

Application Program Interface Supplement  
to the  
Software Communications Architecture Specification

**APPENDIX E**

**Physical Non-Real-Time Building Block Service Definition**

Revision Summary

1.0	Initial Release
-----	-----------------

## Table of Contents

<b>E.1 INTRODUCTION.....</b>	<b>E-1</b>
E.1.1 OVERVIEW .....	E-1
E.1.2 SERVICE LAYER DESCRIPTION. ....	E-1
E.1.3 MODES OF SERVICE.....	E-2
E.1.4 SERVICE STATES .....	E-2
E.1.5 REFERENCED DOCUMENTS.....	E-2
<b>E.2 UUID.....</b>	<b>E-2</b>
<b>E.3 SERVICES.....</b>	<b>E-3</b>
E.3.1 ANTENNA CONTROL.....	E-3
E.3.2 TRANSCEIVER SETUP.....	E-4
E.3.3 MODULATION SETUP.....	E-4
E.3.4 MEDIA SETUP.....	E-4
E.3.5 RADIO MODE .....	E-5
E.3.5.1 Radio Mode State Chart.....	E-5
E.3.6 RECEIVE TERMINATION. ....	E-5
E.3.7 TRANSMIT INHIBIT. ....	E-6
E.3.8 PHYSICAL MANAGEMENT. ....	E-6
E.3.9 EXAMPLES.....	E-6
E.3.9.1 Antenna Control. ....	E-6
E.3.9.2 Transceiver Setup.....	E-8
E.3.9.3 Modulation Setup.....	E-8
E.3.9.4 Media Setup.....	E-11
E.3.9.5 Radio Mode.....	E-13
<b>E.4 SERVICE PRIMITIVES.....</b>	<b>E-14</b>
E.4.1 ANTENNA CONTROL.....	E-14
E.4.1.1 SetRxAntenna.....	E-14
E.4.1.2 setTxAntenna. ....	E-14
E.4.2 TRANSCEIVER SETUP.....	E-15
E.4.2.1 Set Up Receiver Parameters.....	E-15
E.4.2.2 Set Up Transmitter Parameters. ....	E-15
E.4.3 MODULATION SETUP.....	E-16
E.4.3.1 Set Up Receiver Modulation.....	E-16
E.4.3.2 Set Up Transmitter Modulation.....	E-16
E.4.4 MEDIA SETUP.....	E-17
E.4.4.1 Set Up Media Type. ....	E-17
E.4.5 RADIO MODE .....	E-18
E.4.5.1 Radio Mode Set Up.....	E-18
E.4.6 RECEIVE TERMINATION. ....	E-18
E.4.6.1 Drop Capture.....	E-18
E.4.6.2 Abort receive.....	E-19
E.4.7 TRANSMIT INHIBIT.....	E-19

E.4.7.1 Inhibit Transmit.....	E-19
E.4.8 PHYSICAL MANAGEMENT. ....	E-20
E.4.8.1 Maximum Transmission Unit.....	E-20
E.4.8.2 Minimum Transmission Unit. ....	E-20
<b>E.5 ALLOWABLE SEQUENCE OF SERVICE PRIMITIVES.....</b>	<b>E-21</b>
<b>E.6 UTILIZATION OF PHYSICAL NON-REAL-TIME BUILDING BLOCKS .....</b>	<b>E-22</b>
<b>E.7 PRECEDENCE OF SERVICE PRIMITIVES.....</b>	<b>E-23</b>
<b>E.8 SERVICE USER GUIDELINES. ....</b>	<b>E-23</b>
<b>E.9 SERVICE PROVIDER-SPECIFIC INFORMATION.....</b>	<b>E-23</b>
<b>E.10 IDL.....</b>	<b>E-23</b>
<b>E.11 UML. ....</b>	<b>E-23</b>

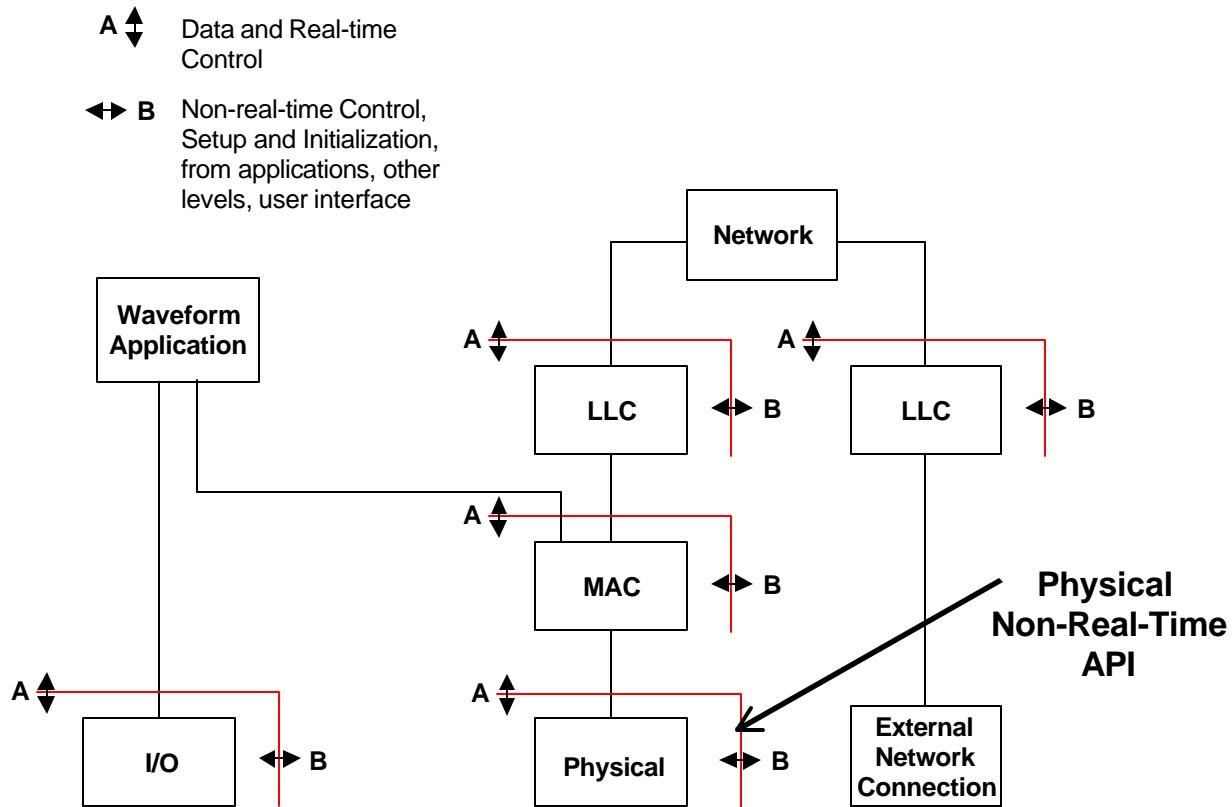
## Table of Figures

Figure 1. Service Definition Overview.....	E-1
Figure 2. Antenna Control Building Block .....	E-3
Figure 3. Transceiver Setup Building Block.....	E-4
Figure 4. Modulation Setup Building Block .....	E-4
Figure 5. Media Setup Building Block.....	E-4
Figure 6. Radio Mode Building Block.....	E-5
Figure 7. Radio Mode State Chart.....	E-5
Figure 8. Receive Termination Interface.....	E-5
Figure 9. Transmit Inhibit Interface .....	E-6
Figure 10. Physical Management.....	E-6
Figure 11. Antenna Control Instantiation.....	E-7
Figure 12. Transceiver Setup Instantiation.....	E-8
Figure 13. FSK Modulation Setup Instantiation.....	E-9
Figure 14. FM Modulation Setup Instantiation.....	E-10
Figure 15. AM Modulation Setup Instantiation.....	E-11
Figure 16. Media Setup Instantiation.....	E-12
Figure 17. Radio Mode Instantiation.....	E-13

## E.1 INTRODUCTION.

### E.1.1 Overview.

This document specifies an SCA conformant building block that is used by the physical layer API Service Definition. Physical Layer Services are grouped into Service Groups or Building Blocks to define abstract services to foster software reuse and commonality between different API implementations. Non-real-time physical layer services provide Service Users with methods to send non-real-time (i.e. independent of user data transfer) configuration and control data into the physical layer. Real-time control and signals are communicated via the packet interface building block and the real-time physical layer building block.



**Figure 1. Service Definition Overview**

### E.1.2 Service Layer Description.

The primary physical layer Service User is the MAC layer.

Each BB is an abstract analysis Service Definition that must be used with concrete types to be realized. Physical layer concrete types are waveform-specific. An API that uses the BB is limited to a specific waveform (e.g., SINCGARS or HF ALE). APIs utilize only the BBs that apply to the defining waveform.

E.1.3 Modes of Service.

There are no specific Modes of Service. This document describes an SCA building block that provides a generalization of the physical layer interface. To form a complete service layer or API, several physical building blocks may need to be combined. When used in this manner, the BB combination may be considered as constituting a physical layer configuration and control service available to other layers of the SCA model.

E.1.4 Service States.

The BB assumes no states.

E.1.5 Referenced Documents.

Software Communication Architecture Specification, v2.0, November 17, 2000.

**E.2 UUID.**

Not Applicable.

### E.3 SERVICES.

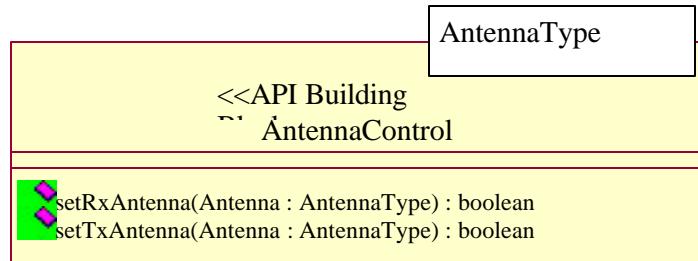
The features of the interface are defined in terms of the services and the individual primitives that may flow between the Service User and Service Provider.

The services are tabulated below and described more fully in the remainder of this section.

SERVICE GROUP	SERVICE	PRIMITIVE
Antenna Control	Set Receive Antenna	setRxAntenna
	Set Transmit Antenna	setTxAntenna
Transceiver Setup	Set Up Receiver Parameters	setUpReceiverParams
	Set Up Transmitter Parameters	setUpTransmitterParams
Modulation Setup	Set Up Receiver Modulation	setUpReceiverModulation
	Set Up Transmitter Modulation	setUpTransmitterModulation
Media Setup	Set Up Media Type	setUpMediaType
Radio Mode	Set Radio Mode	setRadioMode
Receive Termination	Drop Capture	dropCapture
	Abort Receive	abortReceive
Transmit Inhibit	Inhibit Transmit	inhibitTransmit
Physical Management	Maximum Transmission Unit	MaxTU
	Minimum Transmission Unit	MinTU

#### E.3.1 Antenna Control.

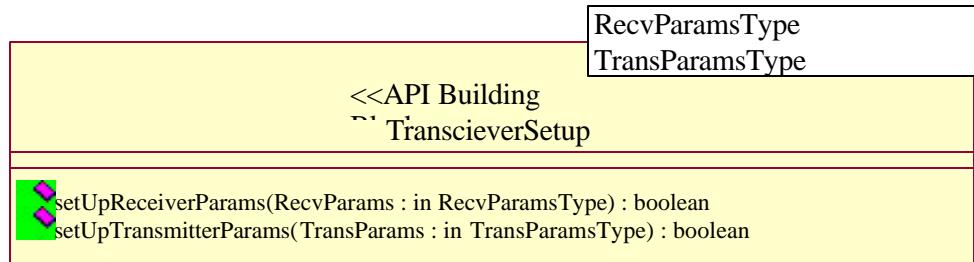
Antenna Control determines which antennas are connected to the transceiver. Different antenna configurations may be accommodated by different instantiations of the parameterized antenna type. This allows custom antenna configurations without changing the basic building block.



**Figure 2. Antenna Control Building Block**

### E.3.2 Transceiver Setup.

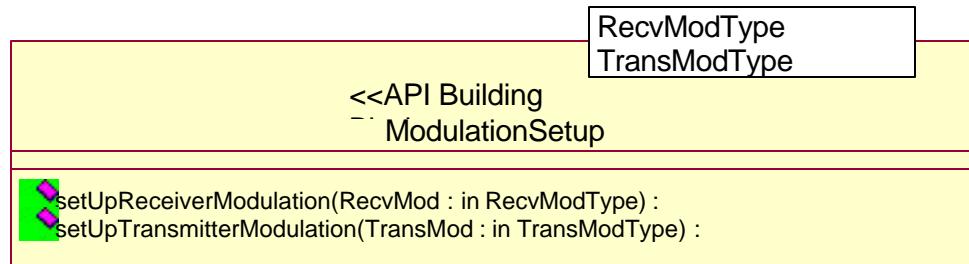
This class sets up the transceiver parameters that are not modulation or media dependent. An example is shown in section E.3.9.2.



**Figure 3. Transceiver Setup Building Block**

### E.3.3 Modulation Setup.

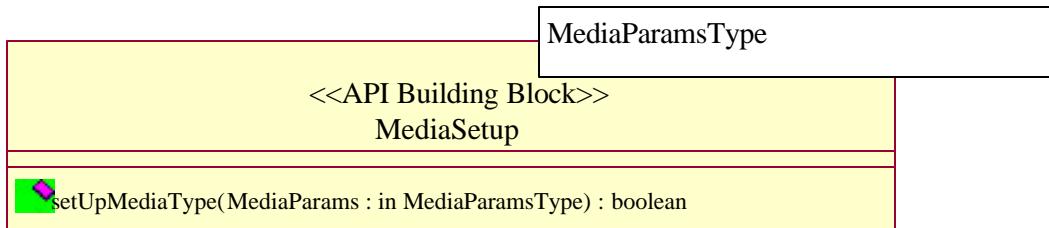
Modulation setup determines the transceiver modulation and demodulation type and settings. Examples of FSK, FM, and AM are in section E.3.9.3.



**Figure 4. Modulation Setup Building Block**

### E.3.4 Media Setup.

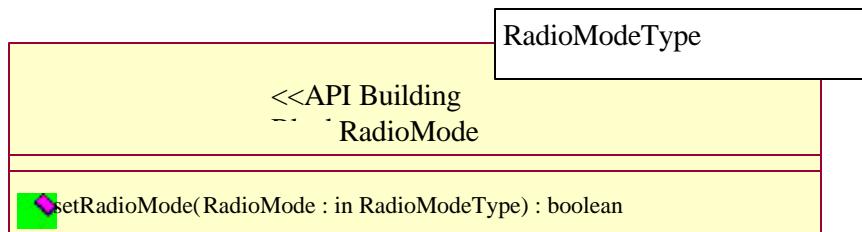
This service group provides the setup necessary for voice, data, etc. An example for voice is in section E.3.9.4.



**Figure 5. Media Setup Building Block**

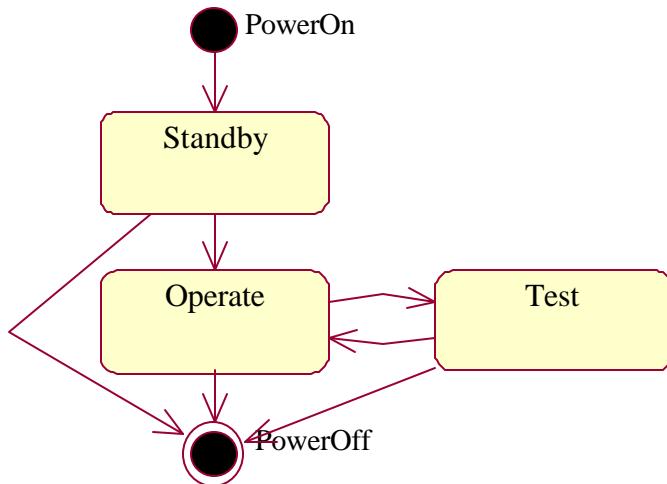
### E.3.5 Radio Mode.

This service group sets the transceiver to off, setup, operational, test, etc.



**Figure 6. Radio Mode Building Block**

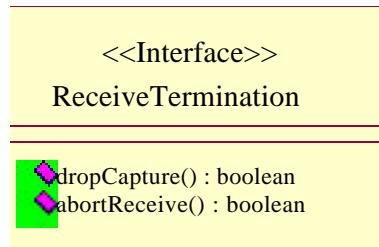
#### E.3.5.1 Radio Mode State Chart.



**Figure 7. Radio Mode State Chart**

### E.3.6 Receive Termination.

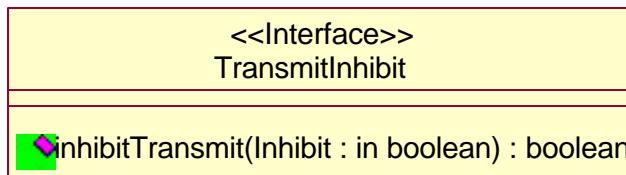
This service group terminates the current reception. Drop Capture returns the receiver to receiver acquire mode and Abort Receive turns off the receiver.



**Figure 8. Receive Termination Interface**

### E.3.7    Transmit Inhibit.

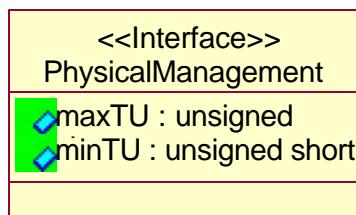
This service group enforces radio silence on the transceiver. Note that may be more than one transceiver in a JTRS physical realization.



**Figure 9. Transmit Inhibit Interface**

### E.3.8    Physical Management.

This service unit provides the user with the maximum and minimum transmission units.



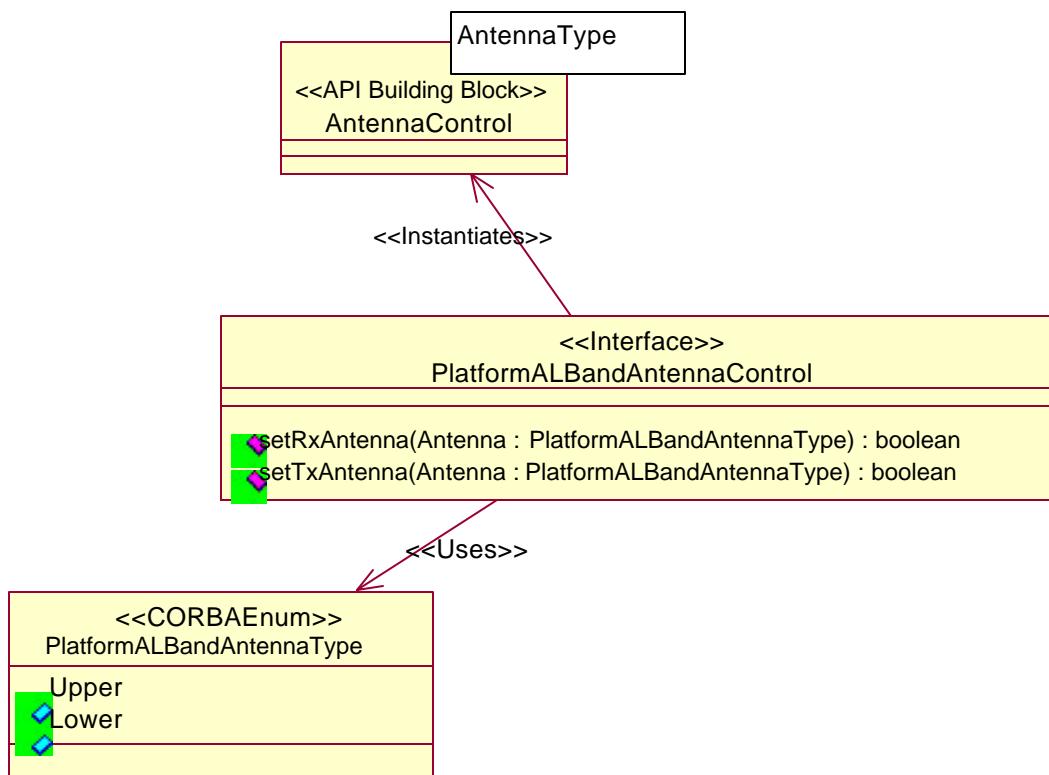
**Figure 10. Physical Management**

### E.3.9    EXAMPLES.

The following are examples of instantiated services using the previous parameterized services

#### E.3.9.1    Antenna Control.

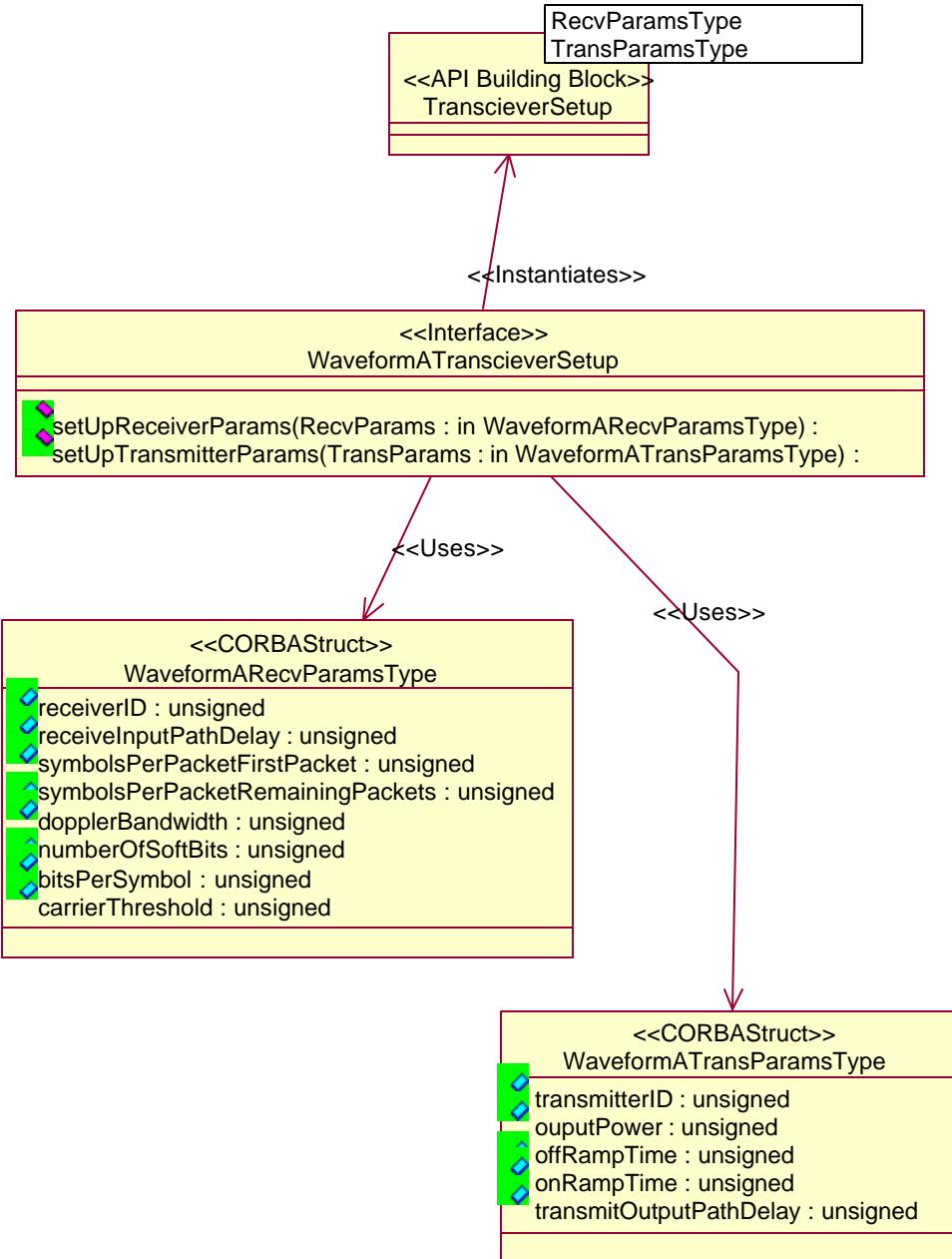
This example shows the usage of two antennas, an upper antenna and a lower antenna.



**Figure 11. Antenna Control Instantiation**

### E.3.9.2 Transceiver Setup.

This example provides the means to configure the physical layer with the primitives that are not modulation or media dependent.

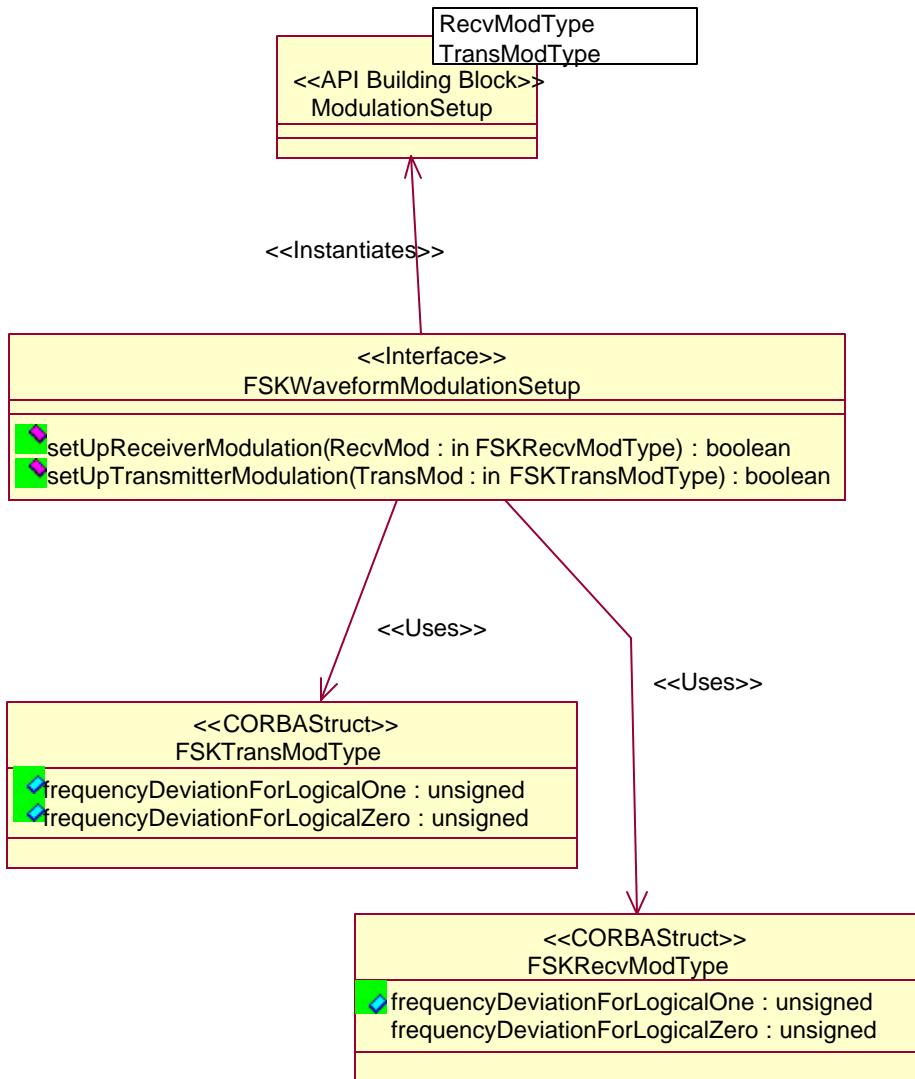


**Figure 12. Transceiver Setup Instantiation**

### E.3.9.3 Modulation Setup.

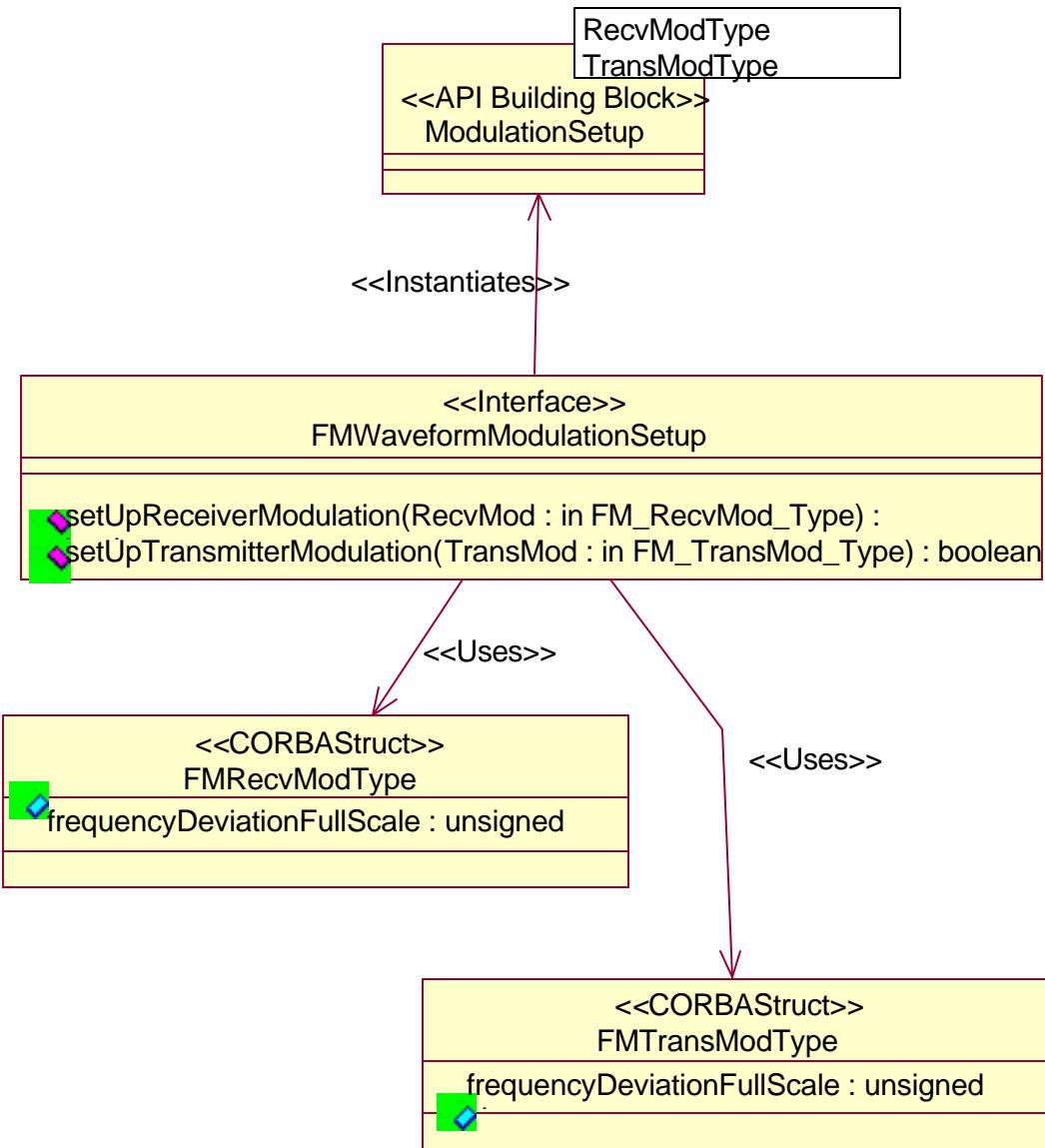
These examples provide the primitives for FSK, FM, and AM.

### E.3.9.3.1 FSK.



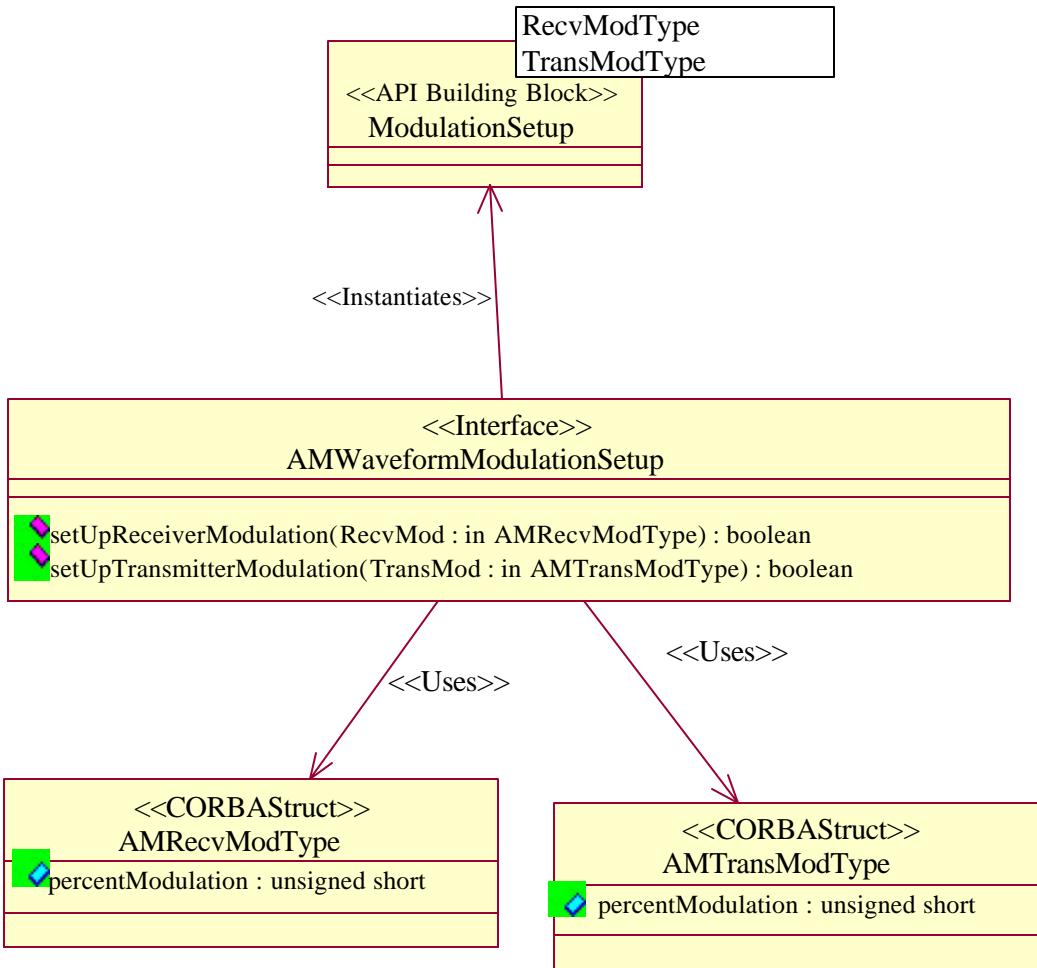
**Figure 13. FSK Modulation Setup Instantiation**

### E.3.9.3.2 FM.



**Figure 14. FM Modulation Setup Instantiation**

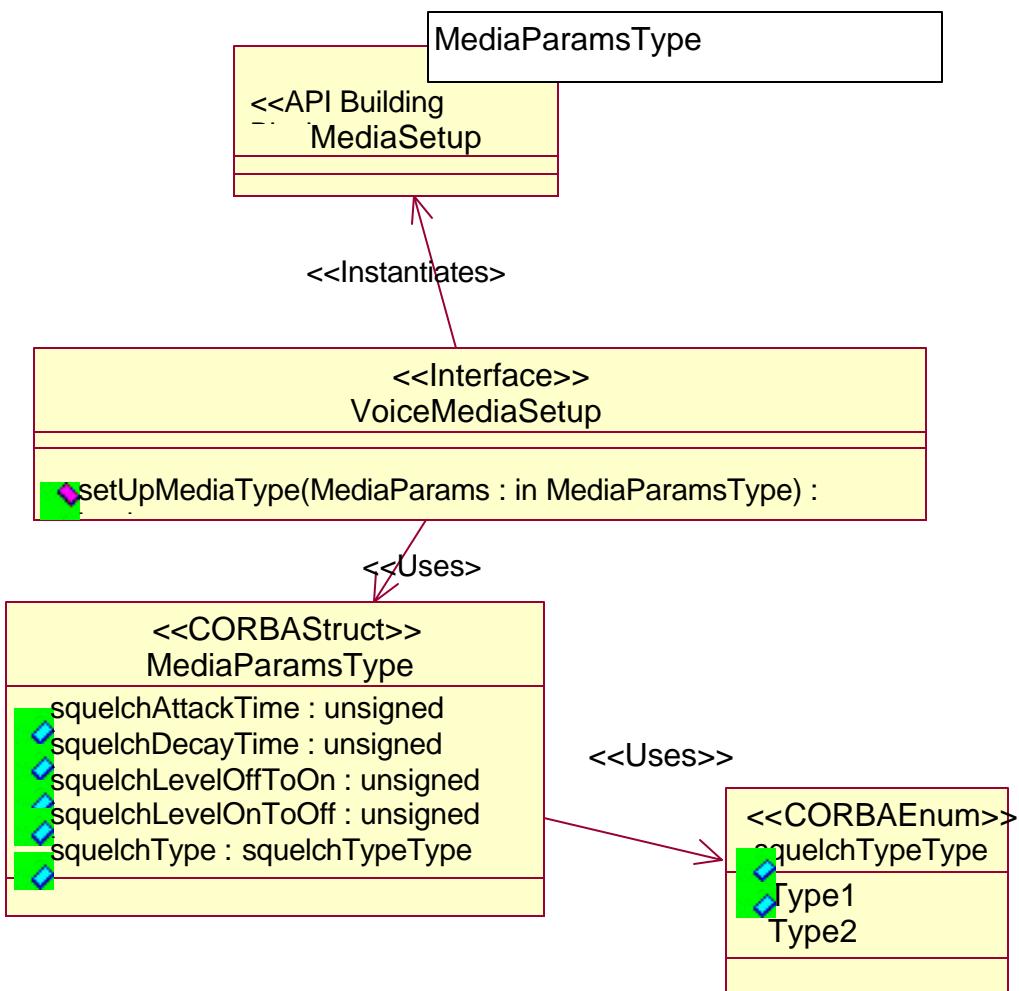
### E.3.9.3.3 AM.



**Figure 15. AM Modulation Setup Instantiation**

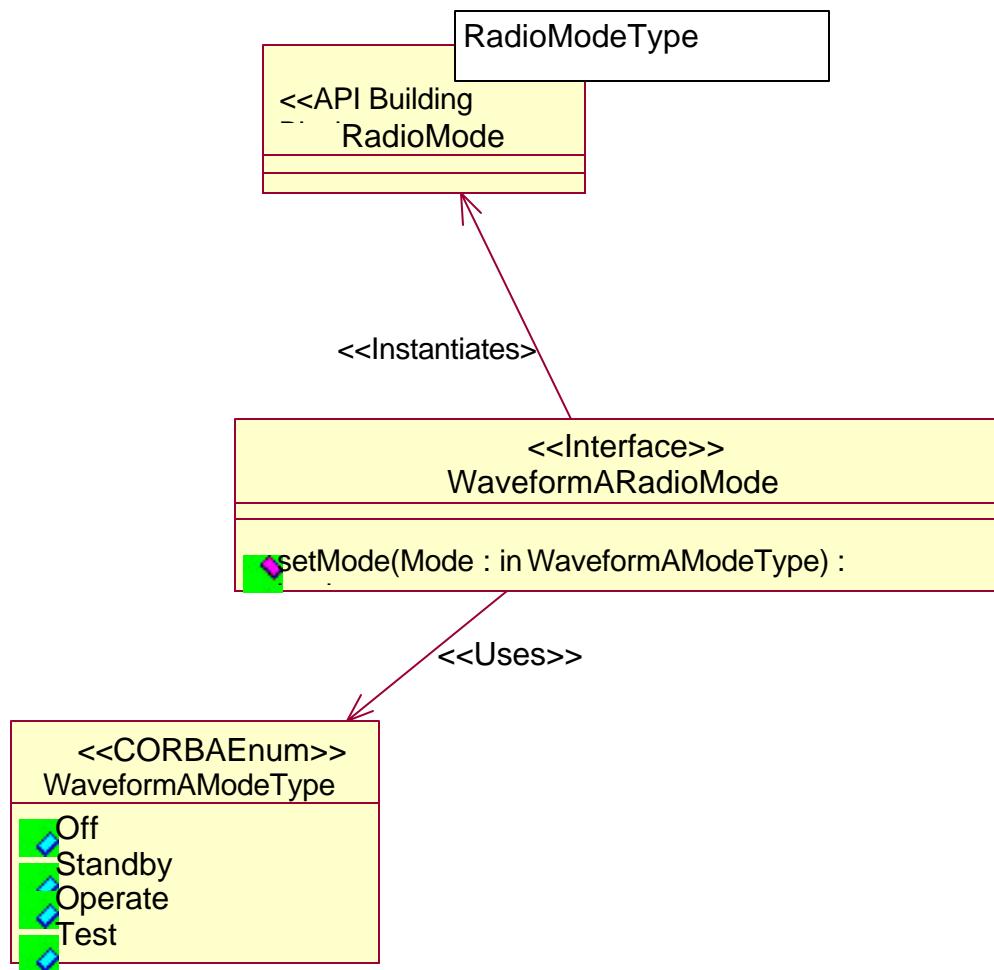
### E.3.9.4 Media Setup.

This example covers the primitives that are specific to voice.



**Figure 16. Media Setup Instantiation**

E.3.9.5 Radio Mode.



**Figure 17. Radio Mode Instantiation**

## E.4 SERVICE PRIMITIVES.

### E.4.1 Antenna Control.

#### E.4.1.1 SetRxAntenna.

This primitive connects the receiver to one of the available antennas.

##### E.4.1.1.1 Synopsis.

boolean setRxAntenna (in AntennaType Antenna)

##### E.4.1.1.2 Parameters.

AntennaType is an enumeration type that lists all of the available antennas in the given installation.

##### E.4.1.1.3 State.

This command is valid in all states.

##### E.4.1.1.4 New State.

This command does cause a state change.

##### E.4.1.1.5 Response.

TRUE if the Rx Antenna is set as specified in the call; FALSE Otherwise.

##### E.4.1.1.6 Originator.

The service user.

##### E.4.1.1.7 Errors/Exceptions.

NA.

#### E.4.1.2 setTxAntenna.

This primitive connects the transmitter to one of the available antennas.

##### E.4.1.2.1 Synopsis.

boolean setTxAntenna(in AntennaType Antenna)

##### E.4.1.2.2 Parameters.

AntennaType is an enumeration type that lists all of the available antennas in the given installation.

##### E.4.1.2.3 State.

This command is valid in all states.

##### E.4.1.2.4 New State.

This command does cause a state change.

##### E.4.1.2.5 Response.

TRUE if the Tx Antenna is set as specified in the call; FALSE Otherwise.

E.4.1.2.6 Originator.

The service user.

E.4.1.2.7 Errors/Exceptions.

NA.

#### E.4.2 Transceiver Setup.

E.4.2.1 Set Up Receiver Parameters.

This primitive sets up the parameters in the transceiver that are specific to receive but are not dependent on a specific modulation.

E.4.2.1.1 Synopsis.

```
boolean setUpReceiverParams(in RecvParamsType RecvParams )
```

E.4.2.1.2 Parameters.

RecvParamsType is a struct containing all the parameters specific to receive but independent of a particular modulation.

E.4.2.1.3 State.

This command valid in all states.

E.4.2.1.4 New State.

This command does cause a state change.

E.4.2.1.5 Response.

TRUE if Receiver Parameters is set as specified in the call FALSE Otherwise.

E.4.2.1.6 Originator.

The service user.

E.4.2.1.7 Errors/Exceptions.

NA.

E.4.2.2 Set Up Transmitter Parameters.

This primitive sets up the parameters in the transceiver that are specific to transmit but are not dependent on a specific modulation.

E.4.2.2.1 Synopsis.

```
boolean setUpTransmitterParams(in TransParamsType TransParams )
```

E.4.2.2.2 Parameters.

TransParamsType is a struct containing all the parameters specific to transmit but independent of a particular modulation.

E.4.2.2.3 State.

This command valid in all states.

E.4.2.2.4 New State.

This command does cause a state change.

E.4.2.2.5 Response.

TRUE if the Rx Antenna is set as specified in the call; FALSE Otherwise.

E.4.2.2.6 Originator.

The service user.

E.4.2.2.7 Errors/Exceptions.

NA.

E.4.3 Modulation Setup.

E.4.3.1 Set Up Receiver Modulation.

This primitive sets up the parameters in the transceiver that are specific to reception of a specific modulation.

E.4.3.1.1 Synopsis.

```
boolean setUpReceiverModulation( in RecvModType RecvMod)
```

E.4.3.1.2 Parameters.

RecvModType is a struct containing all of the parameters specific to receiving a particular modulation.

E.4.3.1.3 State.

This command valid in all states.

E.4.3.1.4 New State.

This command does cause a state change.

E.4.3.1.5 Response.

TRUE if the Rx Antenna is set as specified in the call; FALSE Otherwise.

E.4.3.1.6 Originator.

The service user.

E.4.3.1.7 Errors/Exceptions.

NA.

E.4.3.2 Set Up Transmitter Modulation.

This primitive sets up the parameters in the transceiver that are specific to the transmission of a specific modulation.

E.4.3.2.1 Synopsis.

```
boolean setUpTransmitterModulation(in TransModType TransMod )
```

E.4.3.2.2 Parameters.

TransModType is a struct containing all the parameters specific to transmitting a particular modulation.

E.4.3.2.3 State.

This command valid in all states.

E.4.3.2.4 New State.

This command does cause a state change.

E.4.3.2.5 Response.

TRUE if Transmitter Modulation is set as specified in the call; FALSE Otherwise.

E.4.3.2.6 Originator.

The service user.

E.4.3.2.7 Errors/Exceptions.

NA.

E.4.4 Media Setup.

E.4.4.1 Set Up Media Type.

This primitive sets up the parameters in the transceiver that are specific to the transmission of a specific media (voice, data, video, etc.).

E.4.4.1.1 Synopsis.

```
boolean setUpMediaType(in MediaParamsType MediaParams )
```

E.4.4.1.2 Parameters.

MediaParamsType is a struct containing all of the parameters specific to transmitting a particular media.

E.4.4.1.3 State.

This command valid in all states.

E.4.4.1.4 New State.

This command does cause a state change.

E.4.4.1.5 Response.

TRUE if the Rx Antenna is set as specified in the call; FALSE Otherwise.

E.4.4.1.6 Originator.

The service user.

E.4.4.1.7 Errors/Exceptions.

NA.

#### E.4.5 Radio Mode.

##### E.4.5.1 Radio Mode Set Up.

This primitive sets the mode of the radio, e.g., Off, Standby, Operate, Test.

###### E.4.5.1.1 Synopsis.

```
boolean setRadioMode(in RadioModeType RadioMode :)
```

###### E.4.5.1.2 Parameters.

RadioModeType is an enumeration type listing all of the settable modes of the transceiver.

###### E.4.5.1.3 State.

This command operates in all states.

###### E.4.5.1.4 New State.

The state specified in the command.

###### E.4.5.1.5 Response.

TRUE if the CarrierThreshold is set FALSE Otherwise.

###### E.4.5.1.6 Originator.

This primitive is initiated by the Service User.

###### E.4.5.1.7 Errors/Exceptions.

NA.

#### E.4.6 Receive Termination.

##### E.4.6.1 Drop Capture.

This primitive zeros all state variables relating to signal capture and demodulation and returns to receiver acquisition.

###### E.4.6.1.1 Synopsis.

```
boolean dropCapture()
```

###### E.4.6.1.2 Parameters.

dropCapture is a signal. It has no parameters.

###### E.4.6.1.3 State.

This command is valid in all states. If the transceiver is not currently receiving it has no effect.

###### E.4.6.1.4 New State.

This command does cause a state change.

###### E.4.6.1.5 Response.

TRUE if current reception is terminated or if the transceiver is not actively receiving; FALSE if an active receive is not terminated or if the transceiver does not go back into acquiring after it has been in active receive.

E.4.6.1.6 Originator.

The service user.

E.4.6.1.7 Errors/Exceptions.

NA.

E.4.6.2 Abort receive

This primitive inhibits the receive operation.

E.4.6.2.1 Synopsis.

boolean abortReceive()

E.4.6.2.2 Parameters.

This primitive has no parameters.

E.4.6.2.3 State.

This command is valid in all states. If the transceiver is not currently receiving it has no effect.

E.4.6.2.4 New State.

This command does cause a state change.

E.4.6.2.5 Response.

TRUE if current reception is terminated or if the transceiver is not actively receiving; FALSE if an active receive is not terminated.

E.4.6.2.6 Originator.

The service user.

E.4.6.2.7 Errors/Exceptions.

#### E.4.7 Transmit Inhibit

E.4.7.1 Inhibit Transmit

This primitive inhibits the transceiver from transmitting.

E.4.7.1.1 Synopsis.

boolean inhibitTransmit(in boolean Inhibit)

E.4.7.1.2 Parameters.

boolean

E.4.7.1.3 State.

This command valid in all states

E.4.7.1.4 New State.

This command causes the waveform to stop transmitting.

E.4.7.1.5 Response.

TRUE if radio silence is achieved for this transceiver; FALSE Otherwise.

E.4.7.1.6 Originator.

The service user.

E.4.7.1.7 Errors/Exceptions.

NA.

#### E.4.8 Physical Management.

Physical Management has two read only parameters determined at set up time, maxTU and minTU.

E.4.8.1 Maximum Transmission Unit.

This attribute is read by the Service User to determine the maximum transmission unit length supported by the Service Provider.

E.4.8.1.1 Synopsis.

readonly attribute unsigned short MaxTU.

E.4.8.1.2 Parameters.

Unsigned short range  $2^{16}$ .

E.4.8.1.3 State.

This command valid in all states.

E.4.8.1.4 New State.

This command does cause a state change.

E.4.8.1.5 Response.

The size of the maximum TU that may be sent to the physical layer in octets.

E.4.8.1.6 Originator.

The service user.

E.4.8.1.7 Errors/Exceptions.

NA.

E.4.8.2 Minimum Transmission Unit.

This attribute is read by the Service User to determine the minimum transmission unit length supported by the Service Provider.

E.4.8.2.1 Synopsis.

readonly attribute unsigned short MinTU.

E.4.8.2.2 Parameters.

unsigned short range  $2^{16}$ .

E.4.8.2.3 State.

This command valid in all states.

E.4.8.2.4 New State.

This command does cause a state change.

E.4.8.2.5 Response.

The size of the maximum TU that may be sent to the physical layer in octets.

E.4.8.2.6 Originator.

The service user.

E.4.8.2.7 Errors/Exceptions.

NA.

## **E.5 ALLOWABLE SEQUENCE OF SERVICE PRIMITIVES.**

There is no defined sequences of primitives.

## E.6 UTILIZATION OF PHYSICAL NON-REAL-TIME BUILDING BLOCKS

This table shows the usage of the services specified in these Physical non-real-time building blocks in the implementations of the 7 JTRS waveforms. It is not a requirement that an implementation follow the usage in this table.

Services	Parameters	SINCGARS	WDW NB/WB	HQ 1/2	LOS	HF ALE	DAMA/ DASA	VRC-99	Total Users
Antenna Control	setRxAntenna setTxAntenna	Yes	No	No	Yes	Yes	Yes	Yes	5
Transceiver Setup	setReceiverParameters setTransmitterParameters	No	No	Yes	No	Yes	Yes	Yes	4
Modulation Setup	setUpReceiverModulation setUpTransmitterModulation	Yes	No	Yes	Yes	Yes	Yes	Yes	6
Media Setup	setUpMediaType	Yes	No	Yes	Yes	No	Yes	Yes	5
RadioMode	setRadioMode	Yes	Yes	Yes	Yes	No	Yes	Yes	6
ReceiveTermination	dropCapture dropReceive	Yes	No	No	No	Yes	Yes	Yes	4
Transmit Inhibit	inhibitTransmit	No	No	No	No	No	Yes	Yes	2
PhysicalManagement	MaxTU MinTU	Yes	Yes	No	No	Yes	Yes	Yes	5

## **E.7 PRECEDENCE OF SERVICE PRIMITIVES.**

There is no precedence of primitives.

## **E.8 SERVICE USER GUIDELINES.**

This appendix shall summarize the guidelines for implementing a Service User that will be independent of the implementation of the Service Provider.

## **E.9 SERVICE PROVIDER-SPECIFIC INFORMATION.**

This appendix shall identify the information to be documented for each service provider implementation.

## **E.10 IDL.**

Since this building block contains parameterized classes IDL cannot be generated.

## **E.11 UML.**

This appendix, if provided, shall include the UML class and component diagrams for the Service Definition. The purpose for including these diagrams is to show the relationship between all the elements that make up the service.