

Joint Tactical Radio System (JTRS) Standard JTRS CORBA Types



Version: 1.0.2
02 April 2008

Statement A- Approved for public release; distribution is unlimited (29 March 2007)

REVISION HISTORY

Version	Authorization	Description	Last Modified Date
0.1		Initial Release	07-April-2006
0.2		Updates made from Increment 4 ICWG	04-May-2006
0.3		Additional sequence types	08-May-2006
0.4		Additional exception	08-June-2006
0.5		Updates made from Increment 4d review comments	11-July-2006
0.6		Added StringSequence	28-July-2006
1.0		Updates based in Increment 4d ICWG ICWG Approved	07-August-2006
1.0.1		Preparation for public release	29-March-2007
1.0.1.1		Errata: In JtrsCorbaTypes.idl replaced: <pre>#ifndef __CFTYPES_DEFINED #include "CfTypes.idl" #endif</pre> With: <pre>#include "CF.idl"</pre>	02-May-2007
1.0.2		Errata: In JtrsCorbaTypes.idl added #ifndef/#define wrapper around CF reference	02-April-2008

Table of Contents

A. JTRS CORBA TYPES 5

Table of Contents

A. JTRS CORBA TYPES	5
A.1 Introduction	5
A.1.1 Overview	5
A.1.2 Service Layer Description	5
A.1.3 Modes of Service	5
A.1.4 Service States	5
A.1.5 Referenced Documents	6
A.1.5.1 Government Documents	6
A.1.5.2 Commercial Standards	6
A.2 Services	7
A.3 Service Primitives and Attributes.....	7
A.4 IDL.....	8
A.4.1 JtrsCorbaTypes IDL.....	8
A.5 UML	10
A.5.1 Data Types	10
A.5.1.1 JTRS::BooleanSequence.....	10
A.5.1.2 JTRS::CharSequence	10
A.5.1.3 JTRS::ExtEnumSequence	10
A.5.1.4 JTRS::LongLongSequence	10
A.5.1.5 JTRS::LongSequence	10
A.5.1.6 JTRS::OctetSequence.....	10
A.5.1.7 JTRS::ShortSequence.....	10
A.5.1.8 JTRS::StringSequence.....	10
A.5.1.9 JTRS::UshortSequence.....	11
A.5.1.10 JTRS::UlongLongSequence.....	11
A.5.1.11 JTRS::UlongSequence.....	11
A.5.2 Enumerations.....	11
A.5.2.1 JTRS::ExtEnum.....	11
A.5.3 Exceptions	11
A.5.3.1 JTRS::InvalidParameter	11
A.5.3.2 JTRS::Unsupported	11
Appendix A.A Abbreviations and Acronyms	12
Appendix A.B Performance Specification.....	13

A. JTRS CORBA TYPES

A.1 INTRODUCTION

This document defines the set of common JTRS Common Object Request Broker Architecture (CORBA) Types to be used for JTRS Standard APIs.

A.1.1 Overview

- a. Section A.1, *Introduction*, contains the introductory material regarding the overview, service layer description, modes, states, and referenced documents of this document.
- b. Section A.2, *Services*.
- c. Section A.3, *Service Primitives and Attributes*.
- d. Section A.4, *IDL*.
- e. Section A.5, *UML*, specifies the data types provided by *JTRS CORBA Types*.
- f. Appendix A.A, *Abbreviations and Acronyms*.
- g. Appendix A.B, *Performance Specification*.

A.1.2 Service Layer Description

Not Applicable.

A.1.3 Modes of Service

Not Applicable.

A.1.4 Service States

Not Applicable.

A.1.5 Referenced Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein.

A.1.5.1 Government Documents

A.1.5.1.1 Specifications

A.1.5.1.1.1 Federal Specifications

None

A.1.5.1.1.2 Military Specifications

None

A.1.5.1.2 Other Government Agency Documents

[1] JTRS Standard, "Vocoder Service API," JPEO, Version 1.1.1.1.

[2] JTRS Standard, "Software Communication Architecture (SCA)," JPEO, Version 2.2.2.

A.1.5.2 Commercial Standards

None

A.2 SERVICES

Not Applicable.

A.3 SERVICE PRIMITIVES AND ATTRIBUTES

Not Applicable.

A.4 IDL

A.4.1 JtrsCorbaTypes IDL

```
/*
** JtrsCorbaTypes.idl - JTRS Standard Types
*/

#ifdef __JTRSCORBATYPES_DEFINED
#define __JTRSCORBATYPES_DEFINED

#ifdef __CF_DEFINED
#include "CF.idl"
#endif

#ifdef __PORTTYPES_DEFINED
#include "PortTypes.idl"
#endif

module JTRS
{
    //
    // ExtEnum - Extensible 'enum'
    // Used instead of intrinsic enum unless the definition is complete.
    // Enumeration values should be maintained as constant values by the
    // owning service or its extensions
    // (see: Vocoder.idl for example use).
    typedef unsigned short ExtEnum;
    typedef sequence<ExtEnum> ExtEnumSequence;

    //
    // Unbounded intrinsic sequence types
    // The 'floating point' types are not defined here as they will be
    // avoided on resource restricted platforms and the use of floating
    // point will be limited in the JTRS APIs.
    //
    typedef CF::OctetSequence OctetSequence;
    typedef CF::StringSequence StringSequence;
```



```
typedef PortTypes::BooleanSequence      BooleanSequence;
typedef PortTypes::CharSequence         CharSequence;
typedef PortTypes::LongLongSequence     LongLongSequence;
typedef PortTypes::LongSequence        LongSequence;
typedef PortTypes::ShortSequence        ShortSequence;
typedef PortTypes::UlongLongSequence    UlongLongSequence;
typedef PortTypes::UlongSequence        UlongSequence;
typedef PortTypes::UshortSequence       UshortSequence;

// Bad Method Parameter
exception InvalidParameter
{
};

// Behaviour is not supported (on this platform - others might!)
exception Unsupported
{
};

};
#endif // __JTRSCORBATYPES_DEFINED
```

A.5 UML

A.5.1 Data Types

A.5.1.1 JTRS::BooleanSequence

BooleanSequence is a sequence of type boolean.

```
typedef PortTypes::BooleanSequence BooleanSequence;
```

A.5.1.2 JTRS::CharSequence

CharSequence is a sequence of type char.

```
typedef PortTypes::CharSequence CharSequence;
```

A.5.1.3 JTRS::ExtEnumSequence

ExtEnumSequence is a sequence of the extensible 'enum' types defined in A.5.2.1.

```
typedef sequence<ExtEnum> ExtEnumSequence;
```

A.5.1.4 JTRS::LongLongSequence

LongLongSequence is a sequence of type long long.

```
typedef PortTypes::LongLongSequence LongLongSequence;
```

A.5.1.5 JTRS::LongSequence

LongSequence is a sequence of type long.

```
typedef PortTypes::LongSequence LongSequence;
```

A.5.1.6 JTRS::OctetSequence

OctetSequence is a sequence of type octet.

```
typedef CF::OctetSequence OctetSequence;
```

A.5.1.7 JTRS::ShortSequence

ShortSequence is a sequence of type short.

```
typedef PortTypes::ShortSequence ShortSequence;
```

A.5.1.8 JTRS::StringSequence

StringSequence is a sequence of type string.

```
typedef CF::StringSequence    StringSequence;
```

A.5.1.9 JTRS::UshortSequence

UshortSequence is a sequence of type unsigned short.

```
typedef PortTypes::UshortSequence    UshortSequence;
```

A.5.1.10 JTRS::UlongLongSequence

UlongLongSequence is a sequence of type unsigned long long.

```
typedef PortTypes::UlongLongSequence    UlongLongSequence;
```

A.5.1.11 JTRS::UlongSequence

UlongSequence is a sequence of type unsigned long.

```
typedef PortTypes::UlongSequence    UlongSequence;
```

A.5.2 Enumerations

A.5.2.1 JTRS::ExtEnum

The *ExtEnum* type definition defines an extensible ‘enum’. It is used instead of the intrinsic enum unless the definition is complete. Enumeration values should be maintained as constant values by the owning service or its extensions.

```
typedef          unsigned short          ExtEnum;
```

A.5.3 Exceptions

A.5.3.1 JTRS::InvalidParameter

The *InvalidParameter* exception indicates that the input parameter is invalid.

```
exception InvalidParameter {  
};
```

A.5.3.2 JTRS::Unsupported

The *Unsupported* exception indicates that the input parameter is not supported on the specified platform.

```
exception Unsupported {  
};
```

Appendix A.A Abbreviations and Acronyms

API	Application Program Interface
CF	Core Framework
CORBA	Common Object Request Broker Architecture
ICWG	Interface Control Working Group
IDL	Interface Definition Language
JPEO	Joint Program Executive Office
JTRS	Joint Tactical Radio System
SCA	Software Communications Architecture

Appendix A.B Performance Specification

Not Applicable.