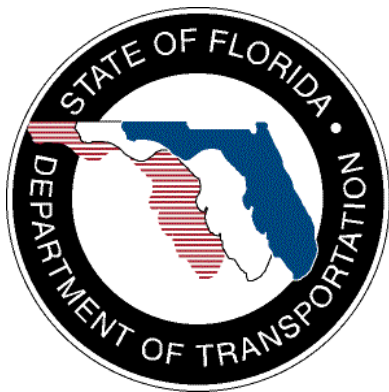


SunGuideSM:

**Data Bus
Client Interface Manager
Interface Control Document**

SunGuide-DB-CIM-ICD-1.0.0



Prepared for:

Florida Department of Transportation
Traffic Engineering and Operations Office
605 Suwannee Street, M.S. 90
Tallahassee, Florida 32399-0450
(850) 410-5600

November 11, 2004

DataBus Interface Control Document

Document Control Panel			
File Name:	SunGuide-DB-CIM-ICD-1.0.0.doc		
File Location:	SunGuide CM Repository		
CDRL:	2-7.1		
	Name	Initial	Date
Created By:	Meredith Moczygemba, SwRI	MRM	11/11/04
Reviewed By:	Steve Dellenback, SwRI	SWD	11/11/04
	Steve Novosad, SwRI	SEN	11/11/04
Modified By:			
Completed By:			

Table of Contents

1.	Scope	1
1.1	Document Identification	1
1.2	Project Overview	1
1.3	Related Documents	2
1.4	Contacts	3
2.	Data	4
2.1	Schema	4
2.1.1	Data Bus User Client Communication	4
2.1.1.1	Retrieve Data Types	4
2.1.1.2	Status Data Description	5
2.1.1.3	Subscribe	6
2.1.1.4	Status	7
2.1.1.5	Subsystem Commands.....	7
2.1.2	Data Provider Communication	7
2.2	Examples	7
2.3	Subsystem Schemas	9
2.4	Driver Schemas	10
3.	Notes	11
	Attachment 1	12
A.1	dataReq.xsd.....	1
A.2	deletionStatus.xsd	2
A.3	providerStatus.xsd.....	3
A.4	status.xsd.....	4
A.5	retrieveDataTypesReq.xsd	5
A.6	statusReq.xsd	6
A.7	subscribeReq.xsd	7
A.8	retrieveDataTypesResp.xsd	8
A.9	statusResp.xsd.....	10
A.10	subscribeResp.xsd.....	11
A.11	providerDisconnectMsg.xsd	13
A.12	providerReconnectMsg.xsd	14
A.13	statusUpdateMsg.xsd	15

List of Figures

Figure 1-1 - High-Level Architectural Concept.....	2
Figure 2-1 - Provider Status.....	5
Figure 2-2 - Id.....	6
Figure 2-3 - Status.....	6
Figure 2-4 - Sample Data Bus Transaction.....	7
Figure 2-5 - Sample Command Transaction.....	8

List of Acronyms

ATMS	Advanced Traffic Management System
DOT	Department of Transportation
FDOT	Florida Department of Transportation
IM	Incident Management
ITS	Intelligent Transportation Systems
ITN	Invitation to Negotiate
SwRI	Southwest Research Institute
TMC	Traffic Management Center
XML	Extensible Markup Language

REVISION HISTORY

Revision	Date	Changes
1.0.0	November 4, 2002	Initial Release

1. Scope

1.1 Document Identification

This Interface Control Document (ICD) describes the system interface between the individual SunGuide clients and the Data Bus. The provider interface between the individual Traffic Management Center (TMC) subsystems and the Data Bus is also described. The general base architecture of the XML communications including connection information, byte order and base transaction classes is delineated in this document. This ICD defines Extensible Markup Language (XML) schemas upon which XML requests shall be based in communicating amongst the various processes. Refer to the SunGuide-General-ICD-1.0.0 document for details regarding data transfer.

1.2 Project Overview

The Florida Department of Transportation (FDOT) is conducting a program that is developing SunGuideSM software. The SunGuideSM software is a set of Intelligent Transportation System (ITS) software that allows the control of roadway devices as well as information exchange across a variety of transportation agencies. The goal of the SunGuideSM software is to have a common software base that can be deployed throughout the state of Florida. The SunGuideSM software development effort is based on ITS software available from both the states of Texas and Maryland; significant customization of the software is being performed as well as the development of new software modules. The following figure provides a graphical view of the software to be developed:

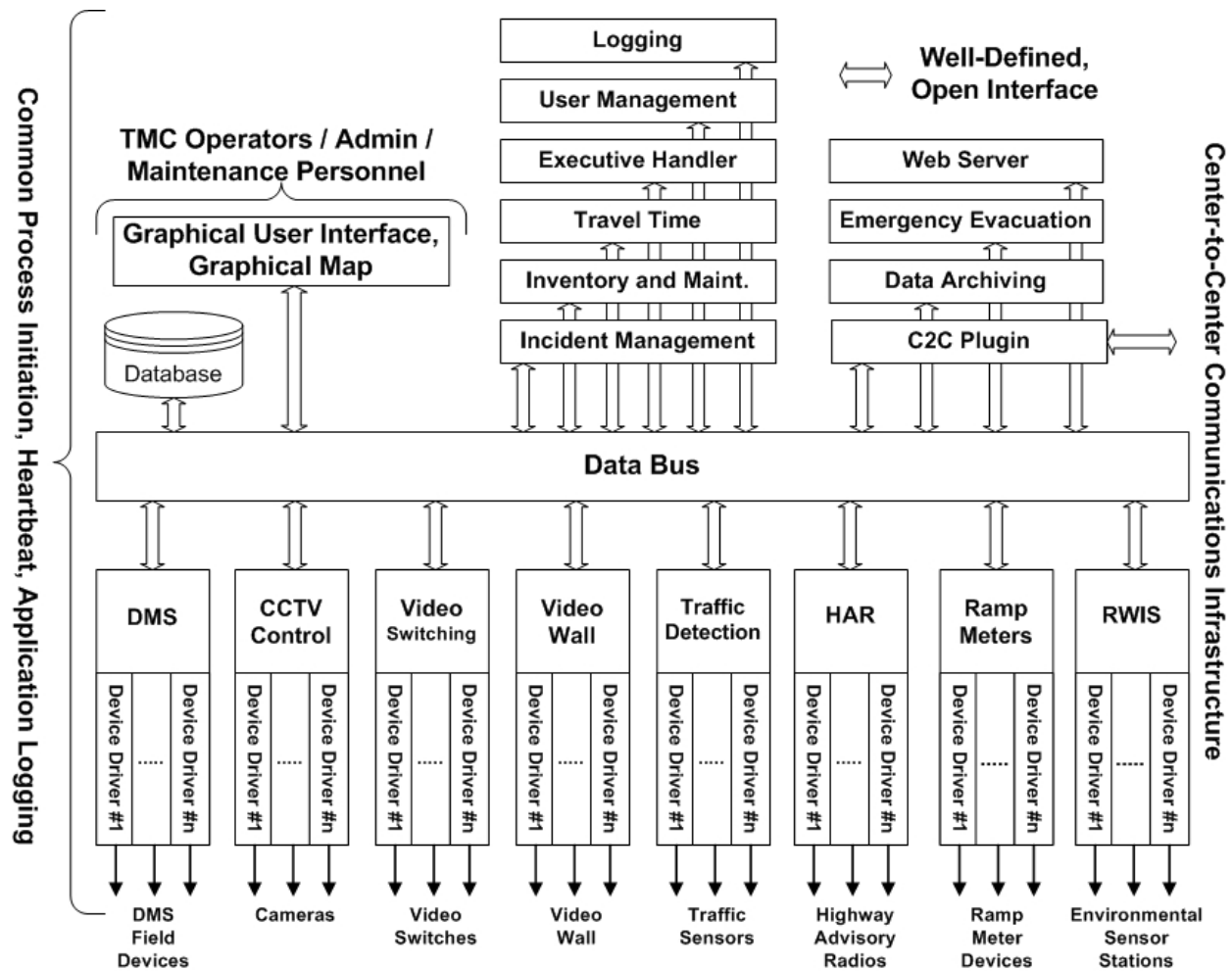


Figure 1-1 - High-Level Architectural Concept

The SunGuideSM development effort spans approximately two years. After the development, the software will be deployed to a number of Districts and Expressway Authorities throughout Florida and support activities will be performed.

1.3 Related Documents

The following documents were used to develop this document:

- SwRI Qualification Response: *Response to the Invitation to Negotiate (ITN): Statewide Transportation Management Center Software Library System, Negotiation Number: ITN-DOT-02/03-9025-RR*, SwRI Proposal No. 10-35924, dated: November 18, 2002.
- SwRI Technical Proposal: *Technical Proposal for Invitation to Negotiate (ITN): Statewide Transportation Management Center Software Library System, Negotiation Number: ITN-DOT-02/03-9025-RR*, SwRI Proposal No. 10-35924, dated: January 31, 2003.

- SwRI Cost Proposal: *Cost Proposal for Invitation to Negotiate (ITN): Statewide Transportation Management Center Software Library System, Negotiation Number: ITN-DOT-02/03-9025-RR*, SwRI Proposal No. 10-35924, dated: January 31, 2003.
- SwRI BAFO letter: *Southwest Research Institute[®] Proposal No. 10-35924, “Invitation to Negotiate (ITN): Statewide Transportation Management Center Software Library System”*, Reference: Negotiation Number: ITN-DOT-02/03-9025-RR, dated: May 5, 2003.
- FDOT procurement document: *Invitation To Negotiate (ITN), Negotiation Number: ITN-DOT-02/03-9025-RR, Statewide Transportation Management Center Software Library System*, dated: October 21, 2002.
- FDOT Scope of Services: *Statewide Transportation Management Center Software Library System: Scope of Services*, September 22, 2003.
- FDOT Requirements Document: *Statewide Transportation Management Center Software Library System: Requirements Specification*, June 3, 2003.
- Southwest Research Institute, *TMC Software Study*, November 15, 2001.
- Southwest Research Institute, *Introduction to an Operational Concept For the Florida Statewide Library*, FDOT – OCD – 1.0, March 31, 2002.
- World Wide Web Consortium (W3) website: <http://www.w3.org>.
- SunGuideSM Project website: <http://sunguide.datasys.swri.edu>.

1.4 Contacts

The following are contact persons for the SunGuideSM software project:

- Elizabeth Birriel, ITS Central Office, elizabeth.birriel@dot.state.fl.us, 850-410-5606
- Liang Hsia, FDOT Project Manager, liang.hsia@dot.state.fl.us, 850-410-5615
- John Bonds, Senior ITS Specialist, jbonds@pbsj.com, 408-873-2514
- David Chang, ITS Specialist, David.Chang@dot.state.fl.us, 850-410-5622
- Steve Dellenback, SwRI Project Manager, sdellenback@swri.org, 210-522-3914
- Robert Heller, SwRI Software Project Manager, rheller@swri.org, 210-522-3824
- Charlie Wallace, PBF Deputy Project Manager, WallaceC@pbworld.com, 352-374-6635
- John Schumitz, PBF Software Project Manager, schumitz@pbworld.com, 301-816-1852

The following are contacts that will be used by the SunGuideSM software project team to assure consistency with other FDOT projects and FDOT procedures:

- Dan Baxter, PB Farradyne, FDOT C2C Project, baxter@pbworld.com, 407-587-7809
- David Lambert, University of North Florida, RWIS, jlambert@unf.edu, 904-620-3881
- Bob Colins, PBS&J, Emergency Evacuation, bobcolins@pbsj.com, 850-575-1800
- John Fain, FDOT, Comptroller, john.fain@dot.state.fl.us, 850-921-7332
- Leslie Jacobson, PB Farradyne, Ramp Metering, jacobsonl@pbworld.com, 206-382-5290

2. Data

The following sections detail the XML transactions that can be exchanged between client and server applications.

2.1 Schema

The schemas for these transactions may be located in the Schemas1.0 directory. The objects directory contains common data schemas that are used by the various request/messages/responses. Schemas are organized in the following tree structure:

- databus
 - messages
 - [providerDisconnectMsg.xsd](#)
 - [providerReconnectMsg.xsd](#)
 - [statusUpdateMsg.xsd](#)
 - objects
 - [dataReq.xsd](#)
 - [deletionStatus.xsd](#)
 - [providerStatus.xsd](#)
 - [status.xsd](#)
 - requests
 - [retrieveDataTypesReq.xsd](#)
 - [statusReq.xsd](#)
 - [subscribeReq.xsd](#)
 - responses
 - [retrieveDataTypesResp.xsd](#)
 - [statusResp.xsd](#)
 - [subscribeResp.xsd](#)

Requests may be sent from a client to the Data Bus or from the Data Bus to a data provider subsystem. Responses may be sent from a data provider to the Data Bus or from the Data Bus to a client. A message can be sent from any process to another process.

2.1.1 Data Bus User Client Communication

2.1.1.1 Retrieve Data Types

The client may determine what providers and resource types are managed by the system by sending a retrieve data types request. The *retrieveDataTypesResp* returns the names of providers and their associated resource types as specified in the configuration file. The Data Bus will attempt to connect to these providers on startup and manage the data types within its status DOM (Document Object Model) trees. Retrieval of a provider name does not guarantee the Data Bus has successfully connected to this data provider or that the respective DOM tree contains data.

2.1.1.2 Status Data Description

Each status resource type stored in the Data Bus consists of an *id* element and a *status* element as shown in the provider status figure below. This *id* element contains attributes for the pertinent provider name, resource type, and center id values. Status stored in the Data Bus is dependent upon the data retrieved from the providers via the *retrieveDataResp*. The Data Bus requests *statusList* from each provider when retrieving status information. Subsystem-specific status data managed by the Data Bus are referenced in the appropriate provider ICDs.

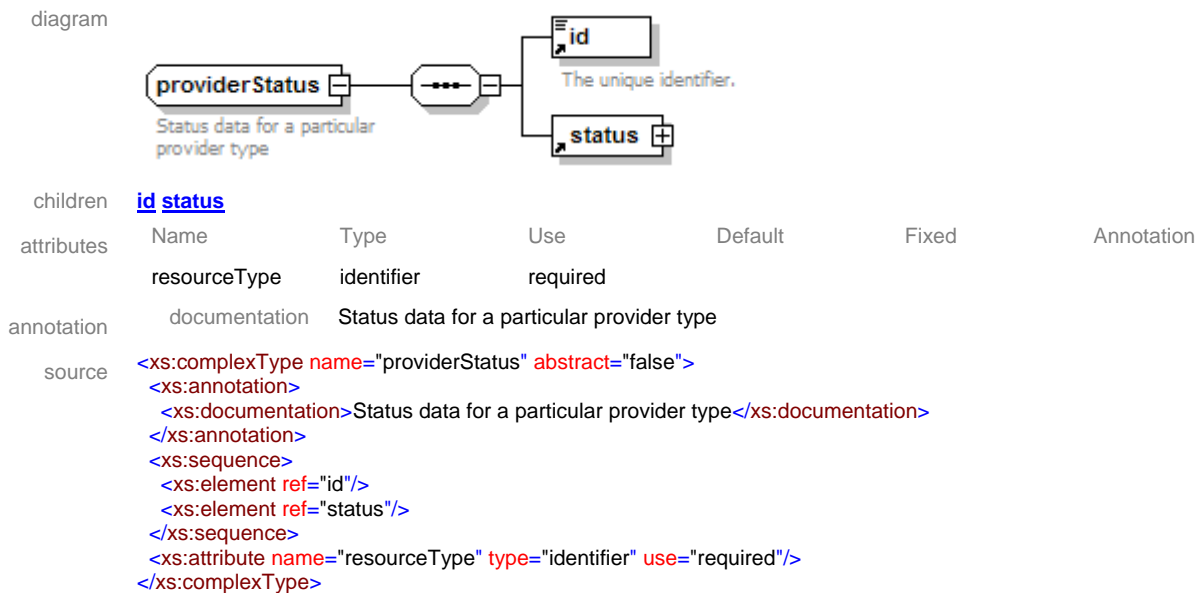
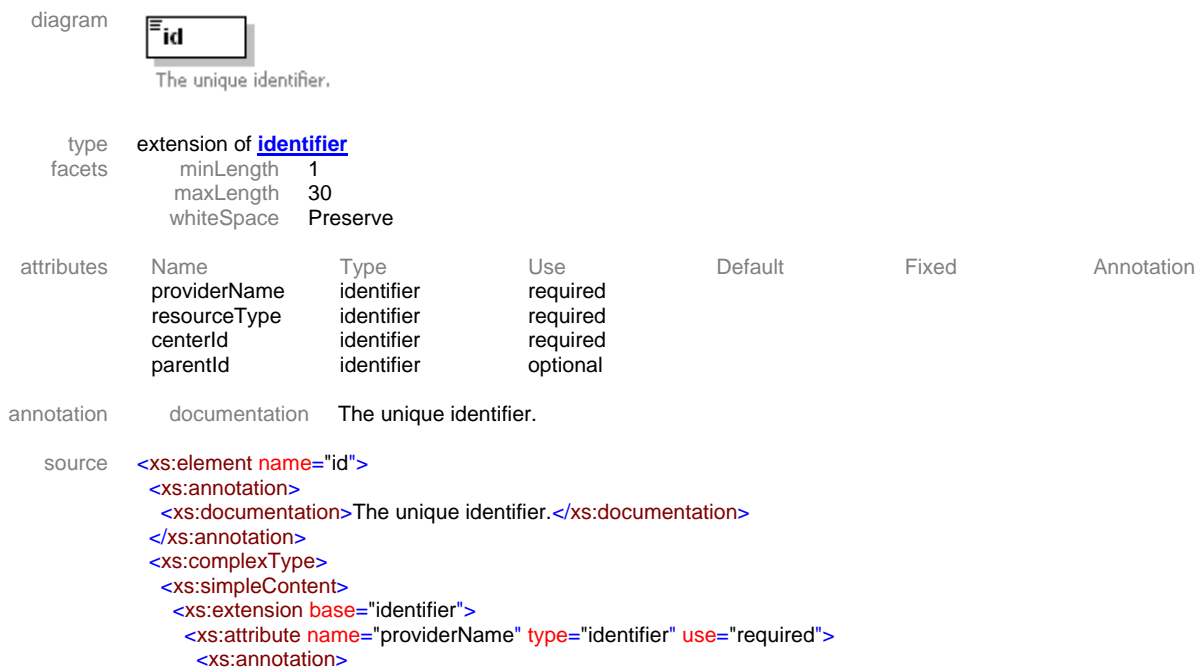


Figure 2-1 - Provider Status



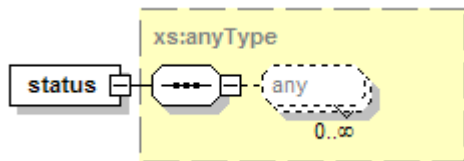
```

        <xs:documentation source="The name of the provider or subsystem to which this item belongs"/>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="resourceType" type="identifier" use="required">
    <xs:annotation>
        <xs:documentation source="The type of resource of this item (e.g., link, dms, camera)"/>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="centerId" type="identifier" use="required">
    <xs:annotation>
        <xs:documentation source="The name of the center that owns this item"/>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="parentId" type="identifier" use="optional">
    <xs:annotation>
        <xs:documentation source="The parent id name, if one exists"/>
    </xs:annotation>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

```

Figure 2-2 - Id

diagram



type	extension of xs:anyType					
used by	complexType	providerStatus				
attributes	Name	Type	Use	Default	Fixed	Annotation
source	<pre> <xs:element name="status"> <xs:complexType> <xs:complexContent> <xs:extension base="xs:anyType"/> </xs:complexContent> </xs:complexType> </xs:element> </pre>					

Figure 2-3 - Status

2.1.1.3 Subscribe

Another request that a client might send to the Data Bus is a subscribe request. The client can identify what types of data updates should be sent. Then, if the specified data changes at a later time, the client will receive unsolicited responses with updated data. The subscribe response will return subscription status information for data to which the client has subscribed. Subscriptions will fail if the data types requested for subscription do not match those indicated by the *retrieveDataTypesResp*. If a client subscribes more than once, the latest subscription will override previously stored subscription data in the system. When a client has successfully subscribed to data updates and wishes to clear his/her subscriptions, an XML subscribe request should be sent to the Data Bus with no status data types requested.

2.1.1.4 Status

SunGuide clients connect to the Data Bus and until a status or subscribe request is sent, no status information will be transmitted or sent to the client. Each connected client lists the desired data types to receive status information. Status requests will not return status data if the data types requested do not match those indicated by the *retrieveDataTypesResp*.

2.1.1.5 Subsystem Commands

Other command schemas and documentation for the provider subsystems can be viewed in the individual subsystem ICDs. For all subsystem command requests sent to the Data Bus, the *TransactionType* attribute named *providerName* is required. This *providerName* must match the provider specification defined in the Data Bus configuration file or an error is returned. All requests, messages, and responses must use the appropriate *Req*, *Msg*, and *Resp* suffixed nomenclature to be successfully routed from client to subsystem and back.

2.1.2 Data Provider Communication

Communication between the Data Bus and provider subsystems is described in the Data Bus Provider Template ICD.

2.2 Examples

For example, if a client wishes to determine what data types are supported by the Data Bus, the client sends a *retrieveDataTypesReq* to the Data Bus. The Data Bus returns a *retrieveDataTypesResp* to the calling client.

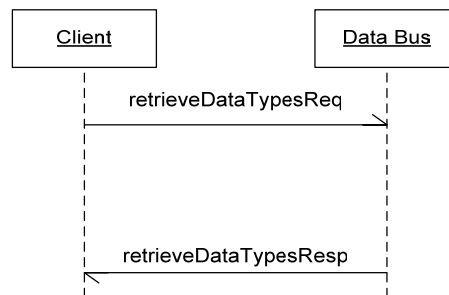


Figure 2-4 - Sample Data Bus Transaction

If a client wishes to send a command request to a data provider connected to the Data Bus, the client sends the command request to the Data Bus, setting the *providerName* *TransactionType* attribute as appropriate. The Data Bus forwards the command request to the appropriate provider, and once the provider processes the request, a provider-generated response is sent to the Data Bus. The Data Bus then sends this response back to the appropriate client.

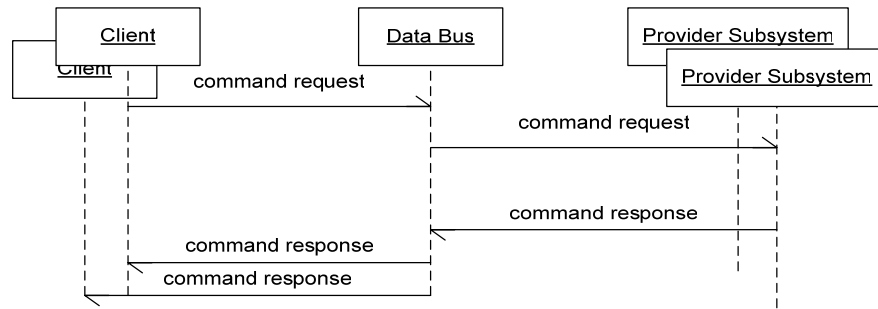


Figure 2-5 - Sample Command Transaction

The tables below show which requests can be sent from a client to the Data Bus and from the Data Bus to a provider subsystem. The responses sent from a provider subsystem to the Data Bus and subsystem to client are also specified. Messages are sent when a response is not required.

2.3 Subsystem Schemas

FC (From client), TC (To client), TP (To provider), FP (From provider)

Usage Description	Requests	FC	TP	Responses	FP	TC	Messages	TP	FP	TC
Used to inform connected clients when a provider subsystem has disconnected from the Data Bus.							providerDisconnectMsg			X
Used to inform connected clients when the Data Bus has reconnected with a provider subsystem.							providerReconnectMsg			X
Used for a client to receive status data types from the Data Bus. (These data types indicate the status subscriptions supported by the Data Bus).	retrieveDataTypesReq	X		retrieveDataTypesResp		X				
Used for a client to receive status information.	statusReq	X		statusResp		X				
Used to send status updates to subscribed clients.							statusUpdateMsg			X

Usage Description	Requests	FC	TP	Responses	FP	TC	Messages	TP	FP	TC
Used for a client to subscribe to status data updates from the Data Bus.	subscribeReq	X		subscribeResp		X				

2.4 Driver Schemas

The provider subsystem ICDs may be referenced for more information regarding command and control requests, responses, and messages routed by the Data Bus.

3. Notes

Information about XML and schemas can be found at the World Wide Web Consortium (W3) website at <http://www.w3.org>.

Attachment 1

Data Bus XML Schema

A.1 dataReq.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="dataReq" type="xs:string">
    <xs:annotation>
      <xs:documentation>Contains selected data types for update
notifications.</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:schema>
```

A.2 deletionStatus.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:include schemaLocation="../../dataBus/objects/status.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:include schemaLocation="../../common/objects/id.xsd"/>
  <xs:complexType name="deletionStatus" abstract="false">
    <xs:annotation>
      <xs:documentation>Deleted status data for a particular provider
type</xs:documentation>
    </xs:annotation>
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="id"/>
    </xs:sequence>
    <xs:attribute name="resourceType" type="identifier" use="required"/>
  </xs:complexType>
</xs:schema>
```

A.3 providerStatus.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:include schemaLocation="../../dataBus/objects/status.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:include schemaLocation="../../common/objects/id.xsd"/>
  <xs:complexType name="providerStatus" abstract="false">
    <xs:annotation>
      <xs:documentation>Status data for a particular provider type</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="status"/>
    </xs:sequence>
    <xs:attribute name="resourceType" type="identifier" use="required"/>
  </xs:complexType>
</xs:schema>
```

A.4 status.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="status">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="xs:anyType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.5 retrieveDataTypesReq.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../objects/dataReq.xsd"/>
  <xs:element name="retrieveDataTypesReq">
    <xs:annotation>
      <xs:documentation>A request sent by client to receive status data types from the
system.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="TransactionType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.6 statusReq.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../objects/dataReq.xsd"/>
  <xs:element name="statusReq">
    <xs:annotation>
      <xs:documentation>A request sent by client to receive status from the
system.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="TransactionType">
          <xs:sequence maxOccurs="unbounded">
            <xs:element ref="dataReq"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```


A.7 subscribeReq.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../objects/dataReq.xsd"/>
  <xs:element name="subscribeReq">
    <xs:annotation>
      <xs:documentation>A request sent by client to subscribe to data updates from the
system</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="TransactionType">
          <xs:sequence maxOccurs="unbounded">
            <xs:element ref="dataReq" minOccurs="0"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.8 retrieveDataTypesResp.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:include schemaLocation="../../objects/providerStatus.xsd"/>
  <xs:element name="retrieveDataTypesResp">
    <xs:annotation>
      <xs:documentation>A response received for a retrieve data types
request.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ResponseType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="retrieveDataTypesData">
    <xs:annotation>
      <xs:documentation>The response data for a retrieve data types
request.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="responseData">
        <xs:sequence>
          <xs:element name="providers">
            <xs:annotation>
              <xs:documentation>Stores the status provider names and pertinent
resource data types that the Data Bus will attempt to monitor.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:sequence maxOccurs="unbounded">
                <xs:element name="provider">
                  <xs:annotation>
                    <xs:documentation>Stores the rovider's subsystem name (i.e.
cctv1, dms1) and pertinent resource data types.</xs:documentation>
                  </xs:annotation>
                  <xs:complexType>
                    <xs:sequence maxOccurs="unbounded">
                      <xs:element name="dataType" type="identifier">
                        <xs:annotation>
                          <xs:documentation>The resource data type
maintained by the Data Bus (i.e. monitor, camera, dms).</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

        </xs:sequence>
        <xs:attribute name="providerName" type="identifier"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="statusDataTypes">
    <xs:annotation>
        <xs:documentation>Stores the status resource data types that the
Data Bus will attempt to store in separate DOM trees.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="dataType">
                <xs:annotation>
                    <xs:documentation>The resource data type maintained by the
Data Bus (i.e. monitor, camera, dms).</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>
```

A.9 statusResp.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:include schemaLocation="../../objects/providerStatus.xsd"/>
  <xs:element name="statusResp">
    <xs:annotation>
      <xs:documentation>Response received for a status request</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ResponseType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="statusData">
    <xs:annotation>
      <xs:documentation>The response data for a status request</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="responseData">
        <xs:all minOccurs="0" maxOccurs="unbounded">
          <xs:element name="statusInfo" type="providerStatus"/>
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

A.10 subscribeResp.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:element name="subscribeResp">
    <xs:annotation>
      <xs:documentation>Response received for a subscribe
request</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="ResponseType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="subscribeData">
    <xs:annotation>
      <xs:documentation>The response data for a subscribe
request</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="responseData">
        <xs:sequence maxOccurs="unbounded">
          <xs:element name="requestedData">
            <xs:annotation>
              <xs:documentation>The data types requested by the
client</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="status" type="subscriptionStatus"
use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="subscriptionStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="successful"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```
<xs:enumeration value="unknownType"/>  
</xs:restriction>  
</xs:simpleType>  
</xs:schema>
```

A.11 providerDisconnectMsg.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:element name="providerDisconnectMsg">
    <xs:annotation>
      <xs:documentation>Message sent to Data Bus clients when a provider subsystem
has disconnected from the Data Bus.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="MessageType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.12 providerReconnectMsg.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:element name="providerReconnectMsg">
    <xs:annotation>
      <xs:documentation>Message sent to Data Bus clients when the Data Bus has
reconnected with the specified provider subsystem.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="MessageType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```


A.13 statusUpdateMsg.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="../../common/objects/transaction.xsd"/>
  <xs:include schemaLocation="../../common/objects/common.xsd"/>
  <xs:include schemaLocation="../objects/providerStatus.xsd"/>
  <xs:include schemaLocation="../objects/deletionStatus.xsd"/>
  <xs:element name="statusUpdateMsg">
    <xs:annotation>
      <xs:documentation>Status update sent to subscribed clients</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="MessageType">
          <xs:sequence>
            <xs:element name="statusUpdateData">
              <xs:annotation>
                <xs:documentation>The status update data</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:choice>
                  <xs:element name="statusDeletedInfo" type="deletionStatus"/>
                  <xs:element name="statusUpdateInfo" type="providerStatus"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>The updated status
information</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```