Transceiver Facility SCA PSM specification

Document WINNF-TS-0008-App02

Version V2.1.1

22 January 2022
TERMS, CONDITIONS & NOTICES

This document has been prepared by the work group of WInnF project SCA-2015-001 “Transceiver Next” to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter “the Forum”). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the Transceiver Next work group.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter’s copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum’s participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Wireless Innovation Forum ™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.
# Table of Contents

TERMS, CONDITIONS & NOTICES .............................................................................................................. i
Table of Contents ........................................................................................................................................ ii
List of Figures ............................................................................................................................................... iv
List of Tables ............................................................................................................................................... iv
Contributors ................................................................................................................................................ v
Transceiver Facility SCA PSM specification .............................................................................................. 1
1 Introduction ............................................................................................................................................... 1
  1.1 Reference definitions ........................................................................................................................... 1
  1.2 Conformance ........................................................................................................................................ 2
    1.2.1 Radio platform items ..................................................................................................................... 2
    1.2.2 Radio application items .................................................................................................................. 2
  1.3 SCA transceivers implementations ..................................................................................................... 3
    1.3.1 Localized or distributed .................................................................................................................. 3
    1.3.2 Local or distant SCA connections .................................................................................................. 3
    1.3.3 Hybrid SCA transceiver ................................................................................................................. 4
  1.4 Document structure .............................................................................................................................. 4
2 SCA PSM management interfaces .......................................................................................................... 5
3 SCA PSM functional interfaces and ports ............................................................................................ 5
  3.1 Provide and Use Services Interfaces .................................................................................................. 5
    3.2 SCA PSM functional ports ................................................................................................................ 6
      3.2.1 Distribution between Tx and Rx channels ...................................................................................... 7
      3.2.2 Service-wise assignment .............................................................................................................. 8
      3.2.3 Services group-wise assignment .................................................................................................. 10
      3.2.4 Exceptions ..................................................................................................................................... 11
4 SCA PSM IDL files ................................................................................................................................... 12
  4.1 Common ............................................................................................................................................... 12
    4.1.1 ScaXcvrTypes.idl ........................................................................................................................... 12
    4.1.2 ScaXcvrExceptions.idl .................................................................................................................. 13
  4.2 Management services group .............................................................................................................. 15
    4.2.1 ScaXcvrManagementReset.idl ........................................................................................................ 15
    4.2.2 ScaXcvrManagementRadioSilence.idl ............................................................................................. 16
  4.3 BurstControl services group .............................................................................................................. 16
    4.3.1 ScaXcvrBurstControlTypes.idl ........................................................................................................ 16
    4.3.2 ScaXcvrBurstControlDirectCreation.idl ......................................................................................... 17
    4.3.3 ScaXcvrBurstControlRelativeCreation.idl ....................................................................................... 17
    4.3.4 ScaXcvrBurstControlAbsoluteCreation.idl ..................................................................................... 18
    4.3.5 ScaXcvrBurstControlStrobedCreation.idl ......................................................................................... 18
    4.3.6 ScaXcvrBurstControlTermination.idl .............................................................................................. 19
  4.4 BasebandSignal services group ........................................................................................................... 20
    4.4.1 ScaXcvrBasebandSignal16SamplesExchange.idl ........................................................................... 20
    4.4.2 ScaXcvrBasebandSignal32SamplesExchange.idl ............................................................................ 21
    4.4.3 ScaXcvrBasebandSignalFloatSamplesExchange.idl .................................................................... 21
    4.4.4 ScaXcvrBasebandSignalRxPacketsLengthControl.idl .................................................................. 22
  4.5 Tuning services group ....................................................................................................................... 23
4.5.1 ScaXcvrTuningTypes.idl ................................................................. 23
4.5.2 ScaXcvrTuningInitialTuning.idl ......................................................... 23
4.5.3 ScaXcvrTuningRetuning.idl ............................................................. 24
4.6 Notifications services group ............................................................. 25
4.6.1 ScaXcvrNotifications.idl ................................................................. 25
4.7 GainControl services group ............................................................... 26
4.7.1 ScaXcvrGainControlGainLocking.idl ................................................. 26
4.7.2 ScaXcvrGainChanges.idl ................................................................. 26
4.8 TimeAccess services group ............................................................... 27
4.8.1 ScaXcvrTransceiverTimeTimeAccess.idl ........................................... 27
4.9 ApplicationStrobe services group ................................................... 27
4.9.1 ScaXcvrStrobingApplicationStrobe.idl ............................................. 27

5 SCA PSM Properties .......................................................................... 29
5.1 Natures of SCA PSM properties ...................................................... 29
5.1.1 SCA capability properties .............................................................. 29
5.1.2 SCA capacity properties ............................................................... 29
5.1.3 SCA configure properties .............................................................. 29
5.2 Versions properties .......................................................................... 30
5.2.1 PIM version ............................................................................... 30
5.2.2 SCA version ............................................................................... 30
5.3 PIM-derived properties ................................................................... 30
5.3.1 Structure properties ................................................................ 30
5.3.2 Behavior properties ................................................................ 32
5.3.3 Notification properties ............................................................... 33
5.3.4 Interface declaration properties ................................................. 33
5.3.5 Initialization properties ............................................................. 33
5.3.6 Storage properties ................................................................ 34
5.3.7 Parameters validity properties .................................................. 34
5.3.8 WCET properties .................................................................... 35
5.3.9 Unspecified mapping ............................................................... 35

6 References ...................................................................................... 36
6.1 Referenced documents ................................................................ 36

END OF THE DOCUMENT ........................................................................... 37
List of Figures

Figure 1 Interfaces addressed by Transceiver Facility SCA PSM specification 1
Figure 2 Localized or distributed SCA transceivers 3
Figure 3 Concept of local or distant SCA connections 3
Figure 4 Concept of hybrid SCA transceiver 4
Figure 5 SCA functional ports distribution between Tx channels and Rx channels 7
Figure 6 SCA functional ports for Tx channels with service-wise assignment 8
Figure 7 SCA functional ports for Rx channels with service-wise assignment 9
Figure 8 SCA functional ports for Tx channels with services group-wise assignment 10
Figure 9 SCA functional ports for Rx channels with services group-wise assignment 11

List of Tables

Table 1 Definitions from Transceiver Facility PIM Specification 1
Table 2 Definitions from Principles for WInnForum Facility Standards 2
Table 3 Definitions from SCA section of WInnForum Facilities PSMs Mapping Rules 2
Table 4 Provide services functional services 5
Table 5 Use services functional interfaces 6
Table 6 XcavrPimVersion defined values 30
Table 7 XcavrScaVersion defined values 30
Table 8 SCA capability properties for simple structure properties 31
Table 9 SCA capability properties for TX_SERVICES structure property 31
Table 10 SCA capability properties for RX_SERVICES structure property 32
Table 11 SCA PSM properties for behavior properties 32
Table 12 SCA capability properties for simple notification properties 33
Table 13 SCA capability properties for EVENTS notification properties 33
Table 14 SCA capability properties for interface declaration properties 33
Table 15 SCA configure properties for initialization properties 34
Table 16 SCA configure properties for storage properties 34
Table 17 SCA capability properties for parameter validity properties 35
Contributors

The following individuals and their organization of affiliation are credited as Contributors to development of the specification, for having been involved in the work group that developed the draft then approved by WInnF member organizations:

- Eric Nicollet, Thales,
- Claude Bélisle, VIAVI Solutions,
- David Hagood, Cynosure,
- Hugues Latour, VIAVI Solutions,
- François Lévesque, VIAVI Solutions,
- Steve Bernier, VIAVI Solutions,
- Marc Adrat, Fraunhofer FKIE,
- Guillaume Delbarre, DGA,
- Olivier Kirsch, KEREVAL,
- Charles Linn, L3Harris,
- David Murotake, HiKE,
- Kevin Richardson, MITRE.
1 Introduction

This document WINNF-TS-0008-App02 is the SCA PSM specification of Transceiver Facility V2.1.0.

It derives from Transceiver Facility PIM Specification [Ref1] in accordance with Principles for WinnForum Facility Standards [Ref2].

It addresses the Software Communications Architecture (SCA) [Ref3] programming paradigm, applying the mapping rules of the SCA section of WinnForum Facilities PSMs Mapping Rules [Ref4] and specifically reporting any deviation to those rules.

The following figure positions of the interfaces addressed by the SCA PSM specification:

![Figure 1 Interfaces addressed by Transceiver Facility SCA PSM specification]

As depicted, the SCA PSM specification addresses the SCA PSM management interfaces and SCA PSM functional interfaces of transceivers, positioned, within an SCA node, between the SCA application components of SCA applications and SCA façades of SCA transceivers.

It addresses version 2.2.2 [Ref5] and version 4.1 [Ref6] of the SCA.

It uses the abbreviation “xvcr” to identify transceiver in formal identifiers.

1.1 Reference definitions

The Transceiver Facility SCA PSM specification applies the following definitions from Transceiver Facility PIM Specification [Ref1]:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Used definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transceiver technical concepts</td>
<td>transceiver, Transceiver Facility, Tx channel, Rx channel</td>
</tr>
<tr>
<td>Transceiver properties</td>
<td>structure property, behavior property, notification property, interface declaration property, initialization property, storage property, parameter validity property</td>
</tr>
</tbody>
</table>

Table 1 Definitions from Transceiver Facility PIM Specification
The Transceiver Facility SCA PSM specification applies the following definitions from Principles for WinInnForum Facility Standards [Ref2]:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Used definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base concepts</td>
<td>radio application</td>
</tr>
<tr>
<td>Architecture concepts</td>
<td>façade</td>
</tr>
<tr>
<td>WinInnForum facility</td>
<td>PIM specification</td>
</tr>
<tr>
<td>Services</td>
<td>service, service interface, provide service, use service, services group</td>
</tr>
<tr>
<td>Primitives</td>
<td>primitive, type, exception</td>
</tr>
<tr>
<td>Attributes</td>
<td>property</td>
</tr>
</tbody>
</table>

Table 2 Definitions from Principles for WinInnForum Facility Standards

The Transceiver Facility SCA PSM specification applies the following definitions from the SCA section of WinInnForum Facilities PSMs Mapping Rules [Ref4]:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Used definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification purpose</td>
<td>SCA PSM specification, SCA PSM functional interfaces, SCA PSM management interfaces</td>
</tr>
<tr>
<td>Software architecture</td>
<td>SCA node, SCA façade, SCA application, SCA application component</td>
</tr>
<tr>
<td>SCA ports</td>
<td>SCA functional port, service-wise assignment, services group-wise assignment</td>
</tr>
<tr>
<td>SCA properties</td>
<td>SCA PSM property</td>
</tr>
</tbody>
</table>

Table 3 Definitions from SCA section of WinInnForum Facilities PSMs Mapping Rules

The term “unspecified” indicates an aspect explicitly left to implementer’s decisions.

1.2 Conformance

1.2.1 Radio platform items

An SCA façade of a transceiver implementation is conformant with the Transceiver Facility SCA PSM specification if it provides an SCA implementation of related service interfaces.

An SCA transceiver is defined as a transceiver implementation with all of its SCA façades being conformant with the SCA PSM specification.

1.2.2 Radio application items

An SCA application component of a radio application is conformant with the Transceiver Facility SCA PSM specification if it can use SCA façades conformant with the SCA PSM specification, without using any non-standard service interface for the transceiver.
1.3 SCA transceivers implementations

1.3.1 Localized or distributed

Depending on its architecture, an SCA transceiver can contain one or several SCA façades, as depicted in the following figure:

![Figure 2: Localized or distributed SCA transceivers](image)

**Figure 2** Localized or distributed SCA transceivers

1.3.2 Local or distant SCA connections

The following figure illustrates the concept of local or distant connections to an SCA transceiver:

![Figure 3: Concept of local or distant SCA connections](image)

**Figure 3** Concept of local or distant SCA connections
### 1.3.3 Hybrid SCA transceiver

The following figure illustrates the concept of hybrid *SCA transceiver*:

![Concept of hybrid SCA transceiver](image)

Figure 4 Concept of hybrid SCA transceiver

### 1.4 Document structure

Section 2 specifies the normative content for the *SCA PSM management interfaces*.
Section 3 specifies the normative classes for the *SCA PSM functional interfaces*.
Section 4 specifies the standard IDL files for the *SCA PSM functional interfaces*.
Section 5 specifies the *SCA PSM properties*. 
2 SCA PSM management interfaces

The *SCA PSM management interfaces* specified in the SCA section of *WInnForum Facilities PSMs Mapping Rules* [Ref4] are applicable.

The property `XcvrScaVersion` (see section 5.2.2) indicates the used SCA version.

3 SCA PSM functional interfaces and ports

This normative section specifies the *SCA PSM functional interfaces* for *transceiver*, according to the SCA section of *WInnForum Facilities PSMs Mapping Rules* [Ref4].

3.1 Provide and Use Services Interfaces

The *service interfaces* for the *provide services* of the *SCA PSM functional interfaces* are specified by the following table:

<table>
<thead>
<tr>
<th>PIM service interface (in Transceiver::)</th>
<th>PIM section</th>
<th>SCA PSM service interface (in WInnF_Sca::Transceiver::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management::Reset</td>
<td>2.4.1.1</td>
<td>Management::Reset</td>
</tr>
<tr>
<td>Management::RadioSilence</td>
<td>2.4.1.2</td>
<td>Management::RadioSilence</td>
</tr>
<tr>
<td>BurstControl::DirectCreation</td>
<td>2.4.2.1</td>
<td>BurstControl::DirectCreation</td>
</tr>
<tr>
<td>BurstControl::RelativeCreation</td>
<td>2.4.2.2</td>
<td>BurstControl::RelativeCreation</td>
</tr>
<tr>
<td>BurstControl::AbsoluteCreation</td>
<td>2.4.2.3</td>
<td>BurstControl::AbsoluteCreation</td>
</tr>
<tr>
<td>BurstControl::StrobedCreation</td>
<td>2.4.2.4</td>
<td>BurstControl::StrobedCreation</td>
</tr>
<tr>
<td>BurstControl::Termination</td>
<td>2.4.2.5</td>
<td>BurstControl::Termination</td>
</tr>
<tr>
<td>BasebandSignal::SamplesTransmission</td>
<td>2.4.3.2</td>
<td>BasebandSignal16::SamplesExchange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BasebandSignal32::SamplesExchange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BasebandSignalFloat::SamplesExchange</td>
</tr>
<tr>
<td>Tuning::InitialTuning</td>
<td>2.4.4.1</td>
<td>Tuning::InitialTuning</td>
</tr>
<tr>
<td>Tuning::Retuning</td>
<td>2.4.4.2</td>
<td>Tuning::Retuning</td>
</tr>
<tr>
<td>GainControl::GainLocking</td>
<td>2.4.6.2</td>
<td>GainControl::GainLocking</td>
</tr>
<tr>
<td>TransceiverTime::TimeAccess</td>
<td>2.4.7.1</td>
<td>TransceiverTime::TimeAccess</td>
</tr>
<tr>
<td>Strobing::ApplicationStrobe</td>
<td>2.4.8.1</td>
<td>Strobing::ApplicationStrobe</td>
</tr>
</tbody>
</table>

Table 4 Provide services functional services
The service interfaces for the use services of the SCA PSM functional interfaces are specified by the following table:

<table>
<thead>
<tr>
<th>PIM service interface (in Transceiver::)</th>
<th>PIM section</th>
<th>SCA PSM service interface (in WinNF_Sca::Transceiver::)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasebandSignal::SamplesReception</td>
<td>2.4.3.1</td>
<td>BasebandSignal16::SamplesExchange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BasebandSignal32::SamplesExchange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BasebandSignalFloat::SamplesExchange</td>
</tr>
<tr>
<td>Notifications::Events</td>
<td>2.4.5.1</td>
<td>Notifications::Events</td>
</tr>
<tr>
<td>Notifications::Errors</td>
<td>2.4.5.2</td>
<td>Notifications::Errors</td>
</tr>
<tr>
<td>GainControl::GainChanges</td>
<td>2.4.6.1</td>
<td>GainControl::GainChanges</td>
</tr>
</tbody>
</table>

Table 5 Use services functional interfaces

The PIM BasebandSignal module maps to a distinct PSM module for each type option: BasebandSignal16 for 16-bit integer, BasebandSignal32 for 32-bit integer and BasebandSignalFloat for 32-bit floating point.

The PIM SamplesTransmission and SamplesReception service interfaces map to a common PSM service interface SamplesExchange.

3.2 SCA PSM functional ports

This normative section specifies the SCA functional ports for the two possible assignment strategies.
3.2.1 Distribution between Tx and Rx channels

The *SCA functional ports* are distributed between *Tx channels* and *Rx channels* as illustrated in the following figure:

![Figure 5 SCA functional ports distribution between Tx channels and Rx channels](image)

3.2.1.1 Tx channels ports

An *SCA transceiver* implements *SCA functional ports* for *Tx channels* if one or more are active (property `TX_CHANNEL > 0`).

The property `TX_SERVICES` specify which services are attached to *Tx channels*.

The number of SCA provide ports implementing the `BasebandSignal<type>::SamplesExchange` interface is at least equal to `TX_CHANNELS`.

All other *SCA functional ports* for *Tx channels* are at most implemented once.

3.2.1.2 Rx channels ports

An *SCA transceiver* implements *SCA functional ports* for *Rx channels* if one or more are active (property `RX_CHANNEL > 0`).

The property `RX_SERVICES` specify which services are attached to *Rx channels*.

The number of SCA use ports implementing the `BasebandSignal<type>::SamplesExchange` interface is at least equal to `RX_CHANNELS`.

All other *SCA functional ports* for *Rx channels* are at most implemented once.
3.2.2 Service-wise assignment

The following figure illustrates the SCA functional ports for Tx channels with service-wise assignment:

Figure 6 SCA functional ports for Tx channels with service-wise assignment
The following figure illustrates the *SCA functional ports* for *Rx channels* with service-wise assignment:

![Diagram of SCA functional ports for Rx channels with service-wise assignment](image)

**Figure 7** SCA functional ports for Rx channels with service-wise assignment
### 3.2.3 Services group-wise assignment

The following figure illustrates the *SCA functional ports* for *Tx channels* with services group-wise assignment:

![Diagram of SCA functional ports for Tx channels with services group-wise assignment](image)

**Figure 8** *SCA functional ports for Tx channels* with services group-wise assignment
The following figure illustrates the SCA functional ports for Rx channels with services group-wise assignment:

![SCA functional ports for Rx channels with services group-wise assignment](image)

Figure 9  SCA functional ports for Rx channels with services group-wise assignment

### 3.2.4 Exceptions

For the use service SamplesReception, the exceptions MaxPacketsLength and PacketsMILT, although present in the IDL signature, are not to be raised by the SCA application components.
4 SCA PSM IDL files

This normative section specifies the standard IDL files (.idl) used by conformant SCA transceiver and SCA application components (see section 1.2).

The IDL files are processed by an IDL compiler to generate source code in a specific programming language from which the implementation is developed.

The specified IDL files have been successfully compiled using:

- TAO™ v2.5.0,
- ORBexpress® RT C++ v3.0.3.

4.1 Common

4.1.1 ScaXcvrTypes.idl

The ScaXcvrTypes.idl file is specified as the IDL file that declares the types specified by section 3.4 of Transceiver Facility PIM Specification [Ref1].

IDL support of the options addressed by interface declaration properties ([Ref1], section 4.5) implies to use different IDL files, with their names distinguishing the selected options, in order to avoid multiple definitions in the same IDL interface.

The number of supported options has therefore been limited to the greatest extent, while preserving options enabling significant savings in execution resources.

This resulted in the following choices:

- For CARRIER_FREQ_TYPE and DELAY_TYPE, the SCA PSM IDL always uses 64-bit types, since the related types appear in control primitives, for which the execution resources saving is low when taking lower size data types,
- For IQ_TYPE, the SCA PSM IDL reflects the PIM types options (16-bit, 32-bit or 32-bit floating point) in specifying one IDL file for each possibility, because of the execution resources savings at stake,
- For TX_META_DATA and RX_META_DATA, the SCA PSM IDL always uses parameters of type CF::Properties as specified by the SCA ([Ref5] and [Ref6]):
  - If TX_META_DATA or RX_META_DATA is true, meta-data are actually supported and the nature of the meta-data remains unspecified,
  - In case TX_META_DATA or RX_META_DATA is false, meta-data are not supported, therefore zero-length CF::Properties are sent over the interface.

The content of ScaXcvrTypes.idl is specified as:

```idl
#ifndef __SCA_XCVR_TYPES_DEFINED
#define __SCA_XCVR_TYPES_DEFINED

#include "ScaXcvrExceptions.idl"

module WInnF_Sca
{
    module Transceiver
    {
```
4.1.2  ScaXcvrExceptions.idl

The ScaXcvrExceptions.idl file is specified as the IDL file that declares the exceptions specified by section 3.2 of the Transceiver Facility PIM Specification [Ref1].

The content of ScaTsfExceptions.idl is specified as:

```idl
#ifndef __SCA_XCVR_EXCEPTIONS_DEFINED
#define __SCA_XCVR_EXCEPTIONS_DEFINED

module WInnF_Sca
{
    module Transceiver
    {
        exception NoAlternateReferencing
        {
        }
        exception NoOngoingProcessing
        {
        }
        exception UnavailableStrobeSource
        {
        }
        exception MinBlockLength
        {
        }
        exception MaxBlockLength
        {
        }
    }
}
#endif // __SCA_XCVR_EXCEPTIONS_DEFINED
```
{
    exception MinCarrierFreq
    {
    }
    exception MaxCarrierFreq
    {
    }
    exception MinFromOngoing
    {
    }
    exception MaxFromOngoing
    {
    }
    exception MinFromPrevious
    {
    }
    exception MaxFromPrevious
    {
    }
    exception MinFromStrobe
    {
    }
    exception MaxFromStrobe
    {
    }
    exception MinGain
    {
    }
    exception MaxGain
    {
    }
    exception MaxNanoSeconds
    {
    }
    exception MaxRxPacketsLength
    {
    }
    exception MaxTuningPreset
    {
    }
    exception MaxPacketsLength
    {
    }
    exception AbsoluteMILT
    {
    }
exception RelativeMILT
{
}

exception RetuningMILT
{
}

exception TuningMILT
{
}

exception PacketsMILT
{
}

exception UnavailableService
{
}
};
#endif /* _SCA_XCVR_EXCEPTIONS_DEFINED */

4.2 Management services group

4.2.1 ScaXcvrManagementReset.idl

The `ScaXcvrManagementReset.idl` file is specified as the IDL file that declares the Management Reset service interface specified by section 3.1.1 of Transceiver Facility PIM Specification [Ref1].

The content of `ScaXcvrManagementReset.idl` is specified as:

```idl
#define __SCA_XCVR_MANAGEMENT_RESET_DEFINED
#include "ScaXcvrTypes.idl"
module WIInnF_Sca
{
    module Transceiver
    {
        module Management
        {
            interface Reset
            {
                void resetChannels() // PIM name is "reset()",
                // changed for compliance with TAO
                raises( UnavailableService );
            }
        }
    }
    //Transceiver
}; // WIInnF_Sca
#endif /* _SCA_XCVR_MANAGEMENT_RESET_DEFINED */
```
4.2.2  **ScaXcvrManagementRadioSilence.idl**

The *ScaXcvrManagementRadioSilence.idl* file is specified as the IDL file that declares the Management RadioSilence service interface specified by section 3.1.2 of *Transceiver Facility PIM Specification* [Ref1].

The content of *ScaXcvrManagementRadioSilence.idl* is specified as:

```
#ifndef SCA_XCVR_MANAGEMENT_RADIO_SILENCE_DEFINED
#define SCA_XCVR_MANAGEMENT_RADIO_SILENCE_DEFINED

#include "ScaXcvrTypes.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module Management
        {
            interface RadioSilence
            {
                void startRadioSilence()
                    raises( UnavailableService );

                void stopRadioSilence()
                    raises( UnavailableService );
            }
        }
    }
}
#endif // SCA_XCVR_MANAGEMENT_RADIO_SILENCE_DEFINED
```

4.3  **BurstControl services group**

4.3.1  **ScaXcvrBurstControlTypes.idl**

The *ScaXcvrBurstControlTypes.idl* file is specified as the IDL file that declares some of the API types used by the BurstControl services that are specified by section 3.4 of *Transceiver Facility PIM Specification* [Ref1] along with the exceptions related to the use of these types in the BurstControl services.

The content of *ScaXcvrBurstControlTypes.idl* is specified as:

```
#ifndef SCA_XCVR_BURST_CONTROL_TYPES_DEFINED
#define SCA_XCVR_BURST_CONTROL_TYPES_DEFINED

#include "ScaXcvrTypes.idl"

module WInnF_Sca
{
    module Transceiver
    {
        // BlockLength
typedef unsigned long BlockLength;
        const BlockLength UNDEFINED_BLOCK_LENGTH = 0xFFFFFFFF;
    }
}
#endif // SCA_XCVR_BURST_CONTROL_TYPES_DEFINED
```
4.3.2 ScaXcvrBurstControlDirectCreation.idl

The ScaXcvrBurstControlDirectCreation.idl file is specified as the IDL file that declares the BurstControl DirectCreation service interface specified by section 3.1.3 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlDirectCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_DIRECT_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_DIRECT_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module BurstControl
        {
            interface DirectCreation
            {
                void startBurst(in BlockLength requestedLength)
                raises(
                    UnavailableService,
                    MinBlockLength,
                    MaxBlockLength);
            };
        };
    };
}; // Transceiver
}; // WInnF_Sca
#endif // __SCA_XCVR_BURST_CONTROL_DIRECT_CREATION_DEFINED
```

4.3.3 ScaXcvrBurstControlRelativeCreation.idl

The ScaXcvrBurstControlRelativeCreation.idl file is specified as the IDL file that declares the BurstControl RelativeCreation service interface specified by section 3.1.4 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlRelativeCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_RELATIVE_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_RELATIVE_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module BurstControl
        {
            interface RelativeCreation
            {
                void scheduleRelativeBurst(in boolean requestedAlternate,
                                            in Delay requestedDelay,
                                            in BlockLength requestedLength)
                raises(
                    UnavailableService,
                    MinBlockLength,
                    MaxBlockLength);
            };
        };
    };
}; // Transceiver
}; // WInnF_Sca
```

4.3.2 ScaXcvrBurstControlDirectCreation.idl

The ScaXcvrBurstControlDirectCreation.idl file is specified as the IDL file that declares the BurstControl DirectCreation service interface specified by section 3.1.3 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlDirectCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_DIRECT_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_DIRECT_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module BurstControl
        {
            interface DirectCreation
            {
                void startBurst(in BlockLength requestedLength)
                raises(
                    UnavailableService,
                    MinBlockLength,
                    MaxBlockLength);
            };
        };
    };
}; // Transceiver
}; // WInnF_Sca
```

4.3.3 ScaXcvrBurstControlRelativeCreation.idl

The ScaXcvrBurstControlRelativeCreation.idl file is specified as the IDL file that declares the BurstControl RelativeCreation service interface specified by section 3.1.4 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlRelativeCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_RELATIVE_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_RELATIVE_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module BurstControl
        {
            interface RelativeCreation
            {
                void scheduleRelativeBurst(in boolean requestedAlternate,
                                            in Delay requestedDelay,
                                            in BlockLength requestedLength)
                raises(
                    UnavailableService,
                    MinBlockLength,
                    MaxBlockLength);
            };
        };
    };
}; // Transceiver
}; // WInnF_Sca
```
in BlockLength requestedLength)
  raises( UnavailableService,
      NoAlternateReferencing,
      MinFromPrevious,
      MaxFromPrevious,
      MinBlockLength,
      MaxBlockLength,
      RelativeMILT);
};
}; //Transceiver
}; // WINnF_Sca
#endif // __SCA_XCVR_BURST_CONTROL_RELATIVE_CREATION_DEFINED

4.3.4 ScaXcvrBurstControlAbsoluteCreation.idl

The ScaXcvrBurstControlAbsoluteCreation.idl file is specified as the IDL file that declares the BurstControl AbsoluteCreation service interface specified by section 3.1.5 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlAbsoluteCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_ABSOLUTE_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_ABSOLUTE_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"

module WINnF_Sca
{
  module Transceiver
  {
    module BurstControl
    {
      interface AbsoluteCreation
      {
        void scheduleAbsoluteBurst(in TimeSpec requestedStartTime,
                                     in BlockLength requestedLength)
          raises( UnavailableService,
                  MaxNanoSeconds,
                  MinBlockLength,
                  MaxBlockLength,
                  AbsoluteMILT);
      };
    };
  };
}; //WINnF_Sca
#endif // __SCA_XCVR_BURST_CONTROL_ABSOLUTE_CREATION_DEFINED
```

4.3.5 ScaXcvrBurstControlStrobedCreation.idl

The ScaXcvrBurstControlStrobedCreation.idl file is specified as the IDL file that declares the BurstControl StrobedCreation service interface specified by section 3.1.6 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBurstControlStrobedCreation.idl is specified as:

```idl
#ifndef __SCA_XCVR_BURST_CONTROL_STROBED_CREATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_STROBED_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"

module WINnF_Sca
{
  module Transceiver
  {
    module BurstControl
    {
      interface StrobedCreation
      {
        void scheduleStrobedBurst(in TimeSpec requestedStartTime,
                                   in BlockLength requestedLength)
          raises( UnavailableService,
                  MaxNanoSeconds,
                  MinBlockLength,
                  MaxBlockLength,
                  StrobedMILT);
      };
    };
  };
}; //WINnF_Sca
#endif // __SCA_XCVR_BURST_CONTROL_STROBED_CREATION_DEFINED
```
```c
#define __SCA_XCVR_BURST_CONTROL_STROBED_CREATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInF_Sca
{
    module Transceiver
    {
        // Strobe
        typedef unsigned short StrobeSource;
        const StrobeSource APPLICATION_STROBE = 0;
        const StrobeSource TIME_REF_PPS = 1;
        const StrobeSource GNSS_PPS = 2;
        // Values ranging from 3 to 7 are reserved for User Defined Strobe.

    module BurstControl
    {
        interface StrobedCreation
        {
            void scheduleStrobedBurst(in StrobeSource requestedStrobeSource,
                                        in Delay requestedDelay,
                                        in BlockLength requestedLength)
            raises( UnavailableService,
                    UnavailableStrobeSource,
                    MinFromStrobe,
                    MaxFromStrobe,
                    MinBlockLength,
                    MaxBlockLength);
        }
    } // Transceiver
}; // WInF_Sca
#endif // __SCA_XCVR_BURST_CONTROL_STROBED_CREATION_DEFINED
```

4.3.6 ScaXcvrBurstControlTermination.idl

The `ScaXcvrBurstControlTermination.idl` file is specified as the IDL file that declares the BurstControl Termination service interface specified by section 3.1.7 of Transceiver Facility PIM Specification [Ref1].

The content of `ScaXcvrBurstControlTermination.idl` is specified as:

```c
#ifndef __SCA_XCVR_BURST_CONTROL_TERMINATION_DEFINED
#define __SCA_XCVR_BURST_CONTROL_TERMINATION_DEFINED
#include "ScaXcvrBurstControlTypes.idl"
module WInF_Sca
{
    module Transceiver
    {
        module BurstControl
        {
            interface Termination
            {
                void setBlockLength(in BlockLength requestedLength)
                raises( UnavailableService,
                        NoOngoingProcessing,
                        MinBlockLength,
                        MaxBlockLength);
            }
        }
    } // Transceiver
}; // WInF_Sca
```

```c
#endif // __SCA_XCVR_BURST_CONTROL_TERMINATION_DEFINED
```
MaxBlockLength);

void stopBurst()
    raises(UnavailableService,
          NoOngoingProcessing);
};
}; //Transceiver
}; // WInnF_Sca

#endif // __SCA_XCVR_BURST_CONTROL_TERMINATION_DEFINED

4.4 BasebandSignal services group

4.4.1 ScaXcvrBasebandSignal16SamplesExchange.idl

The ScaXcvrBasebandSignal16SamplesExchange.idl file is specified as the IDL file that declares the BasebandSignal16 SamplesExchange to address service interfaces specified by section 3.1.8, and 3.1.9 of Transceiver Facility PIM Specification [Ref1], when IQ_TYPE is equal to 16bit.

The content of ScaXcvrBasebandSignal16SamplesExchange.idl is specified as:

```idl
#ifndef __SCA_XCVR_BASEBAND_SIGNAL_16_SAMPLES_EXCHANGE_DEFINED
#define __SCA_XCVR_BASEBAND_SIGNAL_16_SAMPLES_EXCHANGE_DEFINED

#include "ScaXcvrTypes.idl"
#include "CFProperties.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module BasebandSignal16
        {
            typedef short IQ;
            struct BasebandSample
            {
                IQ valueI;
                IQ valueQ;
            };
            typedef sequence <BasebandSample> BasebandPacket;

            interface SamplesExchange
            {
                void pushPacket(in BasebandPacket packet,
                                in boolean endOfBlock,
                                in CF::Properties metaData)
                    raises( UnavailableService,
                            MaxPacketsLength,
                            PacketsMILT);
                
            };
        } // BasebandSignal16
    } //Transceiver
} // WInnF_Sca

#endif // __SCA_XCVR_BASEBAND_SIGNAL_16_SAMPLES_EXCHANGE_DEFINED
```
4.4.2 **ScaXcvrBasebandSignal32SamplesExchange.idl**

The `ScaXcvrBasebandSignal32SamplesExchange.idl` file is specified as the IDL file that declares the `BasebandSignal32 SamplesExchange` to address the service interfaces specified by section 3.1.8, and 3.1.9 of Transceiver Facility PIM Specification [Ref1], when `IQ_TYPE` is equal to 32bit.

The content of `ScaXcvrBasebandSignal32SamplesExchange.idl` is specified as:

```idl
#ifndef __SCA_XCVR_BASEBAND_SIGNAL_32_SAMPLES_EXCHANGE_DEFINED
#define __SCA_XCVR_BASEBAND_SIGNAL_32_SAMPLES_EXCHANGE_DEFINED

#include "ScaXcvrTypes.idl"
#include "CFProperties.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module BasebandSignal32
        {
            typedef long IQ;
            struct BasebandSample
            {
                IQ valueI;
                IQ valueQ;
            };
            typedef sequence <BasebandSample> BasebandPacket;

            interface SamplesExchange
            {
                void pushPacket(in BasebandPacket packet,
                                in boolean endOfBlock,
                                in CF::Properties metaData)
                    raises(UnavailableService,
                            MaxPacketsLength,
                            PacketsMILT);
            };
        } // BasebandSignal32
    } // Transceiver
} // WInnF_Sca
#endif // __SCA_XCVR_BASEBAND_SIGNAL_32_SAMPLES_EXCHANGE_DEFINED
```

4.4.3 **ScaXcvrBasebandSignalFloatSamplesExchange.idl**

The `ScaXcvrBasebandSignalFloatSamplesExchange.idl` file is specified as the IDL file that declares the `BasebandSignalFloat SampleExchange` to address the service interfaces specified by section 3.1.8, and 3.1.9 of Transceiver Facility PIM Specification [Ref1], when `IQ_TYPE` is equal to floatingPoint.

The content of `ScaXcvrBasebandSignalFloatSamplesExchange.idl` is specified as:

```idl
#ifndef __SCA_XCVR_BASEBAND_SIGNAL_FLOAT_SAMPLES_EXCHANGE_DEFINED
#define __SCA_XCVR_BASEBAND_SIGNAL_FLOAT_SAMPLES_EXCHANGE_DEFINED
```
#include "ScaXcvrTypes.idl"
#include "CFProperties.idl"

module WINNF_Sca
{
    module Transceiver
    {
        module BasebandSignalFloat
        {
            typedef float IQ;
            struct BasebandSample
            {
                IQ valueI;
                IQ valueQ;
            };
            typedef sequence <BasebandSample> BasebandPacket;
            interface SamplesExchange
            {
                void pushPacket(in BasebandPacket packet,
                                in boolean endOfBlock,
                                in CF::Properties metaData)
                                raises( UnavailableService,
                                       MaxPacketsLength,
                                       PacketsMILT);
            };
        } //BasebandSignalFloat
    } //Transceiver
} //WINNF_Sca
#endif // __SCA_XCVR_BASEBAND_SIGNAL_FLOAT_SAMPLES_EXCHANGE_DEFINED

4.4.4 ScaXcvrBasebandSignalRxPacketsLengthControl.idl

The ScaXcvrBasebandSignalRxPacketsLengthControl.idl file is specified as the IDL file that declares the RxPacketsLengthControl service interface specified by section 3.1.10 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrBasebandSignalRxPacketsLengthControl.idl is specified as:

#ifndef __SCA_XCVR_BASEBAND_SIGNAL_RX_PACKETS_LENGTH_CONTROL_DEFINED
#define __SCA_XCVR_BASEBAND_SIGNAL_RX_PACKETS_LENGTH_CONTROL_DEFINED

#include "ScaXcvrTypes.idl"
#include "CFProperties.idl"

module WINNF_Sca
{
    module Transceiver
    {
        interface RxPacketsLengthControl
        {
            void setRxPacketsLength(in PacketLength requestedLength)
                                raises( UnavailableService,
                                       PacketsMILT,
                                       MaxPacketsLength);
        };
    } //Transceiver
} //WINNF_Sca
#endif // __SCA_XCVR_BASEBAND_SIGNAL_RX_PACKETS_LENGTH_CONTROL_DEFINED
4.5 Tuning services group

4.5.1 ScaXcvrTuningTypes.idl

The ScaXcvrTuningTypes.idl file is specified as the IDL file that declares some of the API types used by the Tuning services that are specified by section 3.4 of Transceiver Facility PIM Specification [Ref1] along with the exceptions related to the use of these types in the Tuning services.

The content of ScaXcvrTuningTypes.idl is specified as:

```idl
#ifndef __SCA_XCVR_TUNING_TYPES_DEFINED
#define __SCA_XCVR_TUNING_TYPES_DEFINED
#include "ScaXcvrTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        //CARRIER_FREQ_TYPE is define as and IDL typedef using the typedef convention
        typedef unsigned long long CarrierFreqType;
        typedef CarrierFreqType CarrierFreq; // in Hz
        const CarrierFreq UNDEFINED_CARRIER_FREQ = 0x7FFFFFFFFFFFFFFF; // 9223372036854775807
    }; //Transceiver
}; // WInnF_Sca
#endif // __SCA_XCVR_TUNING_TYPES_DEFINED
```

4.5.2 ScaXcvrTuningInitialTuning.idl

The ScaXcvrTuningInitialTuning.idl file is specified as the IDL file that declares the InitialTuning service interface specified by sections 3.1.11 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrTuningInitialTuning.idl is specified as:

```idl
#ifndef __SCA_XCVR_TUNING_INITIAL_TUNING_DEFINED
#define __SCA_XCVR_TUNING_INITIAL_TUNING_DEFINED
#include "ScaXcvrTuningTypes.idl"
module WInnF_Sca
{
    module Transceiver
```

All Rights Reserved
4.5.3 ScaXcvrTuningRetuning.idl

The ScaXcvrTuningRetuning.idl file is specified as the IDL file that declares the Retuning service interface specified by sections 3.1.12 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrTuningRetuning.idl is specified as:

```idl
#ifndef __SCA_XCVR_TUNING_RETUNING_DEFINED
#define __SCA_XCVR_TUNING_RETUNING_DEFINED
#include "ScaXcvrTuningTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module Tuning
        {
            interface Retuning
            {
                void retune(in CarrierFreq requestedFrequency, 
                in Gain requestedGain, 
                in Delay requestedDelay) 
                raises( UnavailableService, 
                    NoOngoingProcessing, 
                    MinCarrierFreq, 
                    MaxCarrierFreq, 
                    MinGain, 
                    MaxGain, 
                    MinFromOngoing, 
                    MaxFromOngoing, 
                    RetuningMILT); 
            }
        }
    }
}
#endif // __SCA_XCVR_TUNING_RETUNING_DEFINED
```
4.6 Notifications services group

4.6.1 ScaXcvrNotifications.idl

The ScaXcvrNotifications.idl file is specified as the IDL file that declares the Notifications service interfaces specified by sections 3.1.13 and 3.1.14 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrNotifications.idl is specified as:

```idl
#ifndef __SCA_XCVR_NOTIFICATIONS_DEFINED
#define __SCA_XCVR_NOTIFICATIONS_DEFINED

module WInnF_Sca
{
    module Transceiver
    {
        module Notifications
        {
            enum Event
            {
                PROCESSING_START_EVENT,
                PROCESSING_STOP_EVENT,
                SILENCE_START_EVENT,
                SILENCE_STOP_EVENT
            }
            enum Error
            {
                DELAYED_TUNING_ERROR,
                TUNING_TIMEOUT_ERROR,
                DELAYED_FIRST_SAMPLE_ERROR,
                FIRST_SAMPLE_TIMEOUT_ERROR,
                TRANSMISSION_UNDERFLOW_ERROR,
                RECEIPTION_OVERFLOW_ERROR,
                SHORTER_TRANSMITTED_BLOCK_ERROR,
                LONGER_TRANSMITTED_BLOCK_ERROR
            }
            interface Events
            {
                void notifyEvent(in Event notifiedEvent);
            }
            interface Errors
            {
                void notifyError(in Error notifiedError);
            }
        }
    }
}; //Transceiver
}; // WInnF_Sca
#endif // __SCA_XCVR_NOTIFICATIONS_DEFINED
```
4.7 GainControl services group

4.7.1 ScaXcvrGainControlGainLocking.idl

The ScaXcvrGainControlGainLocking.idl file is specified as the IDL file that declares the GainControl GainLocking service interface specified by section 3.1.16 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrGainControlGainLocking.idl is specified as:

```idl
#ifndef __SCA_XCVR_GAIN_CONTROL_GAIN_LOCKING_DEFINED
#define __SCA_XCVR_GAIN_CONTROL_GAIN_LOCKING_DEFINED

#include "ScaXcvrTypes.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module GainControl
        {
            interface GainLocking
            {
                void lockGain()
                    raises( UnavailableService,
                            NoOngoingProcessing);

                void unlockGain()
                    raises( UnavailableService,
                            NoOngoingProcessing);
            };
        };
    };

    //Transceiver
}; // WInnF_Sca

#endif // __SCA_XCVR_GAIN_CONTROL_GAIN_LOCKING_DEFINED
```

4.7.2 ScaXcvrGainChanges.idl

The ScaXcvrGainControlGainChanges.idl file is specified as the IDL file that declares the GainControl GainChanges service interface specified by section 3.1.15 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrGainControlGainChanges.idl is specified as:

```idl
#ifndef __SCA_XCVR_GAIN_CONTROL_GAIN_CHANGES_DEFINED
#define __SCA_XCVR_GAIN_CONTROL_GAIN_CHANGES_DEFINED

#include "ScaXcvrTypes.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module GainControl
        {
            typedef unsigned long SampleNumber;

            interface GainChanges
```
4.8 TimeAccess services group

4.8.1 ScaXcvrTransceiverTimeTimeAccess.idl

The ScaXcvrTransceiverTimeTimeAccess.idl file is specified as the IDL file that declares the TransceiverTime TimeAccess service interface specified by section 3.1.17 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrTransceiverTimeTimeAccess.idl is specified as:

```idl
#ifndef __SCA_XCVR_TRANSCEIVER_TIME_TIME_ACCESS_DEFINED
#define __SCA_XCVR_TRANSCEIVER_TIME_TIME_ACCESS_DEFINED
#include "ScaXcvrTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module TransceiverTime
        {
            interface TimeAccess
            {
                void getCurrentTime(out TimeSpec currentTime)
                    raises( UnavailableService );
                void getLastStartTime(out TimeSpec lastStartTime,
                                         out BurstNumber lastBurstNumber)
                    raises( UnavailableService );
            }
        }
    }
}; // Transceiver
}; // WInnF_Sca
#endif // __SCA_XCVR_TRANSCEIVER_TIME_TIME_ACCESS_DEFINED
```

4.9 ApplicationStrobe services group

4.9.1 ScaXcvrStrobingApplicationStrobe.idl

The ScaXcvrStrobingApplicationStrobe.idl file is specified as the IDL file that declares the Strobing ApplicationStrobe service interface specified by section 3.1.18 of Transceiver Facility PIM Specification [Ref1].

The content of ScaXcvrStrobingApplicationStrobe.idl is specified as:

```idl
#ifndef __SCA_XCVR_STROBING_APPLICATION_STROBE_DEFINED
#define __SCA_XCVR_STROBING_APPLICATION_STROBE_DEFINED
#include "ScaXcvrTypes.idl"
module WInnF_Sca
{
    module Transceiver
    {
        module TransceiverTime
        {
            interface StrobeControl
            {
                void indicateGain(in Gain newGain,
                                   in SampleNumber firstValidSample);
            }
        }
    }
}; // Transceiver
}; // WInnF_Sca
#endif // __SCA_XCVR_STROBING_APPLICATION_STROBE_DEFINED
```
#define __SCA_XCVR_STROBING_APPLICATION_STROBE_DEFINED

#include "ScaXcvrTypes.idl"

module WInnF_Sca
{
    module Transceiver
    {
        module Strobing
        {
            interface ApplicationStrobe
            {
                void triggerStrobe()
                raises( UnavailableService );
            }
        }
    } // Transceiver
}; // WInnF_Sca

#ifdef __SCA_XCVR_STROBING_APPLICATION_STROBE_DEFINED
5  SCA PSM Properties

This normative section specifies the SCA PSM properties.

5.1  Natures of SCA PSM properties

Three different natures of SCA PSM properties are considered:

- SCA capability properties,
- SCA capacity properties,
- SCA configure properties.

5.1.1  SCA capability properties

An SCA capability property is defined as an SCA property of kind “allocation” with an action type other than “external”.

SCA capability properties enable to check that an SCA transceiver can support the needs of a radio application.

They are specified by the domain profiles (.prf) of both the radio application and the radio platform:

- The .prf of the radio application specifies the desired value,
- The .prf of the SCA transceiver specifies the list of possible values,
- For instantiation of a radio application, the Core Framework checks if the desired value belongs to the list of possible values. If not, the instantiation aborts.

5.1.2  SCA capacity properties

An SCA capacity property is defined as an SCA property of kind “allocation” with an action type equal to “external”.

SCA capacity properties enable to manage consumption of resources by the instantiated radio applications:

- The .prf of the radio application specifies the desired amount,
- The .prf of the SCA transceiver indicates availability of the property and the total available amount,
- For instantiation of a radio application, if the property is available, the Core Framework reserves the amount desired by the radio application using allocateCapacity(). If the allocation fails, the instantiation aborts.

5.1.3  SCA configure properties

An SCA configure property is defined as an SCA property of kind “configure”.

SCA configure properties enable to configure an SCA transceiver according to the needs of a radio application, before the CONFIGURED state is reached.
The Core Framework sets the SCA configure properties to the desired value using `configure()`.

### 5.2 Versions properties

#### 5.2.1 PIM version

The `XcvrPimVersion` SCA PSM property is specified as an integer indicating the version of the Transceiver Facility PIM Specification [Ref1] from which the SCA PSM specification is derived.

The defined values of `XcvrPimVersion` are specified as:

<table>
<thead>
<tr>
<th>PIM specification version</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2.0.0</td>
<td>0x020000</td>
</tr>
<tr>
<td>V2.1.0</td>
<td>0x020100</td>
</tr>
</tbody>
</table>

Table 6 XcvrPimVersion defined values

`XcvrPimVersion` is equal to `0x020100`.

#### 5.2.2 SCA version

The `XcvrScaVersion` SCA PSM property is specified as an integer indicating the used SCA version.

The defined values for `XcvrScaVersion` are specified as:

<table>
<thead>
<tr>
<th>SCA version</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2.2.2</td>
<td>0x020202</td>
</tr>
<tr>
<td>V4.1</td>
<td>0x040100</td>
</tr>
</tbody>
</table>

Table 7 XcvrScaVersion defined values

### 5.3 PIM-derived properties

The PIM-derived SCA PSM properties are intended to support instantiation of SCA applications, and may be used for other purposes.

#### 5.3.1 Structure properties

The structure properties, specified in section 4.2 of Transceiver Facility PIM Specification [Ref1], specify aspects related to the structure of a transceiver implementation.

Structure properties map to SCA capability properties, with same consistency conditions as in the PIM specification.
The PIM simple *structure properties* map to the following *SCA capability properties*:

<table>
<thead>
<tr>
<th>PIM structure property</th>
<th>SCA capability property</th>
<th>SCA type (Reserved values)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX_CHANNELS</td>
<td>txChannels</td>
<td>unsigned short</td>
</tr>
<tr>
<td>RX_CHANNELS</td>
<td>rxChannels</td>
<td>unsigned short</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>duplex</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: fullDuplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: halfDuplex</td>
</tr>
<tr>
<td>TX_SHAPING</td>
<td>txShaping</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: nominal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: specific</td>
</tr>
<tr>
<td>TIME_COUPLING</td>
<td>timeCoupling</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: autonomous</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: coupled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: coupledToTerminalTime</td>
</tr>
</tbody>
</table>

Table 8  *SCA capability properties for simple structure properties*

The PIM *structure property* TX_SERVICES maps to the following *SCA capability properties*:

<table>
<thead>
<tr>
<th><em>SCA capability property</em></th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasTxServicesReset</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesRadioSilence</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesDirectCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesRelativeCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesAbsoluteCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesStrobedCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesTermination</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesInitialTuning</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesInitialRetuning</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesGainLocking</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesTimeAccess</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesApplicationStrobe</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesEvents</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxServicesErrors</td>
<td>boolean</td>
</tr>
<tr>
<td>hasTxGainChanges</td>
<td>boolean</td>
</tr>
</tbody>
</table>

Table 9  *SCA capability properties for TX_SERVICES structure property*
The PIM structure property **RX_SERVICES** maps to the following SCA capability properties:

<table>
<thead>
<tr>
<th>SCA capability property</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasRxServicesReset</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesRadioSilence</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesDirectCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesRelativeCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesAbsoluteCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesStrobedCreation</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesTermination</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesInitialTuning</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesInitialRetuning</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesGainLocking</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesTimeAccess</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesApplicationStrobe</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesEvents</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxServicesErrors</td>
<td>boolean</td>
</tr>
<tr>
<td>hasRxGainChanges</td>
<td>boolean</td>
</tr>
</tbody>
</table>

Table 10 SCA capability properties for **RX_SERVICES** structure property

### 5.3.2 Behavior properties

The behavior properties, specified in section 4.3 of Transceiver Facility PIM Specification [Ref1], specify aspects related to the behavior of a transceiver instance.

For behavior properties whose nature is specified as SCA configure properties, a transceiver implementation may only support a limited set out of the specified reserved values. A call to `configure()` with a non-supported reserved value generates an error.

The behavior properties map to the following SCA PSM properties:

<table>
<thead>
<tr>
<th>PIM behavior property</th>
<th>SCA property nature</th>
<th>SCA PSM property</th>
<th>SCA type (Reserved values)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TUNING_ASSOCIATION</td>
<td>Configure</td>
<td>tuningAssociation</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: sequential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2: burstReferencing</td>
</tr>
<tr>
<td>AGC</td>
<td>Configure</td>
<td>Agc</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: noAGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2: earlyControl</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3: permanentControl</td>
</tr>
<tr>
<td>ALC</td>
<td>Configure</td>
<td>Alc</td>
<td>long</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: noALC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2: activeALC</td>
</tr>
<tr>
<td>TUNING_TIMEOUT</td>
<td>Capability</td>
<td>tuningTimeout</td>
<td>unsigned long</td>
</tr>
<tr>
<td>1ST_SAMPLE_TIMEOUT</td>
<td>Capability</td>
<td>1stSampleTimeout</td>
<td>unsigned long</td>
</tr>
</tbody>
</table>

Table 11 SCA PSM properties for behavior properties
5.3.3 Notification properties

The notification properties, specified in section 4.4 of Transceiver Facility PIM Specification [Ref1], specify aspects relative to notifications made by a transceiver instance to the radio application.

The simple notification properties map to the following SCA capability properties:

<table>
<thead>
<tr>
<th>PIM notification property</th>
<th>SCA capability property</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCEPTIONS_SUPPORT</td>
<td>exceptionsSupport</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

Table 12 SCA capability properties for simple notification properties

The PIM notification property EVENTS maps to the following SCA capability properties:

<table>
<thead>
<tr>
<th>SCA capability property</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventProcessingStart</td>
<td>Boolean</td>
</tr>
<tr>
<td>eventProcessingStop</td>
<td>Boolean</td>
</tr>
<tr>
<td>eventSilenceStart</td>
<td>Boolean</td>
</tr>
<tr>
<td>eventSilenceStop</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

Table 13 SCA capability properties for EVENTS notification properties

The mapping of the other notification properties (EXCEPTIONS and ERRORS) is unspecified.

5.3.4 Interface declaration properties

The interface declaration properties, specified in section 4.5 of Transceiver Facility PIM Specification [Ref1], specify aspects relative to declarations of service interfaces.

The interface declaration properties map to specific types declaration constructs.

In addition, interface declaration properties map to the following SCA capability properties:

<table>
<thead>
<tr>
<th>PIM interface declaration property</th>
<th>SCA capability property</th>
<th>SCA type (Reserved values)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX_META_DATA</td>
<td>txMetaData</td>
<td>Boolean</td>
</tr>
<tr>
<td>RX_META_DATA</td>
<td>rxMetaData</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

Table 14 SCA capability properties for interface declaration properties

5.3.5 Initialization properties

The initialization properties, specified in section 4.6 of Transceiver Facility PIM Specification [Ref1], specify the conditions to be met by a transceiver implementation when the CONFIGURED state is reached by its Tx channels and Rx channels.
The initialization properties map to the following SCA configure properties:

<table>
<thead>
<tr>
<th>PIM initialization property</th>
<th>SCA configure property</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT_TX_PACKETS_LENGTH</td>
<td>initTxPacketsLength</td>
<td>unsigned long</td>
</tr>
<tr>
<td>INIT_CARRIER_FREQ</td>
<td>initCarrierFreq</td>
<td>unsigned long long</td>
</tr>
<tr>
<td>INIT_GAIN</td>
<td>initGain</td>
<td>short</td>
</tr>
</tbody>
</table>

Table 15 SCA configure properties for initialization properties

5.3.6 Storage properties

The storage properties, specified in section 4.9 of Transceiver Facility PIM Specification [Ref1], specify the number of calls to certain operations a transceiver instance can store before blocking further calls until storage becomes available again.

A call to configure() with a requested value outside the range of what the implementation supports generates an error.

The storage properties map to the following SCA configure properties:

<table>
<thead>
<tr>
<th>PIM storage property</th>
<th>SCA configure property</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATION_STORAGE</td>
<td>creationStorage</td>
<td>unsigned short</td>
</tr>
<tr>
<td>TUNING_STORAGE</td>
<td>tuningStorage</td>
<td>unsigned short</td>
</tr>
<tr>
<td>TX_BASEBAND_STORAGE</td>
<td>txBasebandStorage</td>
<td>unsigned long</td>
</tr>
</tbody>
</table>

Table 16 SCA configure properties for storage properties

5.3.7 Parameters validity properties

The parameter validity properties, specified in section 4.7 of Transceiver Facility PIM Specification [Ref1], specify the validity conditions applicable to parameters in service interfaces primitives.
The parameter validity properties map to the following SCA capacity properties as follows:

<table>
<thead>
<tr>
<th>PIM parameter validity property</th>
<th>SCA capability properties</th>
<th>SCA type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN_BLOCK_LENGTH</td>
<td>minBlockLength</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_BLOCK_LENGTH</td>
<td>maxBlockLength</td>
<td>unsigned long</td>
</tr>
<tr>
<td>ALTERNATEREFERENCING</td>
<td>alternateReferencing</td>
<td>Boolean</td>
</tr>
<tr>
<td>MIN_FROM_PREVIOUS</td>
<td>minFromPrevious</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_FROM_PREVIOUS</td>
<td>maxFromPrevious</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MIN_FROM_STROBE</td>
<td>minFromStrobe</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_FROM_STROBE</td>
<td>maxFromStrobe</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_PACKETS_LENGTH</td>
<td>maxPacketsLength</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_TUNING_PRESET</td>
<td>maxTuningPreset</td>
<td>unsigned short</td>
</tr>
<tr>
<td>MIN_CARRIER_FREQ</td>
<td>minCarrierFreq</td>
<td>unsigned long long</td>
</tr>
<tr>
<td>MAX_CARRIER_FREQ</td>
<td>maxCarrierFreq</td>
<td>unsigned long long</td>
</tr>
<tr>
<td>MIN_GAIN</td>
<td>minGain</td>
<td>Short</td>
</tr>
<tr>
<td>MAX_GAIN</td>
<td>maxGain</td>
<td>Short</td>
</tr>
<tr>
<td>MIN_FROM_ONGOING</td>
<td>minFromOngoing</td>
<td>unsigned long</td>
</tr>
<tr>
<td>MAX_FROM_ONGOING</td>
<td>maxFromOngoing</td>
<td>unsigned long</td>
</tr>
</tbody>
</table>

Table 17  SCA capability properties for parameter validity properties

5.3.8 WCET properties

The PIM WCET properties, specified in section 4.15 of Transceiver Facility PIM Specification [Ref1], specify the maximum length of time, in nanoseconds (ns), possibly taken between the invocation and the return of a primitive.

For PIM WCET properties, SCA capability properties named `<primitiveName>Wcet` can be implemented, of type `unsigned long` (values in ns).

5.3.9 Unspecified mapping

The mapping of the following properties of Transceiver Facility PIM Specification [Ref1] is unspecified:

- Rapidity properties (section 4.8),
- Levels properties (section 4.10),
- Channelization properties (section 4.11),
- Temporal accuracy properties (section 4.12),
- Invocation lead time properties (section 4.13),
- Invocation delay (section 4.14).
6 References

6.1 Referenced documents

   https://sds.wirelessinnovation.org/specifications-and-recommendations
   https://winnf.memberclicks.net/assets/work_products/Specifications/WINNF-TS-0008-V2.1.1.pdf

   https://sds.wirelessinnovation.org/specifications-and-recommendations

[Ref3] Software Communications Architecture
   https://www.jtnc.mil/Resources-Catalog/Category/16990/sca/

   https://sds.wirelessinnovation.org/specifications-and-recommendations
   https://winnf.memberclicks.net/assets/work_products/Reports/WINNF-TR-2008-V1.0.1.pdf

   https://www.jtnc.mil/Resources-Catalog/Category/16990/sca/

   https://www.jtnc.mil/Resources-Catalog/Category/16990/sca/

The URLs above were successfully accessed at release date.
END OF THE DOCUMENT