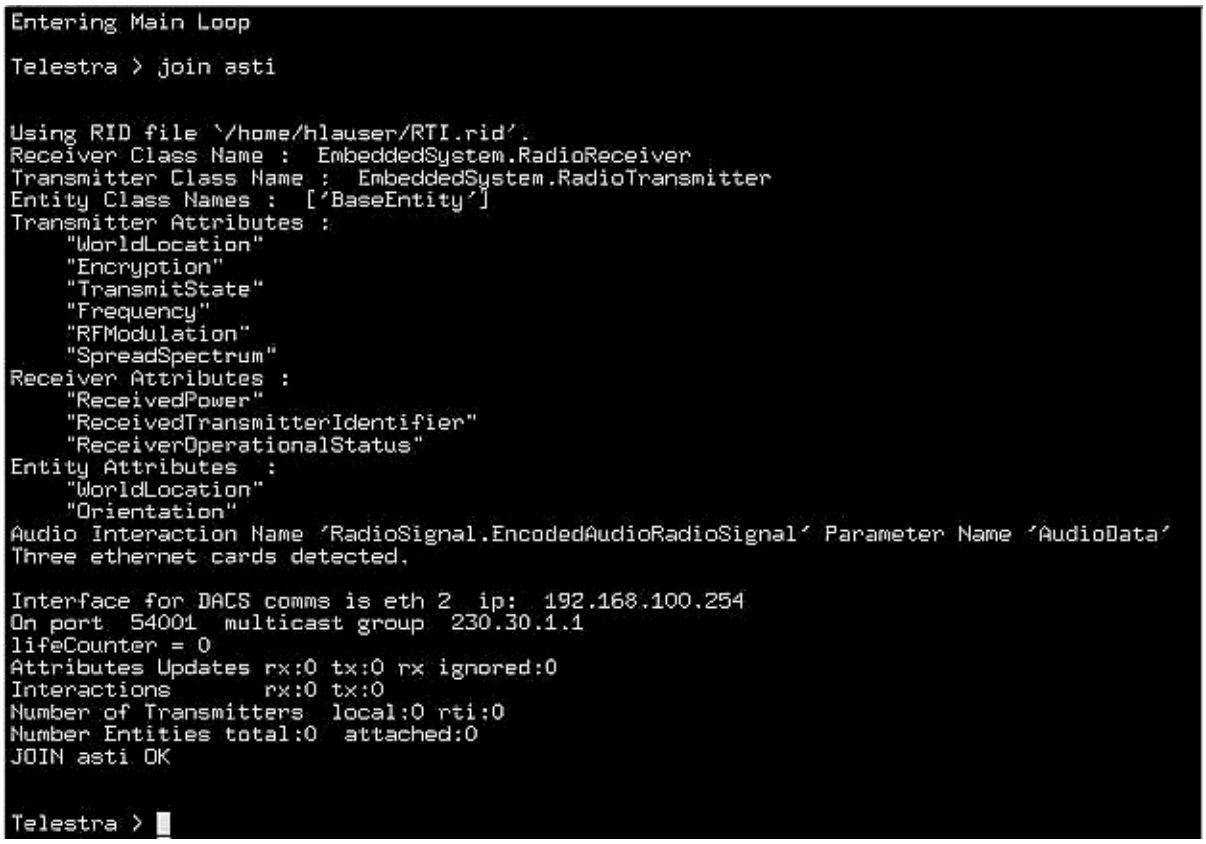
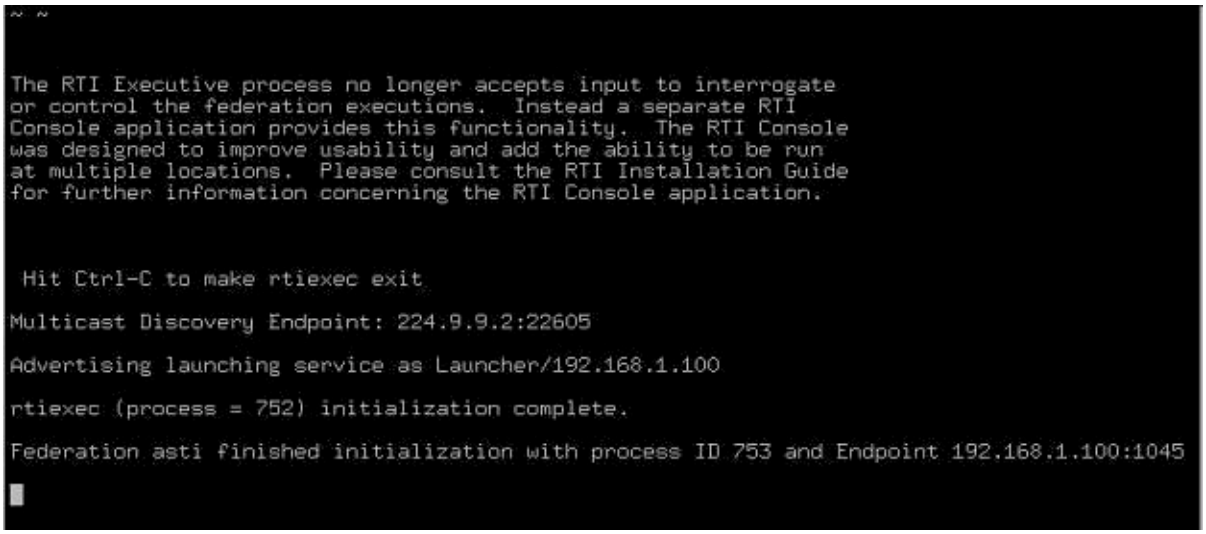
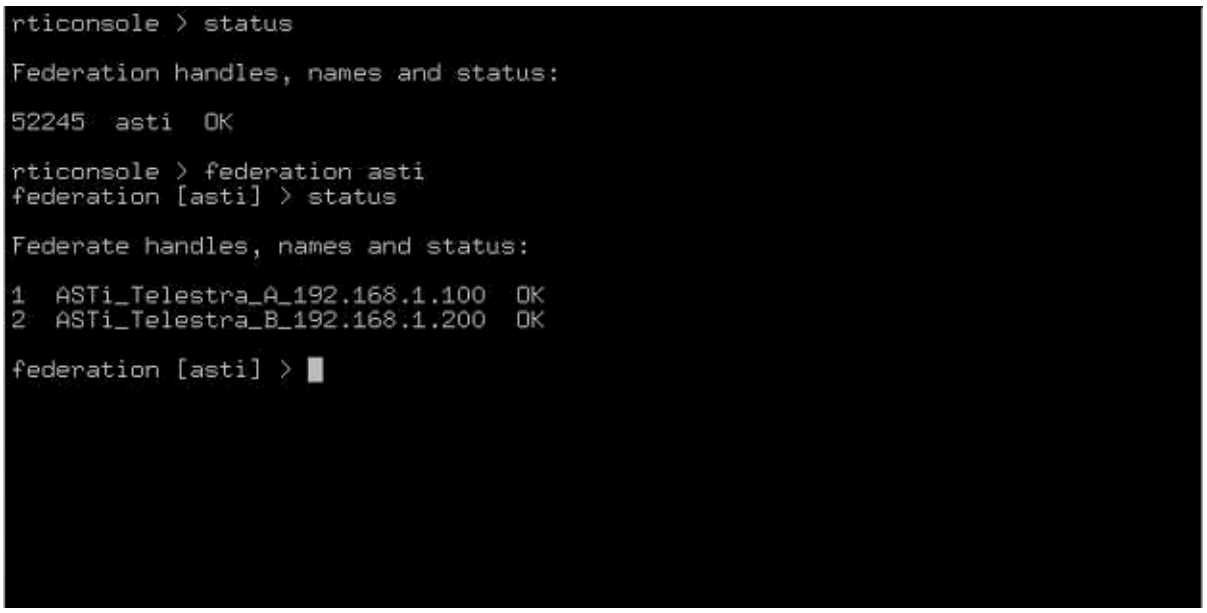


Figure 12: The rtiexec Indicating Successful Federate Initialization



Step 4: On any Telestra system, switch to one of the alternate consoles and execute the command 'rtiConsole'. This will allow the user to view the current Federations, as well as the list of joined Federates. Please see the RTI documentation for more information about using the RTI Console application. The figure below shows some example commands to query the status of the Federation and Federates.

Figure 13: Using the rtiConsole to Check the Federation and Federates



At this point, there is an 'asti' Federation with two Federates joined. Note that the Federate name for each Telestra system was generated based on the Hostname and IP address of the individual machine. In order to properly destroy the federation, you must:

1. Exit the RTI Console using the 'quit' command.
2. Resign each Federate by typing 'resign' at the 'Telestra >' prompt on both Telestra systems.
3. Stop the rtiexec process using CTRL-C.

Appendix B: ASTi Radio SOM

Version 3.0 OMD File

```
(OMDT v1.3.4.16)
(ObjectModel (Name "ASTi Radio SOM")
 (VersionNumber "3.0")
 (Type FOM)
 (Purpose "To support real time,person in the loop radio simulation")
 (ApplicationDomain "Real time, platform level simulations.\r\n")
 (SponsorOrgName "ASTi")
 (POCHonorificName "Mr")
 (POCFirstName "David")
 (POCLastName "Nemeth")
 (POCOrgName "ASTi")
 (POCPhone "(703) 471-2104")
 (POCEmail "davidn@asti-usa.com")
 (ModificationDate 05/19/2000)
 (MOMVersion "1.3")
 (FEDname "FederationName")
 (ComplexDataType (Name "AntennaPatternStruct" [2, 3])
 (ComplexComponent (FieldName "AntennaPatternType")
(DataType "AntennaPatternTypeEnum32")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "The type of radiation pattern of the radio\'s antenna.")
 )
 (ComplexComponent (FieldName "BeamAntenna")
(DataType "BeamAntennaStruct")
 (Cardinality "0-1")
 (Units "meters")
 (Resolution "1.0")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SphericalHarmonicAntenna")
(DataType "SphericalHarmonicAntennaStruct")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "BeamAntennaStruct")
 (ComplexComponent (FieldName "BeamOrientation")
(DataType "OrientationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamAzimuthBeamwidth")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamElevationBeamwidth")
```



```
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "ReferenceSystem")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding1" [9])
(DataType "octet")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding2" [9])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Ez")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Ex")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamPhaseAngle")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "OrientationStruct")
 (ComplexComponent (FieldName "Psi")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Theta")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
```



```
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Phi")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "RFModulationStruct" [10])
 (Description "Structure containing the type of RF Modulation")
 (ComplexComponent (FieldName "MajorModulationType")
(DataType "MajorRFModulationTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AmplitudeModulationType")
(DataType "AmplitudeModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AmplitudeAngleModulationType")
(DataType "AmplitudeAngleModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AngleModulationType")
(DataType "AngleModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "CombinationModulationType")
(DataType "CombinationModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "PulseModulationType")
(DataType "PulseModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "UnmodulatedType")
(DataType "UnmodulatedTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "RFModulationSystemType")
(DataType "RFModulationSystemTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
```



```
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "RTIObjectIdStruct" [1])
 (ComplexComponent (FieldName "ID")
(DataType "string")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "An RTI object name string")
 )
 )
 (ComplexDataType (Name "SINCGARSModulationStruct" [14])
 (Description "Detailed information about the CCTT SINCGARS")
 (ComplexComponent (FieldName "FHNetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "HopSetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "LockoutSetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "FHSynchronizationTimeOffset" [11])
(DataType "long")
 (Cardinality "1")
 (Units "seconds")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "TransmissionSecurityKey" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "SphericalHarmonicAntennaStruct" [7])
 (ComplexComponent (FieldName "Order")
```



```
(DataType "char")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Coefficients")
(DataType "float")
 (Cardinality "1+")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "ReferenceSystem")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "WorldLocationStruct")
 (ComplexComponent (FieldName "X")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Y")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Z")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (EnumeratedDataType (Name "AmplitudeAngleModulationTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence No)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AmplitudeAndAngle")
(Representation 1))
 )
 (EnumeratedDataType (Name "AmplitudeModulationTypeEnum16" [5])
 (Description "Amplitude enumerations")
 (AutoSequence No)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AudioFrequencyShiftKeying")
```



```
(Representation 1))
 (Enumeration (Enumerator "AmplitudeModulation")
(Representation 2))
 (Enumeration (Enumerator "ContinuousWaveModulation")
(Representation 3))
 (Enumeration (Enumerator "DoubleSideband")
(Representation 4))
 (Enumeration (Enumerator "IndependentSideband")
(Representation 5))
 (Enumeration (Enumerator "SSB_LowerSideband")
(Representation 6))
 (Enumeration (Enumerator "SSB_FullCarrier")
(Representation 7))
 (Enumeration (Enumerator "SSB_ReducedCarrier")
(Representation 8))
 (Enumeration (Enumerator "SSB_UpperSideband")
(Representation 9))
 (Enumeration (Enumerator "VestigialSideband")
(Representation 10))
 )
 (EnumeratedDataType (Name "AngleModulationTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence No)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "FrequencyModulation")
(Representation 1))
 (Enumeration (Enumerator "FrequencyShiftKeying")
(Representation 2))
 (Enumeration (Enumerator "PhaseModulation")
(Representation 3))
 )
 (EnumeratedDataType (Name "AntennaPatternTypeEnum32" [6, 13])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "OmniDirectional")
(Representation 0))
 (Enumeration (Enumerator "Beam")
(Representation 1))
 (Enumeration (Enumerator "SphericalHarmonic")
(Representation 2))
 )
 (EnumeratedDataType (Name "CombinationModulationTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AmplitudeAnglePulse")
(Representation 1))
 )
 (EnumeratedDataType (Name "CryptographicSystemTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
```



```
 (Enumeration (Enumerator "KY_28")
(Representation 1))
 (Enumeration (Enumerator "KY_58")
(Representation 2))
 (Enumeration (Enumerator "NarrowSpectrumSecureVoice_NSVE")
(Representation 3))
 (Enumeration (Enumerator "WideSpectrumSecureVoice_WSVE")
(Representation 4))
 (Enumeration (Enumerator "SINCGARS_ICOM")
(Representation 5))
 )
 (EnumeratedDataType (Name "EncodingTypeEnum16" [5])
 (AutoSequence Yes)
 (StartValue 1)
 (Enumeration (Enumerator "Encoding_8-bit_mu-law")
(Representation 1))
 (Enumeration (Enumerator "CVSD_per_MIL-STD-188-113")
(Representation 2))
 (Enumeration (Enumerator "ADPCM_per_CCITT_G721")
(Representation 3))
 (Enumeration (Enumerator "Encoding_16-bit_linear_PCM")
(Representation 4))
 (Enumeration (Enumerator "Encoding_8-bit_linear_PCM")
(Representation 5))
 (Enumeration (Enumerator "VQ_<Vector_Quantization>")
(Representation 6))
 (Enumeration (Enumerator "CVSD_per_CECOM")
(Representation 7))
 )
 (EnumeratedDataType (Name "MajorRFModulationTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Amplitude")
(Representation 1))
 (Enumeration (Enumerator "AmplitudeAndAngle")
(Representation 2))
 (Enumeration (Enumerator "Angle")
(Representation 3))
 (Enumeration (Enumerator "Combination")
(Representation 4))
 (Enumeration (Enumerator "Pulse")
(Representation 5))
 (Enumeration (Enumerator "Unmodulated")
(Representation 6))
 )
 (EnumeratedDataType (Name "PulseModulationTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Pulse")
(Representation 1))
 )
 (EnumeratedDataType (Name "RadioInputSourceEnum8" [4])
 (Description "Duh!")
```



```
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Pilot")
(Representation 1))
 (Enumeration (Enumerator "Copilot")
(Representation 2))
 (Enumeration (Enumerator "FirstOfficer")
(Representation 3))
 (Enumeration (Enumerator "Driver")
(Representation 4))
 (Enumeration (Enumerator "Loader")
(Representation 5))
 (Enumeration (Enumerator "Gunner")
(Representation 6))
 (Enumeration (Enumerator "Commander")
(Representation 7))
 (Enumeration (Enumerator "DigitalDataDevice")
(Representation 8))
 (Enumeration (Enumerator "Intercom")
(Representation 9))
 )
 (EnumeratedDataType (Name "ReceiverOperationalStatusEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Off")
(Representation 0))
 (Enumeration (Enumerator "OnButNotReceiving")
(Representation 1))
 (Enumeration (Enumerator "OnAndReceiving")
(Representation 2))
 )
 (EnumeratedDataType (Name "RFModulationSystemTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Generic")
(Representation 1))
 (Enumeration (Enumerator "HQ")
(Representation 2))
 (Enumeration (Enumerator "HQII")
(Representation 3))
 (Enumeration (Enumerator "HQIIA")
(Representation 4))
 (Enumeration (Enumerator "SINCGARS")
(Representation 5))
 (Enumeration (Enumerator "CCTT_SINCGARS")
(Representation 6))
 )
 (EnumeratedDataType (Name "TransmitterOperationalStatusEnum8" [4])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Off")
(Representation 0))
```



```
 (Enumeration (Enumerator "OnButNotTransmitting")
(Representation 1))
 (Enumeration (Enumerator "OnAndTransmitting")
(Representation 2))
 )
 (EnumeratedDataType (Name "UnmodulatedTypeEnum16" [5])
 (Description "Duh!")
 (AutoSequence Yes)
 (StartValue 0)
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "ContinuousWaveEmission")
(Representation 1))
 )
 (ComplexDataType (Name "CryptographicStruct" [12])
 (ComplexComponent (FieldName "CryptographicSystem")
(DataType "CryptographicSystemTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "EncryptionKey")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "FrequencyStruct")
 (Description "Radio frequency and bandwidth")
 (ComplexComponent (FieldName "Frequency")
(DataType "unsigned long long")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "Frequency of transmission ")
 )
 (ComplexComponent (FieldName "FrequencyBandwidth")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "Bandwidth of transmission ")
 )
 )
 (EnumeratedDataType (Name "SpreadSpectrumEnum16")
 (Description "Describes the spread spectrum (or frequency or time hopping) mode")
 (AutoSequence No)
 (StartValue 1)
 (Enumeration (Enumerator "None")
(Representation 0))
 (Enumeration (Enumerator "SINCGARSFrequencyHop")
(Representation 5))
 )
 (ComplexDataType (Name "TransmitStateStruct")
 (Description "Transmission state parameters")
 (ComplexComponent (FieldName "TransmitterOperationalStatus")
(DataType "TransmitterOperationalStatusEnum8")
 (Cardinality "1")
```



```
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "RadioInputSource")
(DataType "RadioInputSourceEnum8")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding" [9])
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "TransmittedPower")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "StreamTag")
(DataType "unsigned long long")
 (Cardinality "1")
 (Units "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "The tag of the audio stream being transmitted.")
 )
 )
 (ComplexDataType (Name "SpreadSpectrumStruct")
 (Description "Spread Spectrum parameters including frequency hopping")
 (ComplexComponent (FieldName "SpreadSpectrumMode")
(DataType "SpreadSpectrumEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SINCGARSModulationParameters")
(DataType "SINCGARSModulationStruct")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "AudioDataType")
 (ComplexComponent (FieldName "StreamTag")
(DataType "unsigned long long")
 (Cardinality "1")
 (Units "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "The tag of the audio stream the packet belongs to.")
 )
 (ComplexComponent (FieldName "EncodingType")
(DataType "EncodingTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
```



```
 )
 (ComplexComponent (FieldName "Padding" [9])
(DataType "short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SampleRate")
(DataType "unsigned long")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "DataLength")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SampleCount")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Data")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (Class (ID 1)
 (Name "EmbeddedSystem")
 (PSCapabilities N)
 (Description "A base class used to associate sensor and emitting systems with their parent
entity object.")
 (Attribute (Name "HostObjectIdentifier")
 (DataType "RTIObjectIdStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Static)
 (UpdateCondition "N/A")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The ID of the object of which this embedded system is part of.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 )
 (Class (ID 2)
 (Name "RadioReceiver")
 (PSCapabilities PS)
 (Description "A device that converts incoming electromagnetic waves in the radio frequency
range into information.")
 (SuperClass 1)
```



```
 (Attribute (Name "ReceivedPower")
 (DataType "float")
 (Cardinality "1")
 (Units "dB-milliwatts")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> RX PWR THRSH" [8])
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The power of the received transmission.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "ReceivedTransmitterIdentifier")
 (DataType "RTIObjectIdStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The identification of the transmitter that generated the received radio sig-
nal.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "ReceiverOperationalStatus")
 (DataType "ReceiverOperationalStatusEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The state of the radio receiver.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 )
 (Class (ID 3)
 (Name "RadioTransmitter")
 (PSCapabilities PS)
 (Description "A device that sends out information encoded in electromagnetic waves in the
radio frequency range.")
 (SuperClass 1)
 (Attribute (Name "AntennaPatternData" [13])
 (DataType "AntennaPatternStruct")
 (Cardinality "0+")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
```



```
 (Description "The radiation pattern of the radio\'s antenna.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "Encryption")
 (DataType "CryptographicStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The data specifying the encryption of the transmitted signal.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "Frequency")
 (DataType "FrequencyStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> RADIO FREQ THRSH" [8])
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The radio frequency and bandwidth of transmitted radio signals.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "RFModulation")
 (DataType "RFModulationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The system and type of modulation applied to the transmitted radio signal.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "SpreadSpectrum")
 (DataType "SpreadSpectrumStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Parameters describing spread spectrum characteristics of transmission")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Attribute (Name "TransmitState")
```



```
 (DataType "TransmitStateStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Data defines the transmission state power, activity, and input source.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
)
 (Attribute (Name "WorldLocation")
 (DataType "WorldLocationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> TRANS_POS_THRSH_DFLT" [8])
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The location of the radio transmitter in the world coordinate system.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
)
)
 (Class (ID 4)
 (Name "BaseEntity" [15])
 (PSCapabilities S)
 (Description "Object represented in the simulated world.")
 (Attribute (Name "WorldLocation")
 (DataType "WorldLocationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Location of BaseEntity")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
)
 (Attribute (Name "Orientation")
 (DataType "OrientationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Orientation of Base Entity")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
)
)
 (Interaction (ID 1)
 (Name "RadioSignal")
 (ISRType N)
 (Description "The wireless transmission and reception of audio or digital data by means of
electromagnetic waves.")
```



```
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 )
 (Interaction (ID 2)
 (Name "EncodedAudioRadioSignal")
 (ISRType IR)
 (Description "A form of radio signal, where the signal is voice/audio data encoded according
to a standard encoding scheme.")
 (DeliveryCategory "best_effort")
 (MessageOrdering "receive")
 (SuperInteraction 1)
 (Parameter (Name "AudioData")
 (DataType "AudioDataType")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "Data Contained in the audio interaction. This data is contained in a single
parameter because (1) it is high-volume data, and efficiency is crucial to handling it, and (2)
there are no independent fields that could be subscribed to independently and still have a sen-
sible data structure.")
 )
 )
 (Note (NoteNumber 1)
 (NoteText "This is the unique ObjectName associated with each object instance. The user can
define the name to be used in the registerObjectInstance RTI call. If user does not define the
name then the RTI will generate a unique name for the object. RTI generated names may be fairly
long, so federations wishing to conserve bandwidth may wish to implement their own object nam-
ing scheme. \r\n\r\nThe ObjectName is provided by the RTI in the discoverObjectInstance call.
The user can also obtain the ObjectName for a particular object instance using the getObject-
InstanceName call."))

 (Note (NoteNumber 2)
 (NoteText "The current OMT standard does not allow the definition of variant records in com-
plex data types, i.e. where a field may be present or absent according to the state of another
variable (although the RTI does not disallow this, since the application is responsible for the
data marshalling of complex data types). The RPR-FOM uses the convention that a cardinality of
0-1 denotes a variant record field."))

 (Note (NoteNumber 3)
 (NoteText "The contents of the AntennaPatternStruct complex datatype depends on the value of
the AntennaPatternType attribute as follows:\r\n\r\nValues: \"Other\" or \"OmniDirec-
tional\"\r\n Structure is empty\r\n\r\nValue: \"Beam\"\r\n Structure contains 1 \"BeamAn-
tenna\" field and 0 SphericalHarmonicAntenna\" fields\r\n\r\nValue: \"SphericalHarmonic\"\r\n
Structure contains 0 \"BeamAntenna\" field and 1 SphericalHarmonicAntenna\" fields"))

 (Note (NoteNumber 4)
 (NoteText "This is an 8-bit enumeration"))

 (Note (NoteNumber 5)
 (NoteText "This is a 16-bit enumeration"))

 (Note (NoteNumber 6)
 (NoteText "This is a 32-bit enumeration"))

 (Note (NoteNumber 7)
 (NoteText "This structure is taken directly from the IEEE 1278.1-1995 (DIS) definition of the
Spherical Harmonic Antenna Pattern record (section 5.2.4.3). It does not comply with the DIS
field alignment rules (which is an error in the DIS standard). Therefore some federates may not
```



```
be able to process this structure directly, but must copy the contents a field at a time to an
equivalent structure that does comply with the DIS field alignment rules."))

 (Note (NoteNumber 8)
 (NoteText "This attribute is updated if the current value differs from the previously updated
value by more than the value specified by the symbolic name (see section 5.1.4 of IEE 1278.1-
1995 for the actual values of the symbolic names)"))

 (Note (NoteNumber 9)
 (NoteText "All padding fields shall be set to the value 0"))

 (Note (NoteNumber 10)
 (NoteText "The value of the MajorModulationType determines which of the remaining modulation
type fields are used to described the modulation detail."))

 (Note (NoteNumber 11)
 (NoteText "These fields are only used when the SINCGARS radio is in FH mode. They determine
the hopping pattern of the SINCGARS radio. If any one of these fields is different, except FH
Synchronization Time Offset, the hopping pattern is different. Radios on different hopping
patterns shall not communicate."))

 (Note (NoteNumber 12)
 (NoteText "CryptoGraphic Mode, referenced in the RPR fom, is here implied in the Cryptographic
system.Additionally, the enumerated Cryptographic systems given here are only a convience.
When determining cryptographic compatability, the Cryptographic system fields are simply com-
pared, without regard to their value. A system and key both zero implies an unencrypted emis-
sion."))

 (Note (NoteNumber 13)
 (NoteText "This field is not currently supported by the ASTi radio. However, when anistropic
antennae are modeled in the ASTi radio, it will follow these data formats, plus additional ones
(if required)"))

 (Note (NoteNumber 14)
 (NoteText "This structure is preceded by a 16 bit field, so the padding does, in fact, come
out correctly"))

 (Note (NoteNumber 15)
 (NoteText "The ASTi system does not produce entities. It subscribes to their position for the
purpose of doing entity attach."))

)
```


Version 3.0 OMT File

```
(DIF HLA-OMT v1.3 (TYPE Single))
(ObjectModel (Name "ASTi Radio SOM")
 (VersionNumber "3.0")
 (Type FOM)
 (Purpose "To support real time,person in the loop radio simulation")
 (ApplicationDomain "Real time, platform level simulations.\r\n")
 (SponsorOrgName "ASTi")
 (POCHonorificName "Mr")
 (POCFirstName "David")
 (POCLastName "Nemeth")
 (POCOrgName "ASTi")
 (POCPhone "(703) 471-2104")
 (POCEmail "davidn@asti-usa.com")
 (ModificationDate 05/19/2000)
 (MOMVersion "1.3")
 (ComplexDataType (Name "AntennaPatternStruct" [2, 3])
 (ComplexComponent (FieldName "AntennaPatternType")
(DataType "AntennaPatternTypeEnum32")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamAntenna")
(DataType "BeamAntennaStruct")
 (Cardinality "0-1")
 (Units "meters")
 (Resolution "1.0")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SphericalHarmonicAntenna")
(DataType "SphericalHarmonicAntennaStruct")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "BeamAntennaStruct")
 (ComplexComponent (FieldName "BeamOrientation")
(DataType "OrientationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamAzimuthBeamwidth")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamElevationBeamwidth")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
```



```
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "ReferenceSystem")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding1" [9])
(DataType "octet")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding2" [9])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Ez")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Ex")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "BeamPhaseAngle")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "OrientationStruct")
 (ComplexComponent (FieldName "Psi")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Theta")
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Phi")
```



```
(DataType "float")
 (Cardinality "1")
 (Units "radians")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "RFModulationStruct" [10])
 (ComplexComponent (FieldName "MajorModulationType")
(DataType "MajorRFModulationTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AmplitudeModulationType")
(DataType "AmplitudeModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AmplitudeAngleModulationType")
(DataType "AmplitudeAngleModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "AngleModulationType")
(DataType "AngleModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "CombinationModulationType")
(DataType "CombinationModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "PulseModulationType")
(DataType "PulseModulationTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "UnmodulatedType")
(DataType "UnmodulatedTypeEnum16")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "RFModulationSystemType")
(DataType "RFModulationSystemTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "RTIObjectIdStruct" [1])
 (ComplexComponent (FieldName "ID")
```



```
(DataType "string")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "SINCGARSModulationStruct" [14])
 (ComplexComponent (FieldName "FHNetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "HopSetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "LockoutSetID" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "FHSynchronizationTimeOffset" [11])
(DataType "long")
 (Cardinality "1")
 (Units "seconds")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "TransmissionSecurityKey" [11])
(DataType "short")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "SphericalHarmonicAntennaStruct" [7])
 (ComplexComponent (FieldName "Order")
(DataType "char")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
```



```
 (ComplexComponent (FieldName "Coefficients")
(DataType "float")
 (Cardinality "1+")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "ReferenceSystem")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "WorldLocationStruct")
 (ComplexComponent (FieldName "X")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Y")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Z")
(DataType "double")
 (Cardinality "1")
 (Units "meters")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (EnumeratedDataType (Name "AmplitudeAngleModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AmplitudeAndAngle")
(Representation 1))
 )
 (EnumeratedDataType (Name "AmplitudeModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AudioFrequencyShiftKeying")
(Representation 1))
 (Enumeration (Enumerator "AmplitudeModulation")
(Representation 2))
 (Enumeration (Enumerator "ContinuousWaveModulation")
(Representation 3))
 (Enumeration (Enumerator "DoubleSideband")
(Representation 4))
 (Enumeration (Enumerator "IndependentSideband")
(Representation 5))
 (Enumeration (Enumerator "SSB_LowerSideband")
(Representation 6))
 (Enumeration (Enumerator "SSB_FullCarrier")
(Representation 7))
```



```
 (Enumeration (Enumerator "SSB_ReducedCarrier")
(Representation 8))
 (Enumeration (Enumerator "SSB_UpperSideband")
(Representation 9))
 (Enumeration (Enumerator "VestigialSideband")
(Representation 10))
 )
 (EnumeratedDataType (Name "AngleModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "FrequencyModulation")
(Representation 1))
 (Enumeration (Enumerator "FrequencyShiftKeying")
(Representation 2))
 (Enumeration (Enumerator "PhaseModulation")
(Representation 3))
 )
 (EnumeratedDataType (Name "AntennaPatternTypeEnum32" [6, 13])
 (Enumeration (Enumerator "OmniDirectional")
(Representation 0))
 (Enumeration (Enumerator "Beam")
(Representation 1))
 (Enumeration (Enumerator "SphericalHarmonic")
(Representation 2))
 )
 (EnumeratedDataType (Name "CombinationModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "AmplitudeAnglePulse")
(Representation 1))
 )
 (EnumeratedDataType (Name "CryptographicSystemTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "KY_28")
(Representation 1))
 (Enumeration (Enumerator "KY_58")
(Representation 2))
 (Enumeration (Enumerator "NarrowSpectrumSecureVoice_NSVE")
(Representation 3))
 (Enumeration (Enumerator "WideSpectrumSecureVoice_WSVE")
(Representation 4))
 (Enumeration (Enumerator "SINCGARS_ICOM")
(Representation 5))
 )
 (EnumeratedDataType (Name "EncodingTypeEnum16" [5])
 (Enumeration (Enumerator "Encoding_8-bit_mu-law")
(Representation 1))
 (Enumeration (Enumerator "CVSD_per_MIL-STD-188-113")
(Representation 2))
 (Enumeration (Enumerator "ADPCM_per_CCITT_G721")
(Representation 3))
 (Enumeration (Enumerator "Encoding_16-bit_linear_PCM")
(Representation 4))
 (Enumeration (Enumerator "Encoding_8-bit_linear_PCM")
(Representation 5))
 (Enumeration (Enumerator "VQ_<Vector_Quantization>")
(Representation 6))
 (Enumeration (Enumerator "CVSD_per_CECOM")
```



```
(Representation 7))
 )
 (EnumeratedDataType (Name "MajorRFModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Amplitude")
(Representation 1))
 (Enumeration (Enumerator "AmplitudeAndAngle")
(Representation 2))
 (Enumeration (Enumerator "Angle")
(Representation 3))
 (Enumeration (Enumerator "Combination")
(Representation 4))
 (Enumeration (Enumerator "Pulse")
(Representation 5))
 (Enumeration (Enumerator "Unmodulated")
(Representation 6))
 )
 (EnumeratedDataType (Name "PulseModulationTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Pulse")
(Representation 1))
 )
 (EnumeratedDataType (Name "RadioInputSourceEnum8" [4])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Pilot")
(Representation 1))
 (Enumeration (Enumerator "Copilot")
(Representation 2))
 (Enumeration (Enumerator "FirstOfficer")
(Representation 3))
 (Enumeration (Enumerator "Driver")
(Representation 4))
 (Enumeration (Enumerator "Loader")
(Representation 5))
 (Enumeration (Enumerator "Gunner")
(Representation 6))
 (Enumeration (Enumerator "Commander")
(Representation 7))
 (Enumeration (Enumerator "DigitalDataDevice")
(Representation 8))
 (Enumeration (Enumerator "Intercom")
(Representation 9))
 )
 (EnumeratedDataType (Name "ReceiverOperationalStatusEnum16" [5])
 (Enumeration (Enumerator "Off")
(Representation 0))
 (Enumeration (Enumerator "OnButNotReceiving")
(Representation 1))
 (Enumeration (Enumerator "OnAndReceiving")
(Representation 2))
 )
 (EnumeratedDataType (Name "RFModulationSystemTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "Generic")
(Representation 1))
```



```
 (Enumeration (Enumerator "HQ")
(Representation 2))
 (Enumeration (Enumerator "HQII")
(Representation 3))
 (Enumeration (Enumerator "HQIIA")
(Representation 4))
 (Enumeration (Enumerator "SINCGARS")
(Representation 5))
 (Enumeration (Enumerator "CCTT_SINCGARS")
(Representation 6))
 )
 (EnumeratedDataType (Name "TransmitterOperationalStatusEnum8" [4])
 (Enumeration (Enumerator "Off")
(Representation 0))
 (Enumeration (Enumerator "OnButNotTransmitting")
(Representation 1))
 (Enumeration (Enumerator "OnAndTransmitting")
(Representation 2))
 )
 (EnumeratedDataType (Name "UnmodulatedTypeEnum16" [5])
 (Enumeration (Enumerator "Other")
(Representation 0))
 (Enumeration (Enumerator "ContinuousWaveEmission")
(Representation 1))
 )
 (ComplexDataType (Name "CryptographicStruct" [12])
 (ComplexComponent (FieldName "CryptographicSystem")
(DataType "CryptographicSystemTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "EncryptionKey")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "FrequencyStruct")
 (ComplexComponent (FieldName "Frequency")
(DataType "unsigned long long")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "FrequencyBandwidth")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (EnumeratedDataType (Name "SpreadSpectrumEnum16")
 (Enumeration (Enumerator "None")
(Representation 0))
 (Enumeration (Enumerator "SINCGARSFrequencyHop")
(Representation 5))
 )
```



```
 (ComplexDataType (Name "TransmitStateStruct")
 (ComplexComponent (FieldName "TransmitterOperationalStatus")
(DataType "TransmitterOperationalStatusEnum8")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "RadioInputSource")
(DataType "RadioInputSourceEnum8")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding" [9])
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "TransmittedPower")
(DataType "float")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "StreamTag")
(DataType "unsigned long long")
 (Cardinality "1")
 (Units "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "SpreadSpectrumStruct")
 (ComplexComponent (FieldName "SpreadSpectrumMode")
(DataType "SpreadSpectrumEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SINCGARSModulationParameters")
(DataType "SINCGARSModulationStruct")
 (Cardinality "0-1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (ComplexDataType (Name "AudioDataType")
 (ComplexComponent (FieldName "StreamTag")
(DataType "unsigned long long")
 (Cardinality "1")
 (Units "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "EncodingType")
(DataType "EncodingTypeEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
```



```
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Padding" [9])
(DataType "short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SampleRate")
(DataType "unsigned long")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "DataLength")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "SampleCount")
(DataType "unsigned short")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 (ComplexComponent (FieldName "Data")
(DataType "any")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 )
 )
 (Class (ID 1)
 (Name "EmbeddedSystem")
 (PSCapabilities N)
 (Description "A base class used to associate sensor and emitting systems with their parent
entity object.")
 (Attribute (Name "HostObjectIdentifier")
 (DataType "RTIObjectIdStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Static)
 (UpdateCondition "N/A")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The ID of the object of which this embedded system is part of.")
 )
 )
 (Class (ID 2)
 (Name "RadioReceiver")
 (PSCapabilities PS)
 (Description "A device that converts incoming electromagnetic waves in the radio frequency
range into information.")
 (SuperClass 1)
 (Attribute (Name "ReceivedPower")
```



```
 (DataType "float")
 (Cardinality "1")
 (Units "dB-milliwatts")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> RX PWR THRSH" [8])
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The power of the received transmission.")
 )
 (Attribute (Name "ReceivedTransmitterIdentifier")
 (DataType "RTIObjectIdStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The identification of the transmitter that generated the received radio sig-
nal.")
 )
 (Attribute (Name "ReceiverOperationalStatus")
 (DataType "ReceiverOperationalStatusEnum16")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The state of the radio receiver.")
 )
 )
 (Class (ID 3)
 (Name "RadioTransmitter")
 (PSCapabilities PS)
 (Description "A device that sends out information encoded in electromagnetic waves in the
radio frequency range.")
 (SuperClass 1)
 (Attribute (Name "AntennaPatternData" [13])
 (DataType "AntennaPatternStruct")
 (Cardinality "0+")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The radiation pattern of the radio\'s antenna.")
 )
 (Attribute (Name "Encryption")
 (DataType "CryptographicStruct")
 (Cardinality "1")
 (Units "N/A")
 (Resolution "N/A")
```



```
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The data specifying the encryption of the transmitted signal.")
 )
 (Attribute (Name "Frequency")
 (DataType "FrequencyStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> RADIO FREQ THRSH" [8])
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The radio frequency and bandwidth of transmitted radio signals.")
 )
 (Attribute (Name "RFModulation")
 (DataType "RFModulationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "The system and type of modulation applied to the transmitted radio signal.")
 )
 (Attribute (Name "SpreadSpectrum")
 (DataType "SpreadSpectrumStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "On change")
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Parameters describing spread spectrum characteristics of transmission")
 )
 (Attribute (Name "TransmitState")
 (DataType "TransmitStateStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Data defines the transmission state power, activity, and input source.")
 )
 (Attribute (Name "WorldLocation")
 (DataType "WorldLocationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (UpdateCondition "> TRANS_POS_THRSH_DFLT" [8])
 (TransferAccept N)
```



```
 (UpdateReflect UR)
 (Description "The location of the radio transmitter in the world coordinate system.")
 )
 )
 (Class (ID 4)
 (Name "BaseEntity" [15])
 (PSCapabilities S)
 (Description "Object represented in the simulated world.")
 (Attribute (Name "WorldLocation")
 (DataType "WorldLocationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Location of BaseEntity")
 )
 (Attribute (Name "Orientation")
 (DataType "OrientationStruct")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (UpdateType Conditional)
 (TransferAccept N)
 (UpdateReflect UR)
 (Description "Orientation of Base Entity")
 )
 )
 (Interaction (ID 1)
 (Name "RadioSignal")
 (ISRType N)
 (Description "The wireless transmission and reception of audio or digital data by means of
electromagnetic waves.")
 )
 (Interaction (ID 2)
 (Name "EncodedAudioRadioSignal")
 (ISRType IR)
 (Description "A form of radio signal, where the signal is voice/audio data encoded according
to a standard encoding scheme.")
 (SuperInteraction 1)
 (Parameter (Name "AudioData")
 (DataType "AudioDataType")
 (Cardinality "1")
 (Accuracy "perfect")
 (AccuracyCondition "always")
 (Description "Data Contained in the audio interaction. This data is contained in a single
parameter because (1) it is high-volume data, and efficiency is crucial to handling it, and (2)
there are no independent fields that could be subscribed to independently and still have a sen-
sible data structure.")
 )
 )
 (Note (NoteNumber 1)
 (NoteText "This is the unique ObjectName associated with each object instance. The user can
define the name to be used in the registerObjectInstance RTI call. If user does not define the
name then the RTI will generate a unique name for the object. RTI generated names may be fairly
long, so federations wishing to conserve bandwidth may wish to implement their own object nam-
ing scheme. \r\n\r\nThe ObjectName is provided by the RTI in the discoverObjectInstance call.
```



```
The user can also obtain the ObjectName for a particular object instance using the getObject-
InstanceName call."))

 (Note (NoteNumber 2)
 (NoteText "The current OMT standard does not allow the definition of variant records in com-
plex data types, i.e. where a field may be present or absent according to the state of another
variable (although the RTI does not disallow this, since the application is responsible for the
data marshalling of complex data types). The RPR-FOM uses the convention that a cardinality of
0-1 denotes a variant record field."))

 (Note (NoteNumber 3)
 (NoteText "The contents of the AntennaPatternStruct complex datatype depends on the value of
the AntennaPatternType attribute as follows:\r\n\r\nValues: \"Other\" or \"OmniDirec-
tional\"\r\n Structure is empty\r\n\r\nValue: \"Beam\"\r\n Structure contains 1 \"BeamAn-
tenna\" field and 0 SphericalHarmonicAntenna\" fields\r\n\r\nValue: \"SphericalHarmonic\"\r\n
Structure contains 0 \"BeamAntenna\" field and 1 SphericalHarmonicAntenna\" fields"))

 (Note (NoteNumber 4)
 (NoteText "This is an 8-bit enumeration"))

 (Note (NoteNumber 5)
 (NoteText "This is a 16-bit enumeration"))

 (Note (NoteNumber 6)
 (NoteText "This is a 32-bit enumeration"))

 (Note (NoteNumber 7)
 (NoteText "This structure is taken directly from the IEEE 1278.1-1995 (DIS) definition of the
Spherical Harmonic Antenna Pattern record (section 5.2.4.3). It does not comply with the DIS
field alignment rules (which is an error in the DIS standard). Therefore some federates may not
be able to process this structure directly, but must copy the contents a field at a time to an
equivalent structure that does comply with the DIS field alignment rules."))

 (Note (NoteNumber 8)
 (NoteText "This attribute is updated if the current value differs from the previously updated
value by more than the value specified by the symbolic name (see section 5.1.4 of IEE 1278.1-
1995 for the actual values of the symbolic names)"))

 (Note (NoteNumber 9)
 (NoteText "All padding fields shall be set to the value 0"))

 (Note (NoteNumber 10)
 (NoteText "The value of the MajorModulationType determines which of the remaining modulation
type fields are used to described the modulation detail."))

 (Note (NoteNumber 11)
 (NoteText "These fields are only used when the SINCGARS radio is in FH mode. They determine
the hopping pattern of the SINCGARS radio. If any one of these fields is different, except FH
Synchronization Time Offset, the hopping pattern is different. Radios on different hopping
patterns shall not communicate."))

 (Note (NoteNumber 12)
 (NoteText "CryptoGraphic Mode, referenced in the RPR fom, is here implied in the Cryptographic
system.Additionally, the enumerated Cryptographic systems given here are only a convience.
When determining cryptographic compatability, the Cryptographic system fields are simply com-
pared, without regard to their value. A system and key both zero implies an unencrypted emis-
sion."))
```



```
 (Note (NoteNumber 13)
 (NoteText "This field is not currently supported by the ASTi radio. However, when anistropic
antennae are modeled in the ASTi radio, it will follow these data formats, plus additional ones
(if required)"))

 (Note (NoteNumber 14)
 (NoteText "This structure is preceded by a 16 bit field, so the padding does, in fact, come
out correctly"))

 (Note (NoteNumber 15)
 (NoteText "The ASTi system does not produce entities. It subscribes to their position for the
purpose of doing entity attach."))

)
```

