

SunGuide™:

Message Arbitration System Interface Control Document

SunGuide-MAS-ICD-6.2



Prepared for:

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

March 15, 2016

Document Control Panel			
File Name:	SunGuide-MAS-ICD-6.2.doc		
File Location:	SunGuide CM Repository		
CDRL:	6-1		
	Name	Initial	Date
Created By:	Lynne Randolph, SwRI	LAR	05/03/05
Reviewed By:	Steve Dellenback, SwRI	SWD	05/05/05
	Steve Novosad, SwRI	SEN	05/06/05
	Steve Dellenback, SwRI	SWD	11/15/05
	Steve Dellenback, SwRI	SWD	10/16/07
	Steve Dellenback, SwRI	SWD	11/14/07
Modified By:	Lynne Randolph, SwRI	LAR	11/17/05
	Brandon Meiners, SwRI	BM	10/3/07
	Meredith Moczygemba, SwRI	MRM	10/3/07
	Steve Dellenback, SwRI	SWD	11/14/07
	Adam Hoffman, SwRI	AGH	3/15/16
Completed By:			

Table of Contents

1.0.2	v
November 17, 2005.	v
Added new removeEvent request/response.....	v
3.1.1v	
October 16, 2007	v
Updated for Release 3	v
3.1.2v	
November 14, 2007.	v
Added “how to use this document” section.....	v
1. Scope	1
1.1 Document Identification	1
1.2 Project Overview.....	1
1.3 How to Use This Document	2
1.4 Related Documents	2
1.5 Contacts	3
2. Data	4
2.1 Schema	4
2.1.1 Subsystem communication	5
2.1.2 Device subsystem communication	6
2.2 Examples	7
2.3 Subsystem Schemas	9
3. Notes	12

List of Figures

Figure 1.1 - High-Level Architectural Concept.....	1
Figure 1-2 - SunGuide Developer Documentation.....	2
Figure 2.1 - Sample 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
MAS	Message Arbitration System
SwRI	Southwest Research Institute
TMC	Traffic Management Center
VS	Video Switching
VW	Video Wall
XML	Extensible Markup Language

REVISION HISTORY

Revision	Date	Changes
1.0.0	May 12, 2005	Initial Release
1.0.2	November 17, 2005	Added new removeEvent request/response.
3.0.0	October 16, 2007	Updated for Release 3
3.0.1	November 14, 2007	Added “how to use this document” section
6.2	March 15, 2016	Updated for release 6.2

1. Scope

1.1 Document Identification

This Interface Control Document (ICD) describes the interface between individual SunGuide™ clients and the Message Arbitration System (MAS) subsystem and between the MAS subsystem and the associated drivers. The general base architecture of the XML communications including connection information, byte order and base transaction classes is delineated in the general ICD. 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 document for details regarding data transfer.

1.2 Project Overview

The Florida Department of Transportation (FDOT) is conducting a program that is developing SunGuide software. The SunGuide 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 SunGuide software is to have a common software base that can be deployed throughout the state of Florida. The SunGuide software development effort is based on ITS software available from the state of Texas; 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:

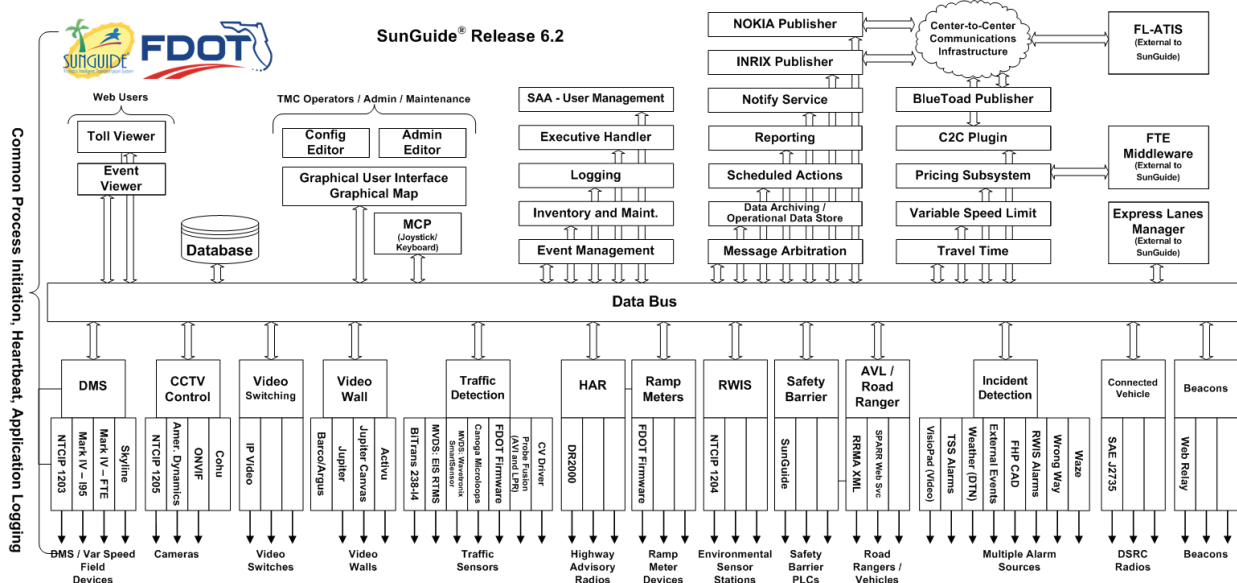


Figure 1.1 - High-Level Architectural Concept

1.3 How to Use This Document

The ICDs describe the specific interface between two SunGuide subsystems or between a SunGuide subsystem and a SunGuide driver. The relationship of appropriate documents is shown in the Figure 1-2.

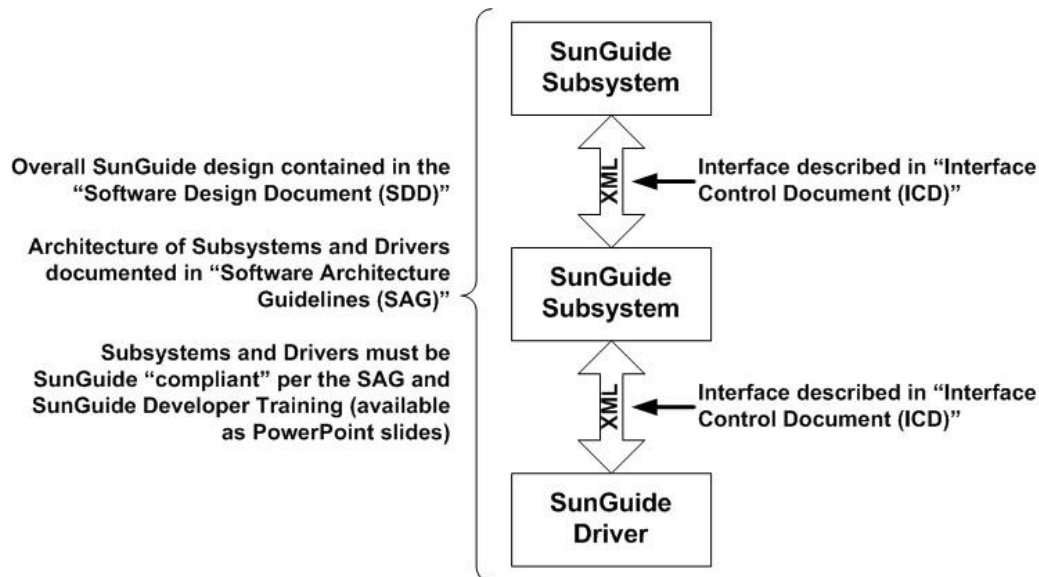


Figure 1-2 - SunGuide Developer Documentation

This document describes an *internal* SunGuide interface. The interface described is between two SunGuide compliant processes. The reader should review the following document to gain an understanding of how SunGuide compliant application is created (this will vary if the application is a driver or subsystem):

SunGuide Software Architecture Guidelines (SAG)

The SAG describes what needs to be included in a SunGuide application to assure that it will work cooperatively in the SunGuide environment. Once the SAG is reviewed, the following document should be reviewed:

SunGuide Software Design Document (SDD)

The SDD will provide an understanding of how individual components of SunGuide were designed. Finally the ICD, along with the associated schema should be reviewed to determine what data needs to be exchanged on the interface being defined in this document.

Additionally, a SunGuide "Developer Training" class is available that provides the students with an introduction into developing SunGuide processes. The SunGuide source code repository has a generic subsystem and a generic driver available that can be used as the basis for developing a new application.

1.4 Related Documents

The following documents were used to develop this document:

- FDOT Scope of Services: *BDQ69, Standard Written Agreement for SunGuide Software Support, Maintenance, and Development, Exhibit A: Scope of Services*. July 1, 2010.
- Notice to Proceed: Letter to Southwest Research Institute® (SwRI®) for BDQ69, July 1, 2010.
- SunGuide Project website: <http://sunguidesoftware.com>.

1.5 Contacts

The following are contact persons for the SunGuide software project:

- Fred Heery, ITS Section, Traffic Engineering and Operations Office Central Office, fred.heery@dot.state.fl.us, 850-410-5606
- Derek Vollmer, ITS Section, Traffic Engineering and Operations Office Central Office, Derek.Vollmer@dot.state.fl.us, 850-410-5615
- Clay Packard, Atkins Project Manager, clay.packard@dot.state.fl.us, 850-410-5623
- David Chang, Atkins Project Advisor, david.chang@dot.state.fl.us, 850-410-5622
- Tucker Brown, SwRI Project Manager, tbrown@swri.com, 210-522-3035
- Roger Strain, SwRI Software Project Manager, rstrain@swri.org, 210-522-6295

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 Schemas directory. The objects directory contains common data schemas that are used by the various requests, messages, and responses.

mas

- messages
 - queueChangedMsg.xsd
 - queueFailureMsg.xsd
 - queueStateChgMsg.xsd
 - queueUpdateMsg.xsd
 - removeOwnerMsg.xsd
- objects
 - message.xsd
 - queue.xsd
- requests
 - addMsgReq.xsd
 - blankQueueReq.xsd
 - changeMsgPriorityReq.xsd
 - getQueueMsgReq.xsd
 - getQueueReq.xsd
 - mergeMsgReq.xsd
 - modifyMsgReq.xsd
 - removeEventReq.xsd
 - removeMsgReq.xsd
 - resendMsgReq.xsd
 - retrieveDataReq.xsd
 - subscribeReq.xsd
 - unmergeMsgReq.xsd
- responses
 - addMsgResp.xsd
 - blankQueueResp.xsd
 - changeMsgPriorityResp.xsd
 - getQueueMsgResp.xsd
 - getQueueResp.xsd
 - mergeMsgResp.xsd
 - modifyMsgResp.xsd
 - removeEventResp.xsd
 - removeMsgResp.xsd
 - resendMsgResp.xsd
 - retrieveDataResp.xsd

- subscribeResp.xsd
- unmergedMsgResp.xsd

Requests may be sent from a client to a subsystem or from a subsystem to a driver. Responses may be sent from a driver to a subsystem or a subsystem to a client. A message can be sent from any process to another process.

2.1.1 Subsystem communication

Initial communication to a subsystem is described in the general ICD. For MAS, the list of queues in the system can be retrieved from the database on startup. Once a client has initiated the connection to MAS, the following requests may be made:

- Messages may be added to queues whether they exist on startup or not,
- Existing messages may be modified or removed,
- Two unmerge messages in a queue may be merged together,
- Two merged messages in a queue may be unmerged,
- A queue may be blanked, or
- A queue or the list of queues may be retrieve.

The following table shows the various subscriptions a client may request. The last column shows the XML updates that will be received if a client has subscribed to this data.

Subscription	Description	Updates Received
queueData	Receive the entire updated queue.	queueChangedMsg
queueNotice	Receive notification that a queue has been updated.	queueUpdateMsg
queueState	Receive notification that the status of a queue has changed.	queueStateChgMsg
queueFailure	Receive notification that a device queue message has failed.	queueFailureMsg
userData	Receive notification that user permissions have been modified.	updateSystemDataMsg

2.1.2 Device subsystem communication

MAS must communicate with other subsystems to send the appropriate device requests. Upon startup, MAS uses data in the config file to establish a connection to the subsystem and determine which requests should be sent.

```

<drivers>
  <driver>
    <identifier>dms</identifier>
    <host>lovelace</host>
    <port>8009</port>
    <resourceType>dms</resourceType>
    <providerType>dms</providerType>
    <msgClassName>multiMsg</msgClassName>
    <package>gov.its.mas.xml.dms</package>
    <sendReqClassName>sendMsgReq</sendReqClassName>
    <sendRespClassName>sendMsgResp</sendRespClassName>
    <terminateReqClassName>terminateMsgReq</terminateReqClassName>
    <terminateRespClassName>terminateMsgResp</terminateRespClassName>
    <username>MasUser</username>
    <password>encryptedPassword</password>
    <pageSeperator>[np]</pageSeperator>
  </driver>
</drivers>

```

The following table shows the correspondence between the config file items and their usage in MAS.

Config item	Description	Usage in MAS
identifier	Name of the subsystem to which MAS must connect.	Used for the providerName in requests to the subsystem.
host	Host for connecting to this subsystem, most often this is the host for Data Bus.	Used in initial connection and reconnection to the subsystem.
port	Port for connecting to this subsystem, most often this is the port for Data Bus.	Used in initial connection and reconnection to the subsystem.
resourceType	Name of the type of resource or device.	Used for the resourceType in requests to the subsystem.
providerType	Type of provider this subsystem is.	Used for the providerType in requests to the subsystem.

Config item	Description	Usage in MAS
msgClassName	The class in MAS that will be created for a message being sent to this subsystem.	For any subsystem, this class must exist in MAS in the package (see below).
package	The package in MAS where the messages, send and terminate requests are placed.	Used to create the appropriate class.
sendReqClassName	The class in MAS that will be created for a send request being sent to this subsystem.	For any subsystem, this class must exist in MAS in the package (see above).
sendRespClassName	The class in MAS that will be created for a send response received from this subsystem.	For any subsystem, this class must exist in MAS in the package (see above).
terminateReqClassName	The class in MAS that will be created for a terminate request being sent to this subsystem.	For any subsystem, this class must exist in MAS in the package (see above).
terminateRespClassName	The class in MAS that will be created for a terminate response received from this subsystem.	For any subsystem, this class must exist in MAS in the package (see above).
username	User name to be used when authenticating. Must be a valid user in the database with appropriate permissions for the subsystem.	Used to authenticate to the subsystem.
password	Encrypted password for the username above.	Used to authenticate to the subsystem.
pageSeperator	The page separator character for this driver.	Used for determining the new page character used when merging two messages in MAS.

2.2 Examples

For example, if a client wishes to add a message to a device, the client sends an addMsgReq to the subsystem. Since MAS has no knowledge of what devices are in the system, an addMsgReq should succeed provided the message conforms to the schema. The highest priority message is sent to the appropriate subsystem for display on the device.

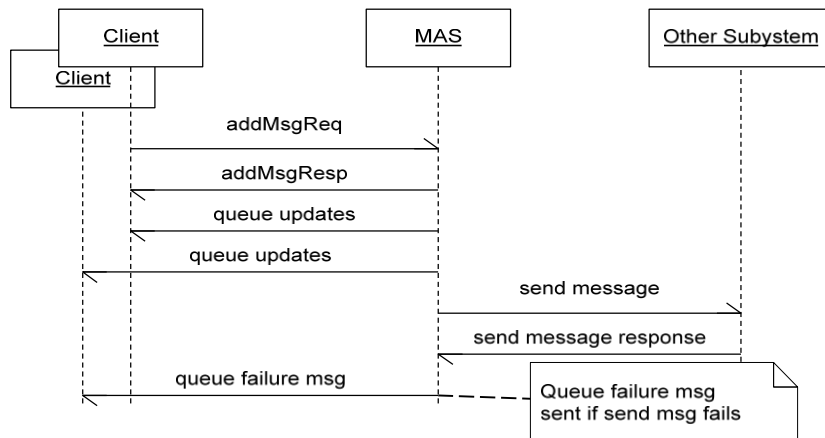


Figure 2.1 - Sample Transaction

The tables below show which requests can be sent from client to subsystem. Requests and responses that are sent to and received from other subsystems for device control will depend upon the other subsystems. The responses sent from subsystem to client are also specified. Messages are sent instead of requests when a response is not required.

2.3 Subsystem Schemas

FC (From client), TC (To client), TD (To driver), FD (From driver)

Usage Description	Requests	FC	Responses	TC	Messages	FC	TC
Used to add a new message to the system.	addMsgReq	X	addMsgResp	X			
Used to remove all messages from a device queue. This results in the device's display being blanked as well.	blankQueueReq	X	blankQueueResp	X			
Used to modify the priority of an existing message.	changeMsgPriorityReq	X	changeMsgPriorityResp	X			
Used to retrieve the full text of a queue message.	getQueueMsgReq	X	getQueueMsgResp	X			
Used to retrieve the queue for a device.	getQueueReq	X	getQueueResp	X			
Used to merge two messages in a device's queue.	mergeMsgReq	X	mergeMsgResp	X			
Used to modify the text of an existing message.	modifyMsgReq	X	modifyMsgResp	X			

Usage Description	Requests	FC	Responses	TC	Messages	FC	TC
Used to remove all messages for a particular event id.	removeEventReq	X	removeEventResp	X			
Used to remove a message from the queue.	removeMsgReq	X	removeMsgResp	X			
Used to resend the message for a device queue. If the queue is blank, this results in a terminate being sent to the appropriate subsystem.	resendMsgReq	X	resendMsgResp	X			
Used to subscribe to updates from the system. Data includes device status, device configuration data, and users.	retrieveDataReq	X	retrieveDataResp	X			
Used to subscribe to updates from the system.	subscribeReq	X	subscribeResp	X			
Used to unmerge two messages in a device's queue.	unmergeMsgReq	X	unmergeMsgResp	X			

Usage Description	Requests	FC	Responses	TC	Messages	FC	TC
Used to send the updated queue to any clients who have subscribed.					queueChangedMsg		X
Used to send a failure notification to the client when a send to device has failed.					queueFailureMsg		X
Used to send an update of the queue's state (failed, completed, pending).					queueStateChgMsg		X
Used to inform a client that a queue has changed (does not contain the changed queue).					queueUpdateMsg		X
Removes all messages owned by a particular owner.					removeOwnerMsg	X	

3. Notes

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