## REVISION HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Authorization</th>
<th>Description</th>
<th>Last Modified Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td></td>
<td>Initial release</td>
<td>12-October-2006</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Revised document/interface names.</td>
<td>26-October-2006</td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td>Updates based in Increment 6 comments.</td>
<td>06-December-2006</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>Updated Appendix A.B and References.</td>
<td>13-December-2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICWG Approved</strong></td>
<td></td>
</tr>
<tr>
<td>1.0.1</td>
<td></td>
<td>Preparation for public release</td>
<td>29-March-2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICWG Approved</strong></td>
<td></td>
</tr>
<tr>
<td>1.0.2</td>
<td></td>
<td>Sections A.2.4.1, A.3.1.1, A.3.1.2, A.3.2.1: -Corrected redline in text</td>
<td>05-May-2010</td>
</tr>
</tbody>
</table>
Table of Contents

A. DEVICE IO API............................................................................................ 6
# Table of Contents

A. DEVICE IO API ........................................................................................................... 6

A.1 Introduction .............................................................................................................. 6
   A.1.1 Overview ............................................................................................................ 6
   A.1.2 Service Layer Description ............................................................................... 7
      A.1.2.1 Device IO Port Diagram .......................................................................... 7
   A.1.3 Modes of Service .............................................................................................. 8
   A.1.4 Service States ................................................................................................... 8
   A.1.5 Referenced Documents .................................................................................... 8
      A.1.5.1 Government Documents .......................................................................... 8
      A.1.5.2 Commercial Standards ........................................................................... 8

A.2 Services .................................................................................................................... 9
   A.2.1 Provide Services ............................................................................................... 9
   A.2.2 Use Services ..................................................................................................... 9
   A.2.3 Interface Modules ............................................................................................ 10
      A.2.3.1 Device IO .................................................................................................. 10
   A.2.4 Sequence Diagrams ......................................................................................... 12
      A.2.4.1 RTS/CTS Sequence ................................................................................. 12

A.3 Service Primitives and Attributes .......................................................................... 13
   A.3.1 DeviceIo::DeviceIoControl............................................................................. 14
      A.3.1.1 enableRtsCts Operation ......................................................................... 14
      A.3.1.2 setRts Operation ............................................................................... 15
   A.3.2 DeviceIo::DeviceIoSignals ............................................................................. 16
      A.3.2.1 setCts Operation ............................................................................... 16

A.4 IDL ............................................................................................................................ 17
   A.4.1 DeviceIo IDL ................................................................................................... 17

A.5 UML .......................................................................................................................... 18
   A.5.1 Data Types ....................................................................................................... 18
   A.5.2 Enumerations ................................................................................................... 18
   A.5.3 Exceptions ........................................................................................................ 18
   A.5.4 Structures ......................................................................................................... 18

Appendix A.A Abbreviations and Acronyms ................................................................. 19
Appendix A.B Performance Specification ................................................................. 20
Lists of Figures

FIGURE 1 – DEVICE IO PORT DIAGRAM ................................................................. 7
FIGURE 2 - DEVICE IO CLASS DIAGRAM ........................................................... 10
FIGURE 3 – DEVICE IO DATA CONSUMER AND DATA PRODUCER INTERFACES ......................................................... 10
FIGURE 4 – DEVICE IO CONTROL INTERFACE ............................................... 10
FIGURE 5 – DEVICE IO SIGNALS INTERFACE .................................................... 11
FIGURE 6 – RTS/CTS SEQUENCE DIAGRAM ................................................... 12
FIGURE 7 – DEVICE IO COMPONENT DIAGRAM ............................................. 18
A. DEVICE IO API

A.1 INTRODUCTION

This document defines a common set of Device IO interfaces to be used by Joint Tactical Radio (JTR) Set Applications and Services. The Device IO interfaces provide methods to enable and signal Request To Send (RTS) and Clear To Send (CTS) messages.

The Device IO interfaces may be used in conjunction with the Packet API [1] interfaces to create the device/service Data Producer and Data Consumer interfaces. The Device IO interfaces are documented within to minimize coupling between the device and service interfaces that utilize these Device IO interfaces.

A.1.1 Overview

This document contains as follows:

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, specifies the interfaces for the component, port connections, and sequence diagrams.

c. Section A.3, Service Primitives and Attributes, specifies the operations that are provided by the Device IO interfaces.

d. Section A.4, IDL.

e. Section A.5, UML.

f. Appendix A.A, Abbreviations and Acronyms.

g. Appendix A.B, Performance Specification.
A.1.2 Service Layer Description

A.1.2.1 Device IO Port Diagram

In Figure 1, port definitions, and port names are for reference only.

The service/device will specify all interfaces comprising the Data Producer and Data Consumer. Figure 1 specifies a generic bidirectional instantiation of the Device IO interfaces to be specified by a service or device. A unidirectional instantiation may be achieved by only specifying the “uses” or the “provides” ports.

![Device IO Port Diagram](image)

Device IO Provides Ports Definitions

- `devio_consumer_provides_port` is provided by the Device or Service to enable and initiate RTS/CTS handshaking.
- `devio_producer_provides_port` is provided by the Device or Service to signal a clear to send condition.

Device IO Uses Ports Definitions

- `devio_consumer_uses_port` is used by the Device or Service to enable and initiate RTS/CTS handshaking.
- `devio_producer_uses_port` is used by the Device or Service to signal a clear to send condition.
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


A.1.5.2 Commercial Standards
None
A.2 SERVICES

A.2.1 Provide Services

Table 1 describes the generic provides service interfaces to be specified by the service or device. These provide service interfaces correspond to the port diagram in Figure 1.

<table>
<thead>
<tr>
<th>Service Group (Port Name)</th>
<th>Service (Interface Provided)</th>
<th>Primitives (Provided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>devio_consumer_provides_port</td>
<td>DeviceIo::DeviceIoControl</td>
<td>enableRtsCts()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>setRts()</td>
</tr>
<tr>
<td>devio_producer_provides_port</td>
<td>DeviceIo::DeviceIoSignals</td>
<td>setCts()</td>
</tr>
</tbody>
</table>

A.2.2 Use Services

Table 2 describes the generic uses service interfaces to be specified by the service or device. These provide service interfaces correspond to the port diagram in Figure 1.

<table>
<thead>
<tr>
<th>Service Group (Port Name)</th>
<th>Service (Interface Provided)</th>
<th>Primitives (Provided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>devio_consumer_uses_port</td>
<td>DeviceIo::DeviceIoControl</td>
<td>enableRtsCts()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>setRts()</td>
</tr>
<tr>
<td>devio_producer_uses_port</td>
<td>DeviceIo::DeviceIoSignals</td>
<td>setCts()</td>
</tr>
</tbody>
</table>
A.2.3 Interface Modules

A.2.3.1 Device IO

The class diagram for the Device IO interfaces is shown in Figure 2.

The Device IO Data Producer and Data Consumer interfaces are shown in Figure 3. The Data Producer and Data Consumer interfaces shown in green are not defined in this API. They have been provided to illustrate the collection of interfaces which will be specified by the service/device or service/device user. Additional interfaces may be required to complete the Data Producer/Consumer interfaces (e.g. Packet API [1]).

A.2.3.1.1 Device IO Control Interface Description

The interface design of Device IO Control is shown in Figure 4. It provides methods to activate or enable RTS/CTS handshaking (enableRtsCts). It also provides the operation to initiate a request to send (setRts). The Device IO Control interface may be inherited by a Data Consumer.
A.2.3.1.2 Device IO Signals Interface Description

The interface design of Device IO Signals is shown in Figure 5. It provides methods to call or signal a clear to send condition (setCts). The Device IO Signals interface may be inherited by a Data Producer.

```
<<Interface>>
DeviceIoSignals
(from DeviceIo)

setCts()
```

Figure 5 – Device IO Signals Interface
A.2.4 Sequence Diagrams

A.2.4.1 RTS/CTS Sequence

Description
The RTS/CTS Sequence Diagram is shown in Figure 6.

The Data Producer is the interface inheriting the Device IO Signals interface while the Data Consumer is the interface inheriting the Device IO Control interface.

The Data Producer activates RTS/CTS handshaking by calling enableRtsCts(TRUE). When the Data Producer wishes to send packet to the Data Consumer it will initiate the exchange by calling setRts(TRUE). The Data Consumer will respond with a setCts(TRUE) when it is ready to accept packets. The Data Producer can now send packets to the Data Consumer.

Pre-conditions
None

Post-conditions
Data flow is controlled by RTS/CTS handshaking.

![Figure 6 – RTS/CTS Sequence Diagram](image-url)
A.3 SERVICE PRIMITIVES AND ATTRIBUTES

To enhance the readability of this API document and to avoid duplication of data, the type definitions of all structured types (i.e., data types, enumerations, exceptions, and structures) used by the Service Primitives and Attributes have been co-located in section A.5. This cross-reference of types also includes any nested structures in the event of a structure of structures or an array of structures.
A.3.1 DeviceIo::DeviceIoControl

A.3.1.1 enableRtsCts Operation

The enableRtsCts operation is called by a Data Producer to enable RTS/CTS signals when starting the data flow.

A.3.1.1.1 Synopsis

oneway void enableRtsCts(in boolean enable);

A.3.1.1.2 Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>boolean</td>
<td>Indicates whether to enable the RTS/CTS method of flow control.</td>
<td>TRUE = enable; FALSE = disable</td>
</tr>
</tbody>
</table>

A.3.1.1.3 State

Not applicable.

A.3.1.1.4 New State

Not applicable.

A.3.1.1.5 Return Value

None

A.3.1.1.6 Originator

Service Provider, Service User.

A.3.1.1.7 Exceptions

None
A.3.1.2 *setRts Operation*

The `setRts` operation is called by a *Data Producer* to send a RTS signal when there is data to be pushed. The `setRts` operation indicates that the RTS signal has been activated. In most cases this represents a “request to transmit” action, but could also indicate a simple transition on the RTS or CTS signal, as is depending on the sourcing components particular behavior.

A.3.1.2.1 **Synopsis**

```plaintext
oneway void setRts(in boolean rtsToState);
```

A.3.1.2.2 **Parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsToState</td>
<td>boolean</td>
<td>Indicates whether the I/O device is ready to</td>
<td>TRUE = enable; FALSE = disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>send data to the interface.</td>
<td></td>
</tr>
</tbody>
</table>

A.3.1.2.3 **State**

Not applicable.

A.3.1.2.4 **New State**

Not applicable.

A.3.1.2.5 **Return Value**

None

A.3.1.2.6 **Originator**

Service Provider, Service User.

A.3.1.2.7 **Exceptions**

None
A.3.2 DeviceIo::DeviceIoSignals

A.3.2.1 setCts Operation

The setCts operation is called by a Data Consumer to transition the consent control (accept or decline) for the flow of data packets.

A.3.2.1.1 Synopsis

oneway void setCts(in boolean ctsToState);

A.3.2.1.2 Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctsToState</td>
<td>boolean</td>
<td>Indicates whether to accept the data flow.</td>
<td>TRUE = enable; FALSE = disable</td>
</tr>
</tbody>
</table>

A.3.2.1.3 State

Not applicable.

A.3.2.1.4 New State

Not applicable.

A.3.2.1.5 Return Value

None

A.3.2.1.6 Originator

Service Provider, Service User.

A.3.2.1.7 Exceptions

None
A.4 IDL

A.4.1 DeviceIo IDL

/*
** DeviceIo.idl
*/

#ifndef __DEVICEIO_DEFINED
#define __DEVICEIO_DEFINED

module DeviceIo
{
    interface DeviceIoControl
    {
        oneway void enableRtsCts( in boolean enable );
        oneway void setRts( in boolean rtsToState );
    };

    interface DeviceIoSignals
    {
        oneway void setCts( in boolean ctsToState );
    };
}
#endif //__DEVICEIO_DEFINED
A.5 UML

This section contains Device IO component UML diagram and the definitions of all data types the definitions of all data types referenced (directly or indirectly) by A.3 Service Primitives and Attributes.

![Device IO Component Diagram]

Figure 7 – Device IO Component Diagram

A.5.1 Data Types
None

A.5.2 Enumerations
None

A.5.3 Exceptions
None

A.5.4 Structures
None
## Appendix A.A  Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>ICWG</td>
<td>Interface Control Working Group</td>
</tr>
<tr>
<td>IDL</td>
<td>Interface Definition Language</td>
</tr>
<tr>
<td>IO</td>
<td>Input/Output</td>
</tr>
<tr>
<td>JPEO</td>
<td>Joint Program Executive Office</td>
</tr>
<tr>
<td>JTRS</td>
<td>Joint Tactical Radio System</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
</tbody>
</table>
Appendix A.B   Performance Specification

Table 3 provides a template for the generic performance specification for the Device IO API which will be documented in the service or device using the interface. This performance specification corresponds to the port diagram in Figure 1.

Table 3 – Device IO Extension Performance Specification

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst Case Command Execution Time for</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>devio_consumer_provides_port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worst Case Command Execution Time for</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>devio_consumer_uses_port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worst Case Command Execution Time for</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>devio_producer_provides_port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worst Case Command Execution Time for</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>devio_producer_uses_port</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: (*) These values should be filled in by individual developers.