

Interface Description Document (IDD)

Range Mission Tool (RMT) Version 4.0 Interfaces

Prepared for:
Pacific Missile Range Facility (PMRF)
Kauai, HI
SOL-02-012
1 March 2002

Prepared by:



SOLIPSYS CORPORATION
6100 CHEVY CHASE DRIVE SUITE 200 LAUREL, MD 20707-2929
301/483-8900 VOICE 301/483-8901 FAX
www.solipsys.com

Table of Contents

1.0	INTRODUCTION	1
2.0	INTERNAL INTERFACES	3
2.1	TDF	3
2.2	SOLINET.....	5
2.3	RMT SERVER - RMT GUI MESSAGING INTERFACE.....	5
2.3.1	<i>Messages Sent from Server to GUI.....</i>	<i>5</i>
2.3.2	<i>Messages Sent from GUI to Server.....</i>	<i>9</i>
2.3.3	<i>Message Formats.....</i>	<i>12</i>
2.3.3.1	DxControlMsg.....	12
2.3.3.2	DxControlResponse.....	13
2.3.3.3	GUICertifiedPassword.....	13
2.3.3.4	GUIConvertTextModel.....	13
2.3.3.5	GUIRequestList	14
2.3.3.6	GUIRequestObject	14
2.3.3.7	GUIRequestVehicleReport.....	14
2.3.3.8	GUIRequestSNvsTime.....	15
2.3.3.9	GUIScenarioControl.....	15
2.3.3.10	GUIScenarioObject	16
2.3.3.11	GUISensor	17
2.3.3.12	GUISensorInformationTableRequest	17
2.3.3.13	GUISensorModeAssignment.....	17
2.3.3.14	GUISensorModel	18
2.3.3.15	GUITicInterval.....	19
2.3.3.16	GUITicReferencePoint	19
2.3.3.17	GUIVehicle.....	19
2.3.3.18	GUIVehicleAnnotation.....	20
2.3.3.19	GUIVehicleModel.....	20
2.3.3.20	GUIVehicleModifyModel.....	21
2.3.3.21	GUIVehicleStartTime	21
2.3.3.22	GUIVehicleSynch.....	21
2.3.3.23	GUIWaypointPosition.....	22
2.3.3.24	SrvCertifiedPasswordResponse.....	22
2.3.3.25	SrvDataSetConversionCompleted	23
2.3.3.26	SrvDisplayTrackState	23
2.3.3.27	SrvListResponse	23
2.3.3.28	SrvLiveTrackUpdate.....	23
2.3.3.29	SrvPlannedPathDump	24
2.3.3.30	SrvPlannedTrackState.....	24
2.3.3.31	SrvScenarioControlResponse.....	24

2.3.3.32	SrvScenarioObjectResponse.....	25
2.3.3.33	SrvScenarioSynch.....	25
2.3.3.34	SrvSensorInformationTableDump.....	25
2.3.3.35	SrvSNvsTimeResponse.....	25
2.3.3.36	SrvSNError.....	25
2.3.3.37	SrvTicStatus.....	26
2.3.3.38	SrvTimeTic.....	26
2.3.3.39	SrvVehicleAnnotation.....	26
2.3.3.40	SrvVehicleReportComplete.....	26
2.3.3.41	SrvVehicleSynch.....	27
2.3.3.42	SrvWaypointError.....	27
2.3.3.43	SrvWaypointPath.....	27
2.3.3.44	SrvWaypointPosition.....	27
3.0	EXTERNAL SYSTEM INTERFACES	30
3.1	DATA FUSION MESSAGING INTERFACE.....	30
3.1.1	<i>ContactUpdate</i>	31
3.1.2	<i>DropTrack</i>	31
3.1.3	<i>TrackUpdateBuffer</i>	31
3.1.4	<i>SrvBearingLine</i>	32
3.1.5	<i>SrvOwnershipReferencePos</i>	32
3.2	SIMULATION MESSAGING INTERFACE.....	33
3.2.1	<i>Track Simulation Using iNet Messages</i>	33
3.2.1.1	<i>iNet Track Data Message (MT 25)</i>	33
3.2.1.2	<i>iNet Track Data Message (MT 26)</i>	35
3.2.2	<i>Track Simulation using DIS Messages</i>	36
3.2.3	<i>Detection Simulation using DTE Messages</i>	36
4.0	FILE INTERFACES.....	37
4.1	GOG FILE.....	37
4.2	NOMINAL DATA SET.....	38
4.3	COMMON ROUTE DEFINITION (CRD) FILE.....	39
5.0	COTS PRODUCTS.....	41
5.1	JAVA SDK.....	41
6.0	APPENDIX A – SAMPLE GOG FILE.....	42
7.0	APPENDIX B – CRD DTD.....	47
8.0	APPENDIX C – SAMPLE CRD FILE.....	54

List of Figures

FIGURE 1-1 RMT DISTRIBUTED SYSTEM ARCHITECTURE AT PMRF2
FIGURE 1-2 ESPRIT/RMT RELATIONSHIP DIAGRAM5

List of Tables

TABLE 2-1 SERVER TO GUI MESSAGES	6
TABLE 2-2 GUI TO SERVER MESSAGES	10
TABLE 3-1 DATA FUSION TO RMT GUI MESSAGES.....	30
TABLE 3-2. INET MESSAGE LAYOUT.....	33
TABLE 3-3. FORMAT FOR INET TRACK DATA MESSAGE 25.....	33
TABLE 3-4. FIELD DEFINITIONS FOR INET TRACK DATA MESSAGE 25.....	34
TABLE 3-5. FIELD DEFINITIONS FOR INET TRACK DATA MESSAGE 26.....	35
TABLE 4-1. "DUCK" TBM DATA SET FORMAT	39

1.0 Introduction

This document defines the interfaces for the Pacific Missile Range Facility's (PMRF) implementation of the Exercise Scenario Real-time Integrated Test (ESPRIT) Tool. The PMRF implementation is referred to as the Range Mission Tool (RMT). The RMT system supports scenario planning, scenario preview, event rehearsal, and real-time event monitoring for missions taking place at PMRF.

RMT is designed as a message-based distributed system. The system consists of one or more RMT clients connected via a TCP network to a single ESPRIT server by an assigned port number. The RMT server, hosted on a system running Solaris or LINUX, performs the bulk of the processing as well as providing storage for scenarios, models, sensors, vehicles, etc. During planning, the RMT Graphical User Interface (GUI) clients send messages to the server requesting that it store or download scenario information via the display network. During scenario preview, the server interprets the scenario and sends messages containing planned vehicle positions versus time to the client for display. The network is also monitored for live track information sent to the RMT GUI for display from a Data Fusion application performing track correlation for all of the sensors at PMRF. During scenario rehearsal, the server can generate simulated radar detection and tracking messages to drive the client displays. depicts the RMT architecture and the RMT's external interfaces at PMRF.

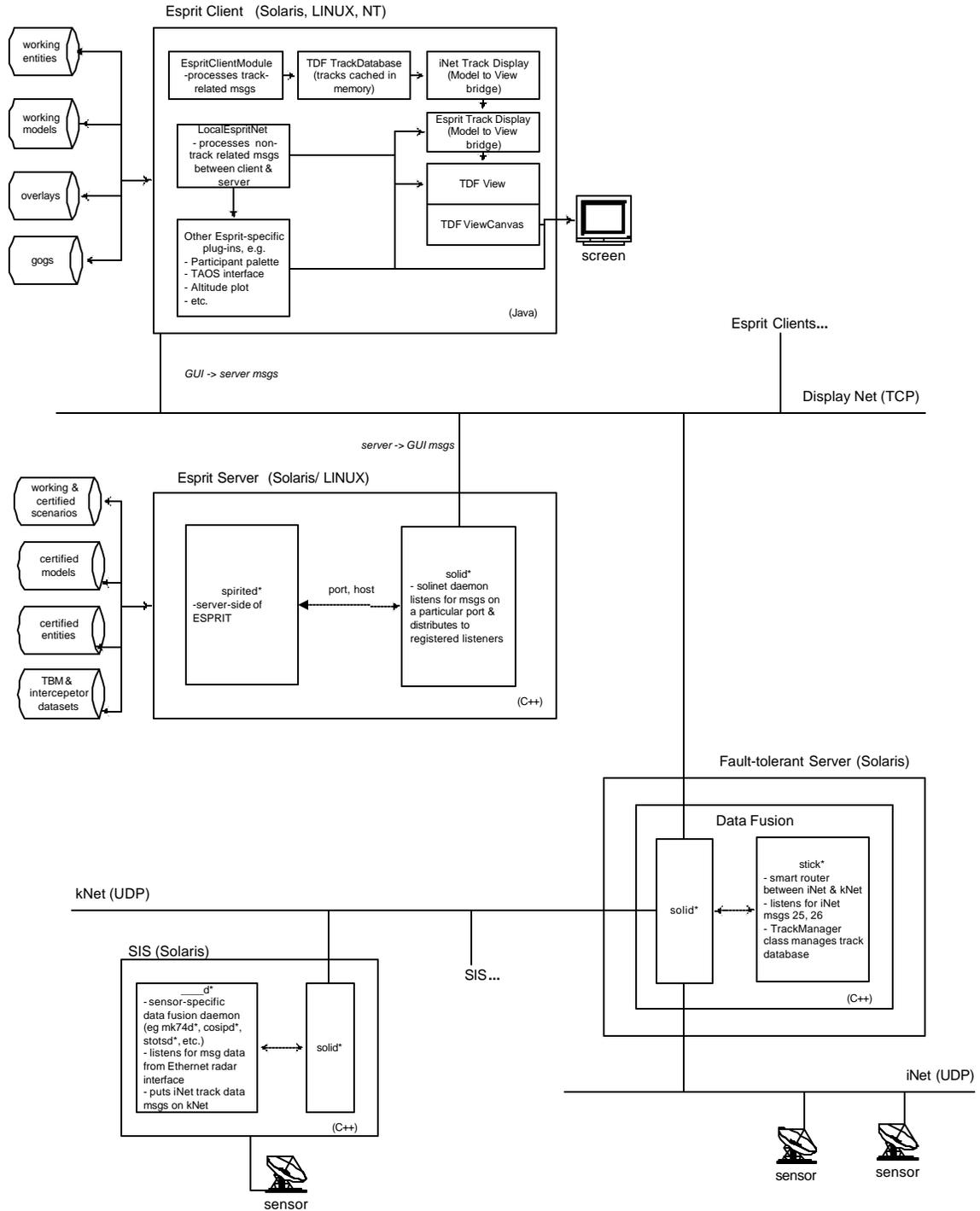


Figure 1-1. ESPRITRMT Distributed System Architecture at PMRF

2.0 Internal Interfaces

RMT's internal interfaces include its embedded use of other Solipsys Corporation software, and the messages that are exchanged between the RMT GUI and the RMT server.

2.1 TDF

The ESPRIT/RMT GUI is built on the foundation of the Solipsys Corporation's Tactical Display Framework (TDF), a visualization tool for the display of tactical Command and Control information. Version 4.0 of RMT relies upon Version 3.4 of TDF.

To promote portability and reusability, TDF is written in the Java programming language. TDF provides a reusable and extensible object-oriented display framework. Its dynamic loader mechanism can load Java classes into the TDF without prior knowledge of their existence. This allows application-specific functionality to be decoupled from the TDF core, allowing functionality to be extended or overridden, and changes and features to be added without ever touching the core TDF components.

TDF is distributed as a set of java archive files - .jar files:

- sol.jar contains the essential framework for applications (color management, views, menus, etc.).
- geo.jar contains a collection of extensive classes that provide support for displaying imagery, projections, coordinates, terrain, etc. using precise WGS84-compliant two- and three-dimensional representations of geographic and vehicular data. The software supports most map and imagery database formats, including WVS, ADRG, CADRG, CIB, and DTED.
- tdf.jar contains the code related to tactical display features and includes a basic set of plug-ins to do track display and playback, history, hooks, details, tags, etc.

Also written in Java, the RMT GUI builds on the TDF object hierarchy, with classes that extend from TDF classes or implement TDF interfaces. Functionality unique to RMT is provided by means of plug-ins that are dynamically loaded into the TDF framework. Figure 1-2 depicts the relationship between the RMT GUI and the TDF hierarchy.

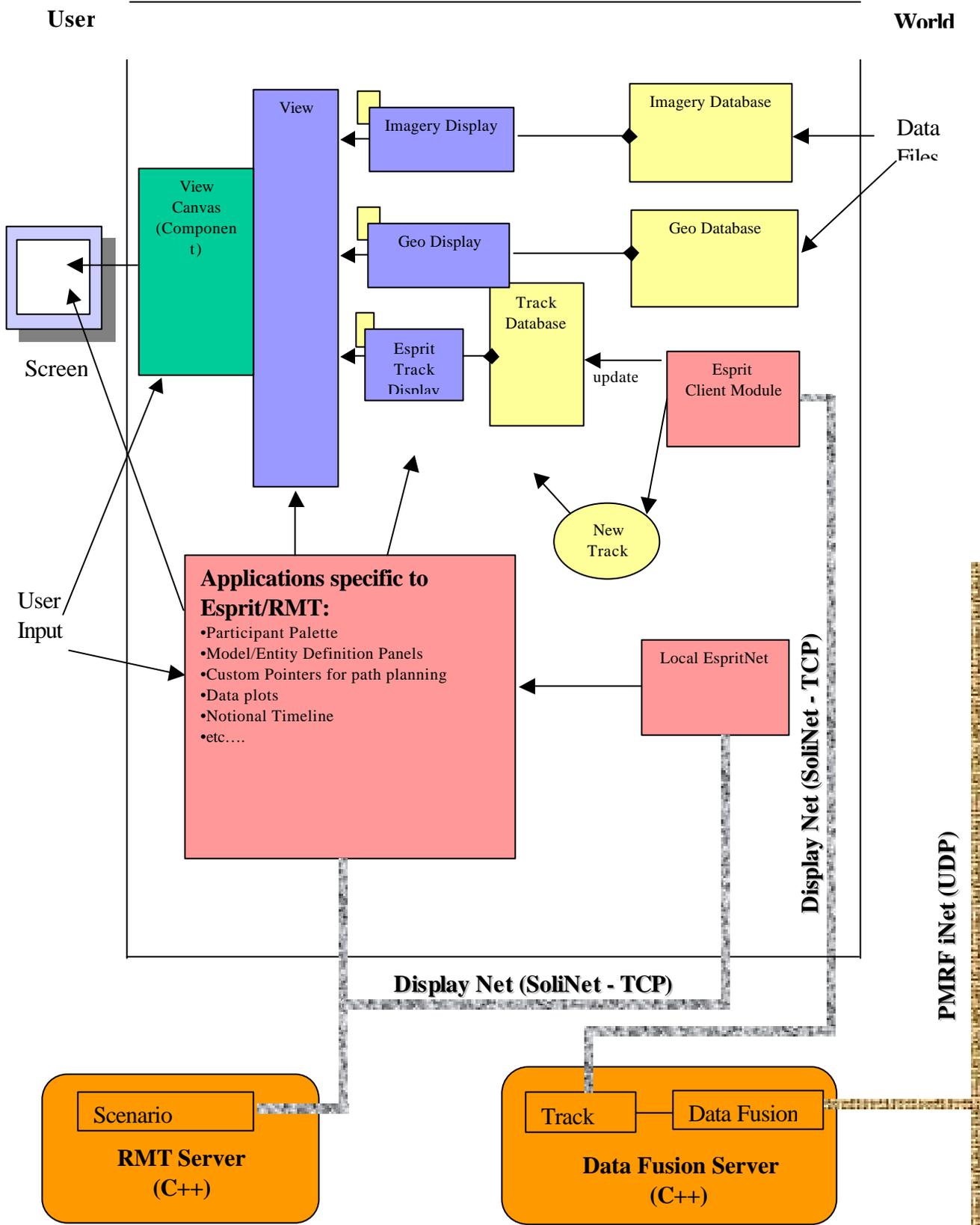


Figure 1-2. ESPRIT/RMT Relationship Diagram

2.2 SoliNet

RMT relies on Solipsys Corporation's SoliNet product to provide the message delivery system between the RMT server and the RMT/TDF displays on the client. SoliNet's message-passing model provides duplexed, multi-threaded client-server communication. The SoliNet "Solid" process runs as a daemon on the RMT server, listening for messages and distributing them to registered listeners. Configuration files specify the port and the host on which communications is to take place.

2.3 RMT Server - RMT GUI Messaging Interface

RMT's distributed system architecture relies on the exchange of messages between the RMT server, known as "spirited", and the RMT GUI running on the client. The SoliNet toolkit provides support for defining these messages using a meta-language approach. The Metagen tool parses the ASCII Meta files corresponding to each message and generates source code that supports fully functional message object classes accessible from both the C++ server application and the Java application that comprises the RMT client. The Cajun tool can be used to parse Meta files that describe generic classes that are to be used as the building blocks for messages. If message formats change, only the Meta files need be changed; running Cajun and Metagen to regenerate the necessary C++ and Java source code effects code changes.

The following sections define the messages used by the RMT communications layer. The messages are grouped into two categories: those sent from the server to the client to communicate display information, and those sent from the client to the server to request information that is stored on the server or to send updates back to the server when the user makes changes via the GUI.

2.3.1 Messages Sent from Server to GUI

The following bullets summarize the messages sent from the RMT server to the GUI running on the client.

- DxControlResponse
- GUIScenarioObject
- GUISensor
- GUISensorModel
- GUISensorModeAssignment
- GUIVehicle
- GUIVehicleModel

- SrvCertifiedPasswordResponse
- SrvDataSetConversionComplete
- SrvDisplayTrackState
- SrvListResponse
- SrvLiveTrackUpdate
- SrvPlannedPathDump
- SrvPlannedTrackState
- SrvScenarioControlResponse
- SrvScenarioObjectResponse
- SrvScenarioSynch
- SrvSensorInformationTableDump
- SrvSNvsTimeResponse
- SrvSNErrors
- SrvTicStatus
- SrvTimeTic
- SrvVehicleAnnotation
- SrvVehicleReportCompleted
- SrvVehicleSynch
- SrvWaypointError
- SrvWaypointPath
- SrvWaypointPosition

Figure2-1 describes each message sent from the RMT server to the GUI running on the client, noting where it is used in the code on the server and in the GUI.

Figure2-1. Server to GUI Messages

Message	Description	Sent from Server	Processed in GUI
DxControlResponse	Sent by the server in response to a DxControlMsg indicating the status of the Dx start/stop request	?	ScenarioListener. processSISDxResponse
GUIScenarioObject	Sent by the server to load a scenario object stored on the server (augmentation, overlay, vehicle model, sensor model, sensor entity)	Scenario	ScenarioListener. processScenarioObjectMsg, ListMsgListener.processGui ScenarioObject
GUISensor	Sent by the server when a scenario is loaded to communicate the scenario's sensor entity information to the	Sensor, SensorInfo, SensorModel	ScenarioListener. processSensorEntityMsg

Message	Description	Sent from Server	Processed in GUI
	client		
GUISensorModel	Sent by the server when a scenario is loaded to communicate the scenario's sensor model information to the client	Sensor, SensorModel	ScenarioListener. processSensorModelMsg
GUISensorModeAssignment	Sent by the server when a scenario is loaded to convey current sensor mode assignments for a particular sensor.	Sensor, SensorMgr	ScenarioListener. ProcessSensorModeAssignmentMsg
GUIVehicle	Sent by the server when a scenario is loaded to communicate the scenario's vehicle entity information to the client	Kine, Scenario, SensorMgr, Vehicle, VehicleMgr	ScenarioListener. processVehicleEntityMsg
GUIVehicleModel	Sent by the server when a scenario is loaded to communicate the scenario's vehicle model information to the client	Vehicle, VehicleModel	ScenarioListener. processVehicleModelMsg
SrvCertifiedPasswordResponse	Sends success/failure status in response to a GuiCertifiedPassword	Scenario	Certification. processCertifiedPasswordResponse
SrvDataSetConversionComplete	Sent by server to indicate success/failure of a GuiConvertTextModel request	GeneralModelData	ScenarioListener. processDataSetConversionCompleted
SrvDisplayTrackState	Sent by the server to communicate track data when in local execution mode	Sensor	EspritClientModule.DisplayTrackState. processMessage
SrvListResponse	Sent by the server in response to a GuiRequestList request; contains success/failure status and requested list data if request was successful	Scenario	ListMsgListener. processSrvListResponse
SrvPlannedPathDump	Sent by server to describe planned path of a vehicle. Typically this message is sent at scenario load to describe the entire path of a TBM.	ModelData	ScenarioListener. processPlannedPathDumpMsg
SrvPlannedTrackState	Sent by server to display a track's planned position during scenario preview mode.	Vehicle	EspritClientModule. PlannedTrackState. processMessage

Message	Description	Sent from Server	Processed in GUI
SrvScenarioControlResponse	Sent by the server in response to a GuiScenarioControl message. Hostname field indicates whether the scenario load messages that follow will be loading scenario information for this client or all clients. Type field indicates type of scenario action that was requested -- create, load, save, or delete a scenario (working or certified); control scenario preview or local execution mode; set simulation type or detection type during local execution mode; query, enable or disable live data; set the clock rate -- so that the client can perform the appropriate initializations, such as clearing the previous scenario or clearing the track database, and fire the appropriate scenario control event	Scenario	ScenarioListener. processScenarioControlResponse, fireScenarioControlEvent
SrvScenarioObjectResponse	Sends success/failure status in response to previous GuiScenarioObject request.	Scenario	ScenarioListener, processScenarioObjectResponse
SrvScenarioSynch	Tells the GUI to synch a scenario by setting the scenario T0 and preview begin times.	Scenario	ScenarioListener. processScenarioSynch
SrvSensorInformationTableDump	Sent by the server in response to a GuiSensorInfoTableRequest. Contains pertinent model and sensor-specific data for each sensor specified as part of the scenario.	SensorMgr, SensoInfoList	*processing disabled! (code in ListMsgListener. ProcessSensorInfoTableDump is commented out)
SrvSNvsTimeResponse	Sent by the server in response to a GuiRequestSNvsTime message	SensorMgr	ScenarioListener. processSNMessage
SrvSNErrror	Sent by the server when an error was encountered while processing a GuiRequestSNvsTime message for a particular vehicle	SensorMgr	ScenarioListener. processSNErrror
SrvTicStatus	Sent by the server when a scenario is loaded to indicate the time tic information for a vehicle	VehicleMgr	ScenarioListener. processTicStatus
SrvTimeTic	Sent by the server when a scenario is loaded to indicate the time tic interval for a vehicle-waypoint pair.	VehicleMgr	ScenarioListener. processTimeTicMsg
SrvVehicleAnnotation	Sent by the server to specify an annotation for a vehicle waypoint.	KineStateGeneral	ScenarioListener. processVehicleAnnotationMsg
SrvVehicleReportComplete	Sent by server in response to a	GeneralModel,	ScenarioListener.

Message	Description	Sent from Server	Processed in GUI
	GuiRequestVehicleReport to indicate success/failure of the report request	Sensor, Vehicle, WaypointMgr	processVehicleReportCompleted
SrvVehicleSynch	Sent by the server during local execution mode to provide a list of vehicle/waypoint pairs that are to be synchronized in time.	VehicleMgr	ScenarioListener. processVehicleSynchMsg
SrvWaypointError	Sent by the server to specify an error message related to a waypoint entered in the GUI (via GUIWaypointPosition message); GUI responds by deleting the waypoint and alerting the user.	Kine, WaypointManager	ScenarioListener. processWaypointErrorMsg
SrvWaypointPath	Sent by the server to describe the planned path for a vehicles planned using waypoints	Waypoint	ScenarioListener. processPathMsg
SrvWaypointPosition	Sent by the server to tell the GUI to add a waypoint for a vehicle	Waypoint	ScenarioListener. processWaypointMsg

2.3.2 Messages Sent from GUI to Server

The following bullets summarize the messages sent from the GUI running the client to the RMT server.

- DxControlMsg
- GUIScenarioObject
- GUISensor
- GUISensorModel
- GUISensorModeAssignment
- GUIVehicle
- GUIVehicleModel
- GUICertifiedPassword
- GUIConvertTextModel
- GUIRequestList
- GUIRequestObject
- GUIRequestVehicleReport
- GUIRequestSNvsTime
- GUIScenarioControl
- GUISensorInformationTableRequest
- GUISensorModeAssignment
- GUITicInterval

- GUITicReferencePoint
- GUIVehicleAnnotation
- GUIVehicleModifyModel
- GUIVehicleStartTime
- GUIVehicleSynch
- GUIWaypointPosition

Table 2-2 describes each message sent from the RMT server to the client GUI, noting where it is used in the code on the server and in the GUI.

Table 2-2. GUI to Server Messages

Message	Description	Sent from GUI	Processed in Server
DxControlMsg	Tells the server to start or stop playing a Dx	LocalEspritNet. controlSISDx	?
GUIScenarioObject	Tells the server to update or delete a scenario object stored on the server (augmentation, overlay, vehicle model, sensor model, sensor entity)	LocalEspritNet. saveSensorEntity, saveSensorModel, saveVehicleModel deleteSensorEntity deleteSensorModel deleteVehicleModel, controlScenario, controlJavaObjects	Scenario
GISensor	Tells the server to create, update or delete a sensor entity	LocalEspritNet removeSensorEntity, sendSensorEntity	Sensor, SensorInfo, SensorModel
GISensorModel	Tells the server to create, update or delete a sensor model	LocalEspritNet. sendSensorModel	Sensor, SensorModel
GISensorModeAssignment	Tells the server to update sensor mode assignments for the specified sensor	LocalEspritNet. sendSensorEntityAssignment	Sensor, SensorMgr
GUIVehicle	Tells the server to create, update or delete a vehicle entity	LocalEspritNet. sendVehicleEntity, sendTBM, sendInterceptor	Kine, Scenario, SensorMgr, Vehicle, VehicleModel
GUIVehicleModel	Tells the server to create, update or delete a vehicle model	LocalEspritNet. sendVehicleModel	Vehicle, VehicleModel
GUICertifiedPassword	Requests the password used to certify model or resets the certified password	LocalEspritNet. ControlCertifiedPassword	Scenario
GUIConvertTextModel	Tells the server to convert the text model identified by the specified name, interpreting it as the specified model type (e.g.	LocalEspritNet. convertAsciiFile	GeneralModelData, Kine, Vehicle

Message	Description	Sent from GUI	Processed in Server
	TBM or intercept)		
GUIRequestList	Requests from the server a list of scenarios, sensor or vehicle models, sensor entities, or vehicle entities (including TBMs or interceptors) for display in a ListPanel	LocalEspritNet. getNames	Scenario
GUIRequestObject	Requests a certified sensor model, vehicle model, or sensor entity from the server	LocalEspritNet. getScenarioObject	Scenario
GUIRequestVehicleReport	Request that the server generate a vehicle report with waypoint timeline info, time interval info, planned position info, or sensor measurement info	LocalEspritNet. requestVehicleReport	GeneralModel, Kine, Scenario, Sensor, Vehicle, Waypoint
GUIRequestSNvsTime	Requests signal-to-noise over time for the specified vehicle	LocalEspritNet. sendSNRequest	SensorMgr
GUIScenarioControl	Tells the server to create, load, save, or delete a scenario (working or certified); control scenario preview or local execution mode; set simulation type or detection type during local execution mode; query, enable or disable live data; set the clock rate	LocalEspritNet. controlScenario	Scenario
GUISensorInformationTableRequest	Requests the sensor information table from the server	LocalEspritNet. getSITList	SensorMgr
GUITicInterval	Tells the server to set the tic interval and tic annotation interval	LocalEspritNet. controlTicInterval	VehicleMgr
GUITicReferencePoint	Tells the server to set the specified vehicle waypoint to T0 or to a reference point	LocalEspritNet. controlTics	VehicleMgr
GUIVehicleAnnotation	Sends an annotation for a vehicle at the specified waypoint	LocalEspritNet. sendVehicleSynchPoint	Kine, Vehicle
GUIVehicleModifyModel	Tells the server to update the specified vehicle with a new start point, angle, and name	LocalEspritNet. modifyVehicle, modifyTBM	GeneralModelData, Kine, Vehicle
GUIVehicleStartTime	Tells the server to define a start time offset for a vehicle	LocalEspritNet. controlVehicleEntity	VehicleMgr
GUIVehicleSynch	Tells the server to add,	LocalEspritNet.	VehicleMgr

Message	Description	Sent from GUI	Processed in Server
	remove, delete, or synchronize a list of vehicle/waypoint pairs	synchPoints	
GUIWaypointPosition	Tells the server to delete, update, move, or add a waypoint for a vehicle. TDF assigns a waypoint id that is unique over an entire scenario.	LocalEspritNet. fireWaypointPlaced, updateWaypoint, undoWaypoint, deletePathDescriptorPoint	Waypoint

2.3.3 Message Formats

The following sections specify the message formats for each of the messages used within RMT. All messages are defined using Metagen's Meta language syntax. General RMT messages are defined in `EspritMsg.Meta`, Data Extract messages in `DxMsg.Meta`, and PMRF-specific messages in `KnetMsg.meta`.

Unless otherwise noted, the following units are used in the messages:

Angles: radians, -pi to pi
 Distance: meters
 Rates: meters / millisecond
 Velocity: Earth centered X,Y,Z
 Time: LSB is milliseconds

Messages defined in `DxMsg.meta`:

2.3.3.1 DxControlMsg

```

message DxControlMsg tag = 3
  enum Action
    etag Invalid = 0
    etag Off     = 1
    etag On      = 2
  end enum Action

  // Explicitly defining the combinations -- recognize that not every
  // value need be tested for. This will get tedious if we have too
many
  // different types of dxing to do, so don't rely on the interim values.

  enum Mask
    etag Debug          = (1<<0)    // 0001
    etag IO             = (1<<1)    // 0010
    etag Eng            = (1<<2)    // 0100
  end enum Mask

  member Jstring      filename

```

```

    member int          dbMask
    member int          action    // one of the enumerated actions
end message DxControlMsg

```

2.3.3.2 DxControlResponse

```

message DxControlResponse tag = 4
    // Action defined in DxControlMsg
    // Mask defined in DxControlMsg

enum Status
    etag Failure = 0
    etag Success = 1
end enum Status

member Jstring    filename
member u_int      dbMask
member u_int      action
member u_int      status
member Jstring    errorString    // valid if status = Failure
end message DxControlResponse

```

Messages defined in EspritMsg.meta:

2.3.3.3 GUICertifiedPassword

```

message GuiCertifiedPassword tag = 49
    enum Action
        etag Invalid      = 0
        etag Request      = 1
        etag Set          = 2
    end enum Action

    member u_int    action
    member Jstring  oldPassword
    member Jstring  newPassword
    member Jstring  requestedBy
end message GuiCertifiedPassword

```

2.3.3.4 GUIConvertTextModel

```

message GuiConvertTextModel tag = 35
    member ConvertTextModel msgObj
end message GuiConvertTextModel

class ConvertTextModel
    member u_int    kineBehavior
    member Jstring  textModelName
    member Jstring  binaryModelName
end class ConvertTextModel

```

2.3.3.5 GUIRequestList

```
message GuiRequestList tag = 36
  enum Action
    etag Invalid                = 0
    etag WorkingScenario        = 1
    etag CertifiedScenario      = 2
    etag CertifiedSensorModel    = 3
    etag CertifiedVehicleModel  = 4
    etag CertifiedSensorEntity   = 5
    etag CertifiedVehicleEntity  = 6
    etag TBM                    = 7
    etag TBMText                 = 8
    etag Interceptor            = 9
    etag InterceptorText        = 10
    etag TAOStype               = 11
    etag TAOStable              = 12
    etag TAOStableProblem       = 13

  end enum Action
  member u_int action
end message GuiRequestList
```

2.3.3.6 GUIRequestObject

```
message GuiRequestObject tag = 37
  enum Type
    etag Invalid                = 0
    etag SensorModelCertified   = 1
    etag VehicleModelCertified  = 2
    etag SensorEntityCertified  = 3
  end enum Type

  member u_int type
  member Jstring name
end message GuiRequestObject
```

2.3.3.7 GUIRequestVehicleReport

```
message GuiRequestVehicleReport tag = 43
  enum Action
    etag Invalid                = 0
    etag Timeline               = 1
    etag Interval               = 2
    etag PlannedPositions       = 3
    etag SensorMeasurements     = 4
  end enum Action

  member u_int action
  member int VID
  member int SID
  member int timeInterval
```

```

    member Time startTime
    member Time stopTime
    member Jstring filename
end message GuiRequestVehicleReport

```

2.3.3.8 GUIRequestSNvsTime

```

message GuiRequestSNvsTime tag = 45
    member int VID
end message GuiRequestSNvsTime

```

2.3.3.9 GUIScenarioControl

```

// scenarioName is only used for the following actions:
//     LoadScenario, SaveScenario
//
// clockRate is only valid for SetClockRate action.
//     2 => twice real time, 0.5 => half real time, etc.
//     0 => fastest rate possible -- typically used for test purposes
//
// classification is only valid for the SetScenarioClassification
//     action.
//
// LocalExecution and Preview cannot be running simultaneously.
// Pause, Resume, and Stop affect one or the other, depending on
// which is enabled at the current time. Preview and live data can
// be enabled simultaneously.
//
message GuiScenarioControl tag = 23
    enum Action
        etag Invalid = 0
        etag LoadCurrentScenario = 1
        etag LoadWorkingScenario = 2
        etag LoadCertifiedScenario = 3
        etag SaveWorkingScenario = 4
        etag SaveCertifiedScenario = 5
        etag NewScenario = 6
        etag CloseScenario = 7 // clears currently defined
scenario
        etag StartPreview = 8
        etag StartLocalExecution = 9
        etag Pause = 10 // for Preview or
LocalExecution
        etag Resume = 11 // for Preview or
LocalExecution
        etag Stop = 12 // for Preview or
LocalExecution
        etag EnableLiveData = 13
        etag DisableLiveData = 14
        etag DisplayScenario = 15 // for server test purposes
only
        etag SetClockRate = 16

```

```

    etag LiveDataEnabled      = 17    // query: Is Live Data
Enabled?
    etag SetScenarioStart    = 18
    etag DeleteWorkingScenario = 19
    etag DeleteCertifiedScenario= 20
    etag SetSimType          = 21
    etag SetDetectionType    = 22
    etag RequestSimType      = 23
    etag SetScenarioClassification = 24
end enum Action

enum SimType
    etag NO_SIM      = 1
    etag iNet25      = 2
    etag iNet26      = 3
    etag iNetBoth    = 4
    etag DIS         = 5
end enum SimType

enum DetectionType
    etag NO_DETECTION = 1
    etag DTE           = 2
    etag DTE_MSCT     = 3
end enum DetectionType

member u_int      action
member Jstring    name
member float      time
member Jstring    saveCertifiedPassword
member u_int      simType
member u_int      detectionType
member Classification classification
end message GuiScenarioControl

```

2.3.3.10 GUIScenarioObject

```

message GuiScenarioObject tag = 47
enum Type
    etag Invalid      = 0
    etag Augmentation = 1
    etag Overlays     = 2
    etag VehicleModel = 3
    etag SensorModel  = 4
    etag SensorEntity = 5
end enum Type

enum Action
    // etag Invalid = 0 // already defined for GuiScenarioObject
    etag Save       = 1
    etag Delete     = 2
    etag ServerLoad = 3 // used when server is broadcasting to TDF
end enum Action

```

```

    member u_int type
    member u_int action
    member Jstring name
    member Jstring saveCertifiedPassword // only used when saving or
                                         // deleting models and entities

    member SerializableJavaObj object
end message GuiScenarioObject

```

2.3.3.11 GUISensor

```

// If VID is zero, the sensor is stationary, the VID is unused, and
// the position field identifies the position of the sensor.  If VID
// is nonzero, the sensor is on a vehicle and the position field
// is unused.
//
message GuiSensor          tag = 25

    enum GuiSensorAction
        etag Invalid      = 0
        etag NewSensor    = 1
        etag UpdateSensor = 2
        etag DeleteSensor = 3
        etag ServerLoad   = 4 // sent by server when new scenario is
loaded
    end enum GuiSensorAction

    member u_int          action
    member u_int          VID
    member u_int          priority
    member u_int          primarySystemId
    member u_int          secondarySystemId
    member u_int          assignedVid
    member int            SID
    member Jstring        sensorName
    member Geodetic       position
    member Jstring        modelName
    member Jstring        modelMode
    member Jstring        description // e.g., "Makaha Ridge"
    member double         shadowElev[360] // radians, index=azimuth
in deg
end message GuiSensor

```

2.3.3.12 GUISensorInformationTableRequest

```

message GuiSensorInformationTableRequest tag = 30
    member Time time
end message GuiSensorInformationTableRequest

```

2.3.3.13 GUISensorModeAssignment

```

message GuiSensorModeAssignment tag = 50
    member int          SID
    member SensorModeTableEntry sensorModeTable[]

```

```
end message GuiSensorModeAssignment
```

2.3.3.14 GUISensorModel

```
// The name and mode fields together uniquely identify the model
//
message GuiSensorModel    tag = 24

    enum GuiSensorModelAction
        etag Invalid          = 0
        etag NewSensorModel   = 1
        etag UpdateSensorModel = 2
        etag DeleteSensorModel = 3
        etag ServerLoad       = 4    // sent by server when new scenario
is loaded
    end enum GuiSensorModelAction

    enum GuiSensorModelType
//    etag Invalid          = 0    // Previously Defined
        etag Tracking      = 1
        etag Surveillance  = 2
        etag BeaconTransmitter = 3
        etag BeaconReceiver = 4
        etag TelemetryTransmitter = 5
        etag TelemetryReceiver = 6
        etag ITCSTransmitter = 7
        etag ITCSReceiver   = 8
    end enum GuiSensorModelType

    // This enum may not need to be defined, but these
    // are the frequency band ranges per Dave Furst
    enum GuiSensorModelFreqBand
//    etag Invalid          = 0    // Previously Defined
        etag L_Band        = 1
        etag S_Band        = 2
        etag C_Band        = 3
        etag X_Band        = 4
        etag Ku_Band       = 5
        etag K_Band        = 6
        etag Ka_Band       = 7
    end enum GuiSensorModelFreqBand

    member Jstring    name
    member Jstring    mode
    member u_int      action
    member u_int      type
    member u_int      updatePeriod    // msec
    member u_int      offTrackPeriod  // msec
    member u_int      dropTrackPeriod // msec
    member u_int      maxNumTargets
    member SensorAccuracy accuracy
    member float      minDetectThreshold // dB
    member float      maxDetectThreshold // dB
```

```

    member float      maxRange          // m
    member float      power             // kW
    member float      transmitAntennaGain // dB
    member float      receiveAntennaGain // dB
    member float      processingGain    // dB
    member float      noiseFigure       // dB
    member float      systemLoss        // dB
    member float      pulsewidth
    member int        frequencyBand     // see above
    member float      maxBearingRate    // radians/ms
    member float      maxBearingAcceleration // radians/ms^2
    member float      maxElevationRate  // radians/ms
    member float      maxElevationAcceleration // radians/ms^2
    member u_int      useRFLoopGain
    member float      RFLoopGain
end message GuiSensorModel

```

2.3.3.15 GUITicInterval

```

message GuiTicInterval tag = 33
    member TicInterval msgObj
end message GuiTicInterval

```

```

class TicInterval
    member u_int ticInterval
    member u_int ticAnnotationInterval
end class TicInterval

```

2.3.3.16 GUITicReferencePoint

```

message GuiTicReferencePoint tag = 32
    member TicReferencePoint msgObj
end message GuiTicReferencePoint

```

```

class TicReferencePoint
    member u_int refType
    member int   VID
    member u_int WPID
end class TicReferencePoint

```

2.3.3.17 GUIVehicle

```

// If the kineBehavior field indicates TBM, the vehicleModelName
// provides the name of the TBM model data. Otherwise the
// vehicleModelName identifies the VehicleModel.
//
message GuiVehicle      tag = 27

// Update and delete are based on VID
//
enum Action

```

```

    etag Invalid      = 0
    etag New          = 1
    etag Update       = 2
    etag Delete       = 3
    etag ServerLoad   = 4    // sent by server when new scenario is
loaded
    end enum Action

    // Additional fields will probably need to be added to this message
    // to support Slaved, Intercept, and Racetrack vehicles.
    //
    enum KineBehavior
//    etag Invalid      = 0        // already defined for GuiVehicle
    etag Stationary   = 1
    etag Slaved       = 2
    etag Waypoints    = 3
    etag TBM          = 4
    etag Intercept    = 5
    etag Orbital      = 6
    etag Interceptor  = 7
    etag DynamicInterceptor = 8
    end enum KineBehavior

    member u_int      action
    member Jstring    vehicleName
    member Jstring    vehicleModelName
    member int        VID
    member int        newVID
    member int        kineBehavior
    member Geodetic   position        // only valid for Stationary
vehicle
    member Augmentation vehicleAugmentation
    end message GuiVehicle

```

2.3.3.18 GUIVehicleAnnotation

```

message GuiVehicleAnnotation tag = 31
    member VehicleAnnotation msgObj
end message GuiVehicleAnnotation

class VehicleAnnotation
    member int        VID
    member u_int     pointID
    member Jstring    annotation
end class VehicleAnnotation

```

2.3.3.19 GUIVehicleModel

```

message GuiVehicleModel    tag = 26

    // Update and delete are based on model name
    //
    enum Action

```

```

        etag Invalid      = 0
        etag New          = 1
        etag Update      = 2
        etag Delete      = 3
        etag ServerLoad  = 4 // sent by server when new scenario is
loaded
    end enum Action

    enum Type
//    etag Invalid      = 0 // already defined for GuiVehicleModel
        etag Surface     = 1
        etag MannedABT   = 2
        etag UnmannedABT = 3
    end enum Type

    member u_int          action
    member Jstring        name
    member u_int          type
    member float          rcsHeadOn // m^2
    member float          rcsSide // m^2
    member VehicleModelPerformance maximum
    member VehicleModelPerformance minimum
    member VehicleModelPerformance nominal
    member float          dryWeight
    member float          fuelCapacity // lbs
    member u_short        pmrfVehicleType
end message GuiVehicleModel

```

2.3.3.20 GUIVehicleModifyModel

```

message GuiVehicleModifyModel tag = 34
    member VehicleModifyModel msgObj
end message GuiVehicleModifyModel

class VehicleModifyModel
    member int          VID
    member Geodetic    startPoint
    member double      angle // radians
    member Jstring     newName
end class VehicleModifyModel

```

2.3.3.21 GUIVehicleStartTime

```

message GuiVehicleStartTime tag = 28
    member int          vid
    member Time         startTime
end message GuiVehicleStartTime

```

2.3.3.22 GUIVehicleSynch

```

message GuiVehicleSynch tag = 29
    enum Action
        etag Invalid      = 0

```

```

    etag Add                = 1
    etag Delete             = 2
    etag Resynch            = 3
    etag ResetAllTimes      = 4
    etag UnsynchAll         = 5
    etag RequestDelete      = 6
end enum Action

member VehicleWaypointPairList vehiclesSynched
member u_int action
end message GuiVehicleSynch

```

2.3.3.23 GUIWaypointPosition

```

message GuiWaypointPosition tag = 20
enum Action
    etag Invalid           = 0
    etag Predecessor       = 1
    etag Successor         = 2
    etag First              = 3
    etag Move               = 4
    etag Delete             = 5
    etag Undo               = 6
end enum Action

member int                 VID
member u_int               currentWPID // WPID is unique over entire
scenario
member u_int               newWPID
member Geodetic            newPosition
member double              newSpeed // m/ms
member double              radiusOfCurve // m
member double              angleOfInclination // rad
member double              zRate // m/ms
member u_int               action
member u_int               straight
member u_int               fuelConsumptionRateOverride
member double              fuelConsumptionRate // lbs/min
member u_int               identity
member Jstring             annotation
member u_int               mode1IFF
member u_int               mode2IFF
member u_int               mode3IFF
member u_int               mode4IFF
end message GuiWaypointPosition

```

2.3.3.24 SrvCertifiedPasswordResponse

```

message SrvCertifiedPasswordResponse tag = 130
member u_int action // echo action from
GuiCertifiedPassword
member u_int status // 1=>success 0=>failure
member Jstring errorString // valid if status=0

```

```

    member Jstring currentPassword // may be "" if not set.
    member Jstring requestedBy
end message SrvCertifiedPasswordResponse

```

2.3.3.25 SrvDataSetConversionCompleted

```

message SrvDatasetConversionCompleted tag = 132
  enum Status
    etag Failure      = 0
    etag Success      = 1
  end enum Status
  member u_int status
  member Jstring errorString
end message SrvDatasetConversionCompleted

```

2.3.3.26 SrvDisplayTrackState

```

message SrvDisplayTrackState tag = 104
  member u_int      EID           // uniquely identifies the track
  member u_int      VID
  member u_int      SID
  member Time       validTime
  member Geodetic   position
  member VelocityGeocentric speed // m/ms
  member u_char     simFlag
  member char       cat
  member u_short    spare
  // this should have another spare like the other track states
end message SrvDisplayTrackState

```

2.3.3.27 SrvListResponse

```

message SrvListResponse tag = 118
  member u_int action // same as for GuiRequestList
  member u_int status // 1=>success 0=>failure
  member Jstring errorString // valid if status=0
  member Jstring nameList[]
end message SrvListResponse

```

2.3.3.28 SrvLiveTrackUpdate

```

message SrvLiveTrackUpdate tag = 122
  member u_int      systemId
  member u_int      EID
  member u_short    VID
  member u_char     SID
  member u_char     securityLevel
  member u_short    mode2Iff
  member u_short    mode3Iff
  member u_short    systemDef1
  member u_short    systemDef2
  member u_short    systemDef3
  member u_char     systemDef4

```

```

    member u_char          trackQuality
    member u_short         vehicleType
    member Time           validTime
    member Geodetic       position
    member VelocityGeocentric velocity // m/ms
end message SrvLiveTrackUpdate

```

2.3.3.29 SrvPlannedPathDump

```

message SrvPlannedPathDump tag = 108
    member int             VID
    member u_short        msgNumber // message number in series of msgs
    member u_short        msgTotal  // number of msgs in this series
    member char           cat
    member char           spare1
    member u_short        spare2
    member TrackStateList stateList // list of points
end message SrvPlannedPathDump

```

2.3.3.30 SrvPlannedTrackState

```

message SrvPlannedTrackState tag = 105
    member u_int          EID
    member u_int          VID
    member Time           validTime
    member Geodetic       position
    member VelocityGeocentric speed // m/ms
    member char           cat
    member u_char         spare1
    member u_short        spare2
end message SrvPlannedTrackState

```

2.3.3.31 SrvScenarioControlResponse

```

// The list field is valid if action=GetScenarioList or GetTBMList and
// status indicates success.
//
// Used by: Scenario
//
message SrvScenarioControlResponse tag = 102
    enum Status
        etag Failure      = 0
        etag Success      = 1
        etag Start        = 2
    end enum Status

    member u_int          action // echo action from
GuiScenarioControl
    member u_int          status
    member Jstring        name
    member Jstring        hostname // "" if sent to all
    member Jstring        errorString // valid if status=0
    member float          clockRate // used for playback

```

```

    member Time          scenarioTime // used for playback
    member u_int         simType
    member u_int         detectionType
end message SrvScenarioControlResponse

```

2.3.3.32 SrvScenarioObjectResponse

```

message SrvScenarioObjectResponse tag = 128
    enum Status
        etag Failure      = 0
        etag Success      = 1
    end enum Status

    member u_int type          // echo type from GuiScenarioObject
        member u_int status    // 1=>success 0=>failure
        member u_int action
    member Jstring name
    member Jstring errorString // valid if status=0
end message SrvScenarioObjectResponse

```

2.3.3.33 SrvScenarioSynch

```

message SrvScenarioSynch tag = 119
    member u_int status
    member Jstring errorString
    member int scenarioBegin // mS relative to T0
    member int previewBegin  // time to begin preview/exec
    member int scenarioEnd
end message SrvScenarioSynch

```

2.3.3.34 SrvSensorInformationTableDump

```

message SrvSensorInformationTableDump tag = 109
    member SensorInfoList sensorList
    member u_int           dummy
end message SrvSensorInformationTableDump

```

2.3.3.35 SrvSNvsTimeResponse

```

message SrvSNvsTimeResponse tag = 121
    member int      VID
    member int      SID
    member u_int    last
    member u_int    numSamples
    member SNvsTime snSamples[]
end message SrvSNvsTimeResponse

```

2.3.3.36 SrvSNErrors

```

message SrvSNErrors tag = 133
    member Time time
    member Jstring errorString

```

```

    member u_int VID
end message SrvSNErrors

```

2.3.3.37 SrvTicStatus

```

message SrvTicStatus tag = 116
    member u_int      refType    // 0:t0 1:reference point
    member int        VID
    member u_int      WPID
    member u_int      ticInterval // ms
    member u_int      ticAnnotationInterval
end message SrvTicStatus

```

2.3.3.38 SrvTimeTic

```

message SrvTimeTic tag = 117
    member int        VID
    member u_int      resetTics // for this vehicle
    member u_short    ticNumber // are these
    member u_short    ticTotal  // really necessary?
    member Time       time
    member Geodetic   position
    member double     heading    // rad
    member double     horizontalSpeed // m/ms
    member double     zRate      // m/ms
    member double     fuelConsumption // lbs
    member double     fuelConsumptionPercent
    member Jstring    annotation
end message SrvTimeTic

```

2.3.3.39 SrvVehicleAnnotation

```

message SrvVehicleAnnotation tag = 115
    member int        VID
    member u_int      PointID // 0 -> N
    member Jstring    Annotation
end message SrvVehicleAnnotation

```

2.3.3.40 SrvVehicleReportComplete

```

message SrvVehicleReportCompleted tag = 131
    enum Status
        etag Failure    = 0
        etag Success    = 1
    end enum Status

    member u_int status
    member Jstring errorString
end message SrvVehicleReportCompleted

```

2.3.3.41 SrvVehicleSynch

```

message SrvVehicleSynch tag = 114
  enum Action
    etag Invalid          = 0
    etag Add              = 1
    etag Delete           = 2
  end enum Action
  member VehicleWaypointPairList vehiclesSynched
  member u_int action
end message SrvVehicleSynch

```

2.3.3.42 SrvWaypointError

```

message SrvWaypointError tag = 110
  member u_int VID
  member u_int WPID
  member Jstring errorMessage
end message SrvWaypointError

```

2.3.3.43 SrvWaypointPath

```

message SrvWaypointPath tag = 100
  member WaypointPath msgObj
end message SrvWaypointPath

class WaypointPath
  member int          VID
  member u_int        resetPath
  member u_short      segmentNumber
  member u_short      segmentTotal
  member u_int        errorVerificationRequested
  member Geodetic     centerCurve
  member double       radiusOfCurve           // m
  member double       startBearingCurve       // rad
  member double       stopBearingCurve        // rad
  member Geodetic     startStraight
  member Geodetic     stopStraight
end class WaypointPath

```

2.3.3.44 SrvWaypointPosition

```

message SrvWaypointPosition tag = 101
  member WaypointPosition msgObj
end message SrvWaypointPosition

class WaypointPosition

  // enum used by GuiWaypointPosition
  enum Action
    etag Invalid          = 0

```

```

    etag Predecessor      = 1
    etag Successor       = 2
    etag First           = 3
    etag Move            = 4
    etag Delete          = 5
    etag Undo            = 6
end enum Action

// enum used by SrvWaypointPosition
enum Identity
    etag Pending         = 0
    etag Unknown         = 1
    etag AssumedFriend   = 2
    etag Friend          = 3
    etag Neutral         = 4
    etag Suspect         = 5
    etag Hostile         = 6
    etag Undefined       = 7
end enum Identity

// common members used by GuiWaypointPosition and
SrvWaypointPosition
// messages.

member int             VID
member double          radiusOfCurve           // m
member double          angleOfInclination      // rad
member double          zRate                   // m/ms
member double          fuelConsumptionRateOverride
member double          fuelConsumptionRate     // lbs/min
member u_int           straight
member u_int           identity
member Jstring         annotation

// members belonging to GuiWaypointPosition

member u_int           currentWPID
member u_int           newWPID
member Geodetic        newPosition
member double          newSpeed                // m/ms
member u_int           action

// members belonging to SrvWaypointPosition

member int             WPID
member u_short         waypointNumber
member u_short         waypointTotal
member Geodetic        position
member double          speed                   // m/ms
member u_int           direction
member u_int           calculated
member Time            time
member double          fuelConsumption        // lbs
member double          fuelConsumptionPercent

```

```
member u_int      mode1IFF
member u_int      mode2IFF
member u_int      mode3IFF
member u_int      mode4IFF

end class WaypointPosition
```

3.0 External System Interfaces

This section documents the interfaces between RMT and external systems at PMRF. Communication with these external systems takes place by means of messages.

Sensors at the range are connected to either the Kahuna Net (kNet) via Source Integration Server (SIS) units, or directly to the PMRF Instrumentation Network (iNet), a distributed processing environment for range instrumentation data. A Fault Tolerant Server (FTS) hosts a Solipsys Corporation Data Fusion application that provides an iNet and a kNet interface, performing integration and correlation of track information received from sensors on both networks. Each SIS hosts a data fusion daemon that receives data from its sensor and broadcasts iNet MT 26 messages onto the kNet. Other legacy sensors broadcast data directly to the iNet in the form of iNet MT 25 messages. The Data Fusion process known as *stick* running on the FTS acts as a smart router between the iNet, the kNet, and the DisplayNet, transforming iNet messages from the iNet and kNet into the track-related messages that drive the RMT displays.

3.1 Data Fusion Messaging Interface

Real-time event monitoring is typically accomplished by executing RMT in Preview mode while observing actual events.

Stick, running on the FTS, acts as the primary event source for the live track-related messages that drive the RMT GUI. It monitors the kNet for iNet MT 26 messages and transforms them into iNet MT25 messages, which it then rebroadcasts on the iNet. From the iNet MT 25 and MT 26 messages, it also generates TrackUpdate messages that it puts on the Display Net.

The sensor-specific data fusion daemon running on the SIS listens for contact information from its sensor and sends out ContactUpdate messages. It monitors the kNet and the iNet for track-related messages, translates them into the corresponding ContactUpdate, TrackUpdate, or DropTrack messages, and puts these translated messages on the DisplayNet. The RMT server then passes the track data messages on to the GUI, where they are processed in the InetClientModule for display.

Table 3-1 summarizes the messages that are sent from Data Fusion processes running on the FTS or on a SIS to the RMT GUI.

Table 3-1. Data Fusion to RMT GUI Messages

Message	Description	Origin	Processed in GUI
ContactUpdate	Sent from the fault-tolerant server or SIS to the GUI for	SIS data fusion daemon	EspritClientModule, ContactUpdateProcessor.p

Message	Description	Origin	Processed in GUI
	sensor contact display		rocessMessage
DropTrack	Sent by the server to instruct the GUI to drop a track	SIS data fusion daemon – iNet Event message MT45	EspritClientModule.DropTrackProcessor.processMessage
TrackUpdateBuffer	Sent from the fault-tolerant server to the GUI to tell it to display one or more tracks for situation awareness	<i>Stick</i> –iNet message Track Data messages MT25, MT26	EspritClientModule.LiveTrackState.processMessage
SrvBearingLine	Tells the GUI the position and bearing of Ownship.	SIS data fusion daemon	ScenarioListener.processBearingLine
SrvOwnshipReferencePos	Tells the GUI to set the ownship reference position to the specified geodetic point.	SIS data fusion daemon	ScenarioListener.processOwnshipReferencePos

The following subsections specify the message formats for each of the Data Fusion messages sent to the RMT GUI. Like the RMT server-GUI messages, they are defined using Metagen's Meta language syntax.

Messages defined in KNetMsg.meta:

3.1.1 ContactUpdate

```
message ContactUpdate tag = 4
  member u_char          SID
  member u_char          spare1
  member u_short         modelIFF
  member u_short         mode2IFF
  member u_short         mode3AIFF
  member u_short         mode4IFF
  member u_int           modeCIFF
  member Time            validTime
  member Geocentric      position
end message ContactUpdate
```

3.1.2 DropTrack

```
message DropTrack tag = 5
  member u_int EID
end message DropTrack
```

3.1.3 TrackUpdateBuffer

```
message TrackUpdateBuffer tag = 6
  member TrackUpdateObj tracks[]
end message TrackUpdateBuffer

class TrackUpdateObj
  member u_int          systemId
  member u_int          EID // uniquely identifies the track
  member u_short        VID
```

```

    member u_char          SID
    member u_char          securityLevel
    member u_short         mode1IFF
    member u_short         mode2IFF
    member u_short         mode3IFF
    member u_short         mode4IFF
    member u_char          trackQuality
    member u_char          spare
    member u_short         vehicleType
    member Time            validTime
    member Geocentric      position
    member VelocityGeocentric velocity // m/ms
end class TrackUpdateObj

```

Messages defined in RMTMsg.meta:

3.1.4 SrvBearingLine

```

message SrvBearingLine tag = 125
    member u_int          clear // 0 = dont clear
                                // 1 = clear bearing line
    member double         bearing
    member Geodetic       position
end message SrvBearingLine

```

3.1.5 SrvOwnshipReferencePos

```

message SrvOwnshipReferencePos tag = 124
    member u_int          clear // 0 = dont clear
                                // 1 = clear ownship pos
    member Geodetic       position
end message SrvOwnshipReferencePos

```

3.2 Simulation Messaging Interface

During local execution, the rehearsal mode of RMT, the RMT can be set to generate simulated radar detection and/or radar tracking messages in addition to the planned vehicle positions. When detection simulation is enabled, the user can select whether messages in DTE format or in the Multi Source Correlator Tracker (MSCT) variant of DTE format are to be generated.

For track simulation, the user can select whether track simulation is to be performed using Distributed Interactive Simulation (DIS) or by generating PMRF iNet messages.

3.2.1 Track Simulation Using iNet Messages

When track simulation is enabled, the user can specify that RMT generate iNet MT25, iNet MT26, or iNet MT25 & MT26 messages. Table 3-2 provides an overview of the iNet message layout, which consists of a 12-byte header, message data, and a 4-byte trailer.

Table 3-2. iNet Message Layout

31	16 15	0
Block Length		Message Identifier
Message Time (Seconds since beginning of year)		
Message Data Area (0-1452 bytes)		
Trailer		

3.2.1.1 iNet Track Data Message (MT 25)

The Track Data message is sent by sensors on the iNet to report track data. This message is comprised of header words, track data, and a message trailer, as shown in Figure2-1. above. A track is identified by a unique VID / SID pair in conjunction with a System ID (in the interface header). Table 3-3 shows the format of the Track Data Message 25. For further detail, refer to the PMRF iNet Interface Design Specification.

Table 3-3. Format for iNet Track Data Message 25.

31	24 23	16 15	8 7	0 word
VID		SID	Security Level	0
System Defined 1		SystemDefined 2		1
System Defined 3		System Defined 4	Track Quality	2
Data Time (Milliseconds)		Vehicle Type		3
Time (Seconds since beginning of year)				4

X Position (feet)	5
Y Position (feet)	6
Z Position (feet)	7
X Velocity (feet/sec)	8
Y Velocity (feet/sec)	9
Z Velocity (feet/sec)	10

Table 3-4 supplies the field definitions for the track data message 25.

Table 3-4. Field Definitions for iNet Track Data Message 25

Word	Field Name and Description	Units
0 (bits 16-31)	VID – Vehicle Identifier -- Integer from 1-999	
0 (bits 8-15)	SID – Sensor Identifier -- Integer from 1-255	
0 (bits 0-7)	Security Level – 1-5=Unclassified, 6-10=Confidential, 11-15=Secret	
1 (bits 16-31)	System Defined 1	
1 (bits 0-15)	System Defined 2	
2 (bits 16-31)	System Defined 3	
2 (bits 8-15)	System Defined 4	
2 (bits 0-7)	Track Quality – 0=Invalid, 1-6=Data Valid (system-dependent), 7=Data Valid	
3 (bits 16-31)	Data Time in milliseconds	msec
3 (bits 0-15)	Vehicle Type – See PMRF iNet IDS, appendix B	
4	Data Time in seconds since the beginning of the year (coordinated universal time)	sec
5	X Position –x coordinate of geocentric position referenced to the DOD-WGS-84 earth model (1sb = 1 ft)	feet
6	Y Position –y coordinate of geocentric position referenced to the DOD-WGS-84 earth model (1sb = 1 ft)	feet
7	Z Position –z coordinate of geocentric position referenced to the DOD-WGS-84 earth model (1sb = 1 ft)	feet
8	X Velocity –x component of geocentric velocity referenced to the DOD-WGS-84 earth model (1sb = 1/256 ft/sec)	Feet/second

Word	Field Name and Description	Units
9	Y Velocity –y component of geocentric velocity referenced to the DOD-WGS-84 earth model (lsb = 1/256 ft/sec)	Feet/second
10	Z Velocity –z component of geocentric velocity referenced to the DOD-WGS-84 earth model (lsb = 1/256 ft/sec)	Feet/second

3.2.1.2 iNet Track Data Message (MT 26)

The Track Data message 26 is sent by sensors on the kNet to report track data. It differs from the message 25 Track Data Format by being variable in size and content. However, elements 1 and 2 are required for each track. Table 3-5 supplies the field definitions for the track data message 26. For further detail, refer to the PMRF iNet Interface Design Specification.

Table 3-5. Field Definitions for iNet Track Data Message 26

Element ID	Length	Data type	Definition
1	var	multi-element	group of elements pertaining to one track
2	72	struct	id, time, position and velocity
3	32	double (3)	Acceleration (xyz geocentric)(feet/sec/sec)
4	16	double (1)	radar altimeter (feet)
5	16	double (1)	barometric altimeter (feet)
6	32	double (3)	Attitude (Euler – yaw pitch roll re: local horizontal plane) (degrees) yaw re: north; pitch re: horizontal; roll re: body axis All are increasing positive during a climbing right turn.
7	32	double (3)	Radar variance (xyz) (feet)
8	32	double (3)	Processed - Range, Azimuth, Elevation
9	32	double (3)	Raw – Range, Azimuth, Elevation
10	32	double (3)	Bias Error - Range, Azimuth, Elevation
11	32	double (3)	Radar Correction error - Range, Elevation, Refractivity
12	24	int (5)	Radar Filter Sizes (# of data points of smoothing)
13	32	double (3)	Course, Speed, Altitude
14	32	double (3)	Latitude, Longitude, Altitude
15	24	float (5)	Radar temperature data
16	56	double (6)	3-state (geocentric position) symmetrical positive definite (6-element) covariance matrix (feet) Note2.
17	176	double (21)	6-state (geocentric position and velocity) symmetrical positive definite (21 elements) covariance matrix (feet) Note2.

Element ID	Length	Data type	Definition
18	368	double (45)	9-state (geocentric position, velocity, and acceleration) symmetrical positive definite (45 element) covariance matrix (feet) Note2.
19			Raw radar return (Spherical) (similar to #9?) (Solipsys)
20	16	u short (4)	SIF codes: This element contains the IFF data that is lifted from the system defined areas of a message 25 when the reporting system is either APIS or NTDS.(Solipsys)
21		struct ?	Future for SHOTS variance? 20Feb2001 not used?
22	16	double (1)	Time to Intercept
23	32	double (3)	Acquisition - Range, Azimuth, Elevation
24	---	---	future expansion

3.2.2 Track Simulation using DIS Messages

Enabling DIS simulation will cause RMT to broadcast DIS EntityStatePDU's on the broadcast network defined by the environment variable DIS_NET (defaults to 10.0.1.255 or the primary interface) and the UDP port defined by the environment variable DIS_PORT (defaults to 3000). Please refer to the DIS Specification for a description of the EntityStatePDU format.

3.2.3 Detection Simulation using DTE Messages

Radar detections can be simulated using DTE messages, either in DTE format or in the MSCT variant of DTE format. Please refer to the DTE Specification for a description of the message format.

4.0 File Interfaces

RMT allows the user to import and export scenario information using a variety of ASCII file formats. Currently, The Johns Hopkins University Applied Physics Laboratory (JHU/APL) plans the Theater Ballistic Missile Defense (TBMD) missions that are conducted at PMRF, with Bill Millard supplying the models for target and interceptor trajectories. RMT's file interfaces are used to import scenario information when planning is performed outside of the tool.

4.1 GOG File

RMT allows overlay information to be imported/exported via ASCII files that follow the Generalized Overlay Generator (GOG) file format. This file format, widely used at PMRF can be output using the NTADS XY Geo Tracking Display. A GOG file specifies static overlay information that is to be drawn on the screen. As with NTADS, an overlay can be made dynamic by manually attaching it to a track so that it moves with the track.

RMT uses GOG files to import information for TBMD operations that are planned by APL. APL supplies a participant location map in the form of an .xls file that is then massaged into GOG file format and then used to import participant positions, annotations for participants, hazard regions, etc.

Overlays created in RMT using the overlay editor can also be saved in GOG file format.

The following bullets summarize the format rules for GOG files. Please refer to the sample GOG file in Appendix A for further details.

- Each object starts with "start" and ends with "end".
- Commands can be upper or lower case or mixture of both.
- Files cannot have blank lines -- use highly indented "comment" to provide appearance of blank lines.
- Maximums:
 - 100 xy items per object
 - 30 objects
 - 5 files per program
- Units
 - lat/lon are in degrees
 - xy are in yards
- Valid commands
 - comment
 - start
 - off - ignore object to end, place directly after start
 - poly = polygon

- ellipse
- circle
- arc
- annotation
- line
- linesegs
- points
- linecolor = fillcolor
- filled
- centerxy
- centerll = centerlatlon
- ref = referencepoint
- majoraxis
- minoraxis
- radius
- xy
- ll = lat lon [alt – ignored if displaying in 2D]
- anglestart
- angleend
- angledeg
- rotate
- end

In order to be accessible from RMT, GOG files should be placed on the client in a **data/RMT/gog** subdirectory of the directory into which the RMT client was installed, as depicted below:

```

${INSTALL_DIR}/
  data/
    RMT/
      gog/
```

4.2 Nominal Data Set

Nominal data set files are ASCII files that specify the Tactical Ballistic Missile (TBM) and interceptor models used for defining target and interceptor trajectories. The fixed format file consists of a one-line header specifying the name of each data element, followed by any number of fixed length records specifying the values corresponding to the data elements identified in the header. The final line of the file is signaled with a line containing the “@” character.

Table 4-1 summarizes the data format for each field in a nominal data set.

Table 4-1. Nominal Data Set Format

Header	Data Format
Time	Time(s)
Xecfc	X position (m; geocentric coordinates)
Yecfc	Y position (m; geocentric coordinates)
Zecfc	Z position (m; geocentric coordinates)
Xecfcdt	X velocity (m/sec)
Yecfcdt	Y velocity (m/sec)
Zecfcdt	Z velocity (m/sec)
Xecfcdt2	X acceleration (m/sec ²)
Yecfcdt2	Y acceleration (m/sec ²)
Zecfcdt2	Z acceleration (m/sec ²)
Pitch	Pitch
Yaw	Yaw
Roll	Roll

The nominal data sets should be placed on the server in the **esprit/nominalTextModels** subdirectory, whether the data set describes a TBM trajectory or an interceptor trajectory. Before a data set can be used within RMT, it must be converted to a binary format. Conversion is made in the GUI by selecting the **Esprit -> Models -> Convert Data Set** menu option. This allows the user to select the ASCII data file that is to be converted, to select the corresponding Vehicle Type, and to specify a name for the model; pressing the **Convert** button performs the necessary conversion, creating a file in the **esprit/nominalModels** with the specified model name.

The described directory hierarchy is depicted below:

```

${INSTALL_DIR}/
  esprit/
    nominalTextModels/
    nominalModels/

```

4.3 Common Route Definition (CRD) File

RMT permits import of vehicle trajectories in the file format used by the Portable Flight Planning System (PFPS), an XML format known as Common Route Definition (CRD) format. The XML document is parsed using a DOM parser to translate the information into object format and thence into a waypoint defined vehicle trajectory. The format of the XML document

is described by the Data Type Definition (DTD) included in Appendix B. Appendix C shows a sample CRD file.

5.0 COTS Products

This section defines RMT's dependencies of on Commercial Off-the-Shelf (COTS) products.

5.1 Java SDK

The RMT client relies on Java Developer's Kit Version 1.3 (JDK 1.3) having been installed on the system and set to be the default version of Java.

6.0 Appendix A – Sample GOG File

The following is a sample GOG file.

```
comment this is a gog file with examples for various graphical objects
start
annotation SAMPLE_GOG_19Apr98
11 23 -158.666
end
comment
comment the rules:
comment each object starts with "start"
comment and ends with "end"
comment commands can be upper or lower case or mixture of both
comment files cannot have blank lines
comment use highly indented "comment" to provide appearance of blank
lines
                comment
comment maximums:
comment 100 xy items per object
comment 30 objects
comment 5 files per program
comment
comment lat/lon are in degrees
comment xy are in yards
comment
comment valid colors: linked to overlay colors
comment the color name means nothing,
comment the color that is displayed depends on the setting of the
comment overlay colors in xygeo
comment red, green, blue, yellow, orange, white
comment color2, color3, color4, color5, color6, color7
comment
comment valid commands:
comment comment
comment start
comment off - ignore object to end, place directly after start
comment poly = polygon
comment ellipse
comment circle
comment arc
comment annotation
comment line
comment linesegs
comment points
comment
comment linecolor = fillcolor
comment filled
comment centerxy
comment centerll = centerlatlon
```

```
comment ref = referencepoint
comment majoraxis
comment minoraxis
comment radius
comment xy
comment ll = latlon
comment anglestart
comment angleend
comment angledeg
comment rotate
comment end
comment
comment default options: (ignored)
comment linestyle solid
comment linewidth thin
comment
comment non-implemented options:
comment linewidth med
comment linewidth medium
comment linewidth thick
comment linestyle dash, dashed
comment linestyle dot, dotted
comment
comment
start
comment box with lower left corner at 22:20:00 -160:40:00
comment drawn using xy method
line
ref 22.333333 -160.666667
xy 0 0
xy 10000 0
xy 10000 10000
xy 0 10000
xy 0 0
linewidth thin
linecolor white
end
comment
start
comment box with lower left corner at 22:20:00 -160:40:00
comment drawn using xy method
line
ref 22.433333 -160.666667
xy 0 0
xy 10000 0
xy 10000 10000
xy 0 10000
xy 0 0
linewidth med
linecolor white
end
comment
start
comment box with lower left corner at 22:20:00 -160:40:00
```

```
comment drawn using xy method
line
ref 22.533333 -160.666667
xy 0 0
xy 10000 0
xy 10000 10000
xy 0 10000
xy 0 0
linewidth thick
linecolor white
end
                                comment
start
annotation thickline
ref 22.533333 -160.666667
xy 0 0
end
                                comment
start
line
ref 23.1 -160
ll 23.1 -160
ll 23.1 -160.1
ll 23 -160.1
ll 23 -160
ll 23.1 -160
linecolor yellow
end
                                comment
start
points
ref 23.5 -160.5
xy 0 -50000
linecolor yellow
end
                                comment
comment annotation works better if a "_" starts the text
comment only 1 word is allowed, connect multiple words with a non-space
"_"
                                comment
start
annotation _Point_#1
ll 23.5 -160.5
end
                                comment
start
points
xy 0 0
linecolor green
end
                                comment
start
annotation _BC
xy 0 0
```

```
linecolor green
end
comment
start
arc
ref 22.7 -159.5
centerxy 0 0
radius 20000
anglestart 180.0
comment clockwise around to 90 degrees
angledeg 270.0
comment (use angle+360 to draw arc through 360 degrees)
end
comment
start
annotation .arc
l1 22.7 -159.5
end
comment
start
line
ref 22.7 -159.5
xy 0 -20000
xy 0 0
xy 20000 0
linecolor orange
end
comment
start
annotation _LINE
ref 22.7 -159.5
xy 0 -20000
linecolor orange
end
comment
start
poly
xy -4000 -12000
xy 4000 -12000
xy 4000 -20000
xy -4000 -20000
linecolor orange
end
comment
start
annotation _poly
xy 4000 -12000
linecolor orange
end
comment
start
ellipse
ref 21.5 -160
centerxy 0 0
```

```
majoraxis 12000
minoraxis 6000
linecolor orange
end

                                comment
start
annotation _ELLIPSE_12000x6000
ll 21.5 -160
end

                                comment
start
ellipse
ref 21.5 -160.25
centerxy 0 0
majoraxis 12000
minoraxis 6000
linecolor green
filled
end
comment -----
comment   color sampler: red, green, blue, white, yellow, orange
comment -----
start
annotation COLOR2(default)
ll 22.7 -158.666
end
```

7.0 Appendix B – CRD DTD

The following is the Data Type Definition file describing the format for a Common Route Definition XML file.

```

<!--
*****-->
<!-- CRD Data Type Definition File v1.1.2
-->
<!-- Created: 3/10/2000
-->
<!-- Author: 46 TS/OGET (TYBRIN)
-->
<!-- To include in xml:
-->
<!-- <!DOCTYPE CRD SYSTEM "1.1.2">
-->
<!--
*****-->

<!ELEMENT ACCURACY_INFO (VERTICAL_ACCURACY,QUALITY,HORIZONTAL_ACCURACY)>
<!ELEMENT ADDITIONAL_POINT
(RS_BLOCK_LIST?,KIND?,DIVERT_INFO?,ID,POINT_REFERENCE,
TARGET_INFO?)>
<!ELEMENT AIMPOINT_INFO (PIXEL_X?,PIXEL_Y?)>
<!ELEMENT AIR_REFUELING_INTENT
(RENDEZVOUS_TYPE,ROUTE_SOURCE,RECEIVER_ID,
CLOSING_AIRSPED,DEVICE_TYPE,RECEIVER_NAME,ROUTE_TYPE)>
<!ELEMENT AIRDROP_INTENT (TRANSITION_POINT_TURN_TYPE,TURN_DELAY,
USABLE_DROP_ZONE_DISTANCE,FORWARD_TRAVEL_DISTANCE,TRANSITION_POINT_TURN_
SETTING,
TRANSITION_POINT_TO_SLOWDOWN_POINT_DISTANCE,DROP_ALTITUDE,RUN_IN_OPTION,
DROP_ZONE_TRUE_COURSE)>
<!ELEMENT AIRDROP_STATE (DROP_POINT_TO_RED_LIGHT_DISTANCE?)>
<!ELEMENT AIRSPEED (TRUE_AIRSPED,INPUT_TYPE)>
<!ELEMENT ALTITUDE (ALTITUDE_TYPE,VALUE)>
<!ELEMENT BEARING_AND_RANGE (RANGE,BEARING)>
<!ELEMENT CONFIGURATION_ITEM (INTERNAL_FUEL_WEIGHT?,EMPTY_WEIGHT?,NAME,
INTERNAL_FUEL_CAPACITY?,CENTER_OF_GRAVITY_X?,CENTER_OF_GRAVITY_Y?,
CENTER_OF_GRAVITY_Z?,DRAG?,CONFIGURATION_ITEM_LIST?,ID,EXTERNAL_FUEL_CAP
ACITY?,
EXTERNAL_FUEL_WEIGHT?)>
<!ELEMENT CRD
(SOURCE,CLASSIFICATION_LABEL?,CLASSIFICATION,MISSION_LIST)>
<!ELEMENT DELAY_INTENT (AUTO_TIME_FLAG?,COMMANDED_TIME?)>
<!ELEMENT DELAY_STATE (HOLDING_MSL_ALTITUDE?)>
<!ELEMENT DIVERT_INFO (FUEL_FLOW?,FPM_INTENT?,WIND,AIRSPED,TEMPERATURE,
MSL_ALTITUDE,FUEL_CONSUMED?)>
<!ELEMENT ELEVATION (VALUE,SOURCE?)>
<!ELEMENT FPM_INPUT_PAIR (ID,VALUE)>
<!ELEMENT FPM_INTENT (CRUISE_FPM_MODE_INTENT,MANUAL_DESCENT_RATE,

```

```

MANUAL_CLIMB_AIRSPEED,MANUAL_DESCENT_AIRSPEED,DESCENT_FPM_MODE_INTENT,
CLIMB_FPM_MODE_INTENT,MANUAL_CLIMB_RATE)>
<!ELEMENT FPM_MODE_INTENT (FLAG?,MANUAL_FUELFLOW, ID, INPUT_LIST?)>
<!ELEMENT FUEL_CHANGE_INTENT
(FUEL_RATE?,TRANSFER_METHOD,FUEL_TOPPED_OFF_FLAG?,
END_FUEL?,LOADED_FUEL?)>
<!ELEMENT HOVER_INTENT
(AUTO_DELAY_FLAG?,HOVER_TWO_INPUT_LIST,HOVER_MODE_ONE,
HOVER_ONE_INPUT_LIST,STORE_WEIGHT_CHANGE,FUEL_CHANGE,MANUAL_FUEL_FLOW_FL
AG,
HOVER_TIME,MANUAL_FUEL_FLOW?,HOVER_MODE_TWO)>
<!ELEMENT IMAGE_INFO (IMAGE_TYPE?,AIMPOINT_INFO?,IMAGE_FILENAME)>
<!ELEMENT INTENT
(SPEED_CHANGE_INTENT?,DELAY_INTENT?,ORBIT_INTENT?,LANDING_INTENT?,
AIRDROP_INTENT?,TURN_INTENT?,COMMANDED_LEG_AIRSPEED?,AIR_REFUELING_INTEN
T?,
CLIMB_DESCENT_WIND?,TEMPERATURE?,HOVER_INTENT?,ID,COMMANDED_ALTITUDE?,
FUEL_CHANGE_INTENT?,FUELFLOW_INCREMENT_PERCENTAGE?,RESET_HACK_TIME_FLAG?
,
COMMANDED_EXIT_ROUTE_POINT_REFERENCE,TIME_ON_TARGET?,WEAPONS_RELEASE_INT
ENT?,
COMMANDED_ENTRY_ROUTE_POINT_REFERENCE,CRUISE_WIND?,DELTA_GROSS_WEIGHT?,D
ELTA_DRAG?,
TRACK_INTENT?,TAKEOFF_INTENT?,FPM_INTENT?)>
<!ELEMENT LANDING_INTENT
(RECOVERY_PLATFORM_TYPE?,FUEL?,TIME?,RUNWAY_INFO?)>
<!ELEMENT MESSAGE (TEXT, ID)>
<!ELEMENT MISSION (ID,DAFIF_DATE_TIME,CREATION_DATE_TIME,POINT_LIST?,
MISSION_NAME?,ROUTE_LIST,VEHICLE_LIST?,THREAT_LIST?)>
<!ELEMENT ORBIT_INTENT
(COMMANDED_COMPLETION_TIME?,TURN_RIGHT,CLOCK_TYPE,
AUTO_TIME_FLAG,INBOUND_TRUE_COURSE,LEG_DISTANCE,BANK_ANGLE)>
<!ELEMENT ORBIT_STATE (HOLDING_MSL_ALTITUDE?,INBOUND_TRUE_HEADING?,
OUTBOUND_TRUE_HEADING?)>
<!ELEMENT PATH (TRANSITION_LIST, ID,HEADING_CALCULATION_OPTION?,
MAGNETIC_HEADING_CALCULATION_OPTION?,EMERGENCY_SAFE_AGL_ALTITUDE?,BINGO_
FUEL?,
MESSAGE_LIST?,PATH_TYPE?,INS_XREF_POINT_REFERENCE?)>
<!ELEMENT POINT
(ID,MAGNETIC_VARIATION?,DESCRIPTION?,DATABASE_LOOKUP_STRING?,
ELEVATION?,WGS84_POSITION,SOURCE?,NAME?,INPUT_DATUM?)>
<!ELEMENT ROUTE
(INTENT_LIST,STATE_LIST,POINT_LIST?,VEHICLE_REFERENCE?,NAME?,
RS_BLOCK_LIST?,DEPARTURE_DATE?,ID,RECOVERY_FUEL?,IMAGE_DATA?,PATH_LIST,
ROUTE_POINT_LIST,TIME_ZONE_DESIGNATION?)>
<!ELEMENT ROUTE_POINT (POINT_ACCURACY?,RS_BLOCK_LIST?,POINT_REFERENCE,
ADDITIONAL_POINT_LIST?,TARGET_INFO?,HORIZONTAL_LOOK_ANGLE?,VERTICAL_LOOK
_ANGLE?,
INFRARED_POLARITY?,DTD_ID?,KIND?,SENSOR_FIELD_OF_VIEW?,FIELD_OF_VIEW?,ID
,
CLOUD_CEILING?)>
<!ELEMENT RS_BLOCK (ID,BASE64,REMARK?,TYPE)>
<!ELEMENT RUNWAY_INFO
(MAGNETIC_HEADING?,APPROACH_BARRIERS?,SLOPE?,SURFACE?,

```

```

DEPARTURE_BARRIERS?, LENGTH?, RUNWAY_ID?)>
<!ELEMENT SPEED_CHANGE_INTENT
(COMMANDED_TYPE, SLOWDOWN_ONE_LEG_ELEVATION,
SLOWDOWN_ONE_LEG_TEMPERATURE, SLOWDOWN_ONE_LEG_DELTA_TEMPERATURE,
SLOWDOWN_TWO_LEG_FUELFLOW?, SLOWDOWN_ONE_LEG_TIME_ADJUSTMENT,
SLOWDOWN_ONE_LEG_PRIOR_TIME, SLOWDOWN_ONE_LEG_PRIOR_DISTANCE,
SLOWDOWN_ONE_LEG_AIRSPPEED, SLOWDOWN_TWO_LEG_WIND?, SLOWDOWN_ONE_LEG_WIND,
COMMANDED_TIME_ADJUSTMENT, SLOWDOWN_TWO_LEG_PRIOR_TIME?,
SLOWDOWN_TWO_LEG_DELTA_TEMPERATURE?, SLOWDOWN_TWO_LEG_TEMPERATURE?,
SLOWDOWN_TWO_LEG_TIME_ADJUSTMENT?, SLOWDOWN_TWO_LEG_AIRSPPEED?,
SLOWDOWN_TWO_LEG_PRIOR_DISTANCE?, SLOWDOWN_ONE_LEG_ALTITUDE,
SLOWDOWN_ONE_LEG_FUELFLOW)>
<!ELEMENT STATE (ID, REMAINING_TIME_DISTANCE_AND_FUEL?, GROSS_WEIGHT?,
CALCULATED_GROSS_WEIGHT?, ACTUAL_MSL_ALTITUDE?, CUMULATIVE_TIME_DISTANCE_A
ND_FUEL?,
LEG_FUELFLOW?, ROUTE_POINT_REFERENCE, ORBIT_STATE?, TURN_STATE?, STORE_WEIGH
T?,
NAVAID_RADIAL?, DELAY_STATE?, LEG_TIME_DISTANCE_AND_FUEL?, ACTUAL_WIND?,
INS_XREF_BEARING_AND_RANGE?, CONTINUATION_FUEL?, MAGNETIC_HEADING?, CLOCK_T
IME?,
AIRDROP_STATE?, CALCULATED_DRAG?, CALCULATED_TRUE_AIRSPPEED?, MAGNETIC_COURS
E?,
FLIGHT_ATTITUDE?, CALCULATED_ACTUAL_FUEL_CHANGE?, ACTUAL_TEMPERATURE?, TRUE
_COURSE?,
ALTERNATE_MSL_ALTITUDE?, TRUE_HEADING?)>
<!ELEMENT TAKEOFF_INTENT (LAUNCH_PLATFORM_TYPE?, TRUE_HEADING,
TIME_DISTANCE_AND_FUEL?, RUNWAY_INFO?)>
<!ELEMENT TARGET_INFO (OSUFFIX, BE_NUMBER, LENGTH, HEIGHT, WIDTH)>
<!ELEMENT TEMPERATURE (VALUE, TEMPERATURE_TYPE)>
<!ELEMENT THREAT_EQUIPMENT_INFO
(RADAR_EQUIPMENT_CODE, SAM_EQUIPMENT_CODE)>
<!ELEMENT THREAT_INFO
(VERTICAL_DETECTION_RANGE?, HORIZONTAL_LETHAL_RANGE?,
HORIZONTAL_DETECTION_RANGE?, POINT_REFERENCE, THREAT_TYPE, DETECT_RANGE_FLA
G,
THREAT_EQUIPMENT?, LETHAL_RANGE_FLAG, VERTICAL_LETHAL_RANGE?)>
<!ELEMENT TIME_DISTANCE_AND_FUEL (FUEL, DISTANCE, TIME)>
<!ELEMENT TRACK_INTENT (TRACK_TYPE)>
<!ELEMENT TRANSITION (END_STATE_REFERENCE, CALCULATION_FLAG?,
MINIMUM_SAFE_ALTITUDE?, INTENT_REFERENCE, START_STATE_REFERENCE, TRANSITION
_LIST?, ID)>
<!ELEMENT TURN_INTENT (TURN_SETTING?, S_TURN_INTERCEPT_ANGLE?, TURN_TYPE)>
<!ELEMENT TURN_SETTING (TURN_SETTING_TYPE?, VALUE)>
<!ELEMENT TURN_STATE
(TURN_ARC?, TURN_CENTER_WGS84_POSITION?, RADIUS_OF_TURN?)>
<!ELEMENT VEHICLE
(CONFIGURATION_ITEM?, FUEL_TYPE?, FPM_VERSION?, MINIMUM_FUEL?,
FPM_VEHICLE_ID?, ID)>
<!ELEMENT WEAPONS_RELEASE_INTENT (CONFIGURATION_ITEM_REFERENCE)>
<!ELEMENT WGS84_POSITION (LONGITUDE, LATITUDE)>
<!ELEMENT WIND (SPEED, DIRECTION)>
<!--
*****_-->

```

```

<!-- Define LIST Objects
-->
<!--
*****_-->
<!ELEMENT ADDITIONAL_POINT_LIST (ADDITIONAL_POINT+)>
<!ELEMENT CONFIGURATION_ITEM_LIST (CONFIGURATION_ITEM+)>
<!ELEMENT HOVER_ONE_INPUT_LIST (FPM_INPUT_PAIR+)>
<!ELEMENT HOVER_TWO_INPUT_LIST (FPM_INPUT_PAIR+)>
<!ELEMENT INPUT_LIST (FPM_INPUT_PAIR+)>
<!ELEMENT INTENT_LIST (INTENT+)>
<!ELEMENT MESSAGE_LIST (MESSAGE+)>
<!ELEMENT MISSION_LIST (MISSION+)>
<!ELEMENT PATH_LIST (PATH+)>
<!ELEMENT POINT_LIST (POINT+)>
<!ELEMENT ROUTE_LIST (ROUTE+)>
<!ELEMENT ROUTE_POINT_LIST (ROUTE_POINT+)>
<!ELEMENT RS_BLOCK_LIST (RS_BLOCK+)>
<!ELEMENT STATE_LIST (STATE+)>
<!ELEMENT THREAT_LIST (THREAT_INFO+)>
<!ELEMENT TRANSITION_LIST (TRANSITION+)>
<!ELEMENT VEHICLE_LIST (VEHICLE+)>
<!--
*****_-->
<!-- Define DATA Objects
-->
<!--
*****_-->
<!ELEMENT ACTUAL_WIND (WIND)>
<!ELEMENT CLIMB_DESCENT_WIND (WIND)>
<!ELEMENT CLIMB_FPM_MODE_INTENT (FPM_MODE_INTENT)>
<!ELEMENT CLOSING_AIRSPEED (AIRSPEED)>
<!ELEMENT COMMANDED_ALTITUDE (ALTITUDE)>
<!ELEMENT COMMANDED_LEG_AIRSPEED (AIRSPEED)>
<!ELEMENT CRUISE_FPM_MODE_INTENT (FPM_MODE_INTENT)>
<!ELEMENT CRUISE_WIND (WIND)>
<!ELEMENT CUMULATIVE_TIME_DISTANCE_AND_FUEL (TIME_DISTANCE_AND_FUEL)>
<!ELEMENT DESCENT_FPM_MODE_INTENT (FPM_MODE_INTENT)>
<!ELEMENT DROP_ALTITUDE (ALTITUDE)>
<!ELEMENT IMAGE_DATA (IMAGE_INFO)>
<!ELEMENT INS_XREF_BEARING_AND_RANGE (BEARING_AND_RANGE)>
<!ELEMENT LEG_TIME_DISTANCE_AND_FUEL (TIME_DISTANCE_AND_FUEL)>
<!ELEMENT MANUAL_CLIMB_AIRSPEED (AIRSPEED)>
<!ELEMENT MANUAL_DESCENT_AIRSPEED (AIRSPEED)>
<!ELEMENT MINIMUM_SAFE_ALTITUDE (ALTITUDE)>
<!ELEMENT POINT_ACCURACY (ACCURACY_INFO)>
<!ELEMENT REMAINING_TIME_DISTANCE_AND_FUEL (TIME_DISTANCE_AND_FUEL)>
<!ELEMENT SLOWDOWN_ONE_LEG_AIRSPEED (AIRSPEED)>
<!ELEMENT SLOWDOWN_ONE_LEG_ALTITUDE (ALTITUDE)>
<!ELEMENT SLOWDOWN_ONE_LEG_ELEVATION (ELEVATION)>
<!ELEMENT SLOWDOWN_ONE_LEG_TEMPERATURE (TEMPERATURE)>
<!ELEMENT SLOWDOWN_ONE_LEG_WIND (WIND)>
<!ELEMENT SLOWDOWN_TWO_LEG_AIRSPEED (AIRSPEED)>
<!ELEMENT SLOWDOWN_TWO_LEG_TEMPERATURE (TEMPERATURE)>
<!ELEMENT SLOWDOWN_TWO_LEG_WIND (WIND)>

```

```
<!ELEMENT THREAT_EQUIPMENT (THREAT_EQUIPMENT_INFO)>
<!ELEMENT TRANSITION_POINT_TURN_SETTING (TURN_SETTING)>
<!ELEMENT TURN_CENTER_WGS84_POSITION (WGS84_POSITION)>
<!ELEMENT ALTITUDE_TYPE (#PCDATA)>
<!ELEMENT APPROACH_BARRIERS (#PCDATA)>
<!ELEMENT CLASSIFICATION (#PCDATA)>
<!ELEMENT CLOCK_TYPE (#PCDATA)>
<!ELEMENT COMMANDED_TYPE (#PCDATA)>
<!ELEMENT DEPARTURE_BARRIERS (#PCDATA)>
<!ELEMENT DEVICE_TYPE (#PCDATA)>
<!ELEMENT FLAG (#PCDATA)>
<!ELEMENT FLIGHT_ATTITUDE (#PCDATA)>
<!ELEMENT FPM_VEHICLE_ID (#PCDATA)>
<!ELEMENT FUEL_TYPE (#PCDATA)>
<!ELEMENT HEADING_CALCULATION_OPTION (#PCDATA)>
<!ELEMENT IMAGE_TYPE (#PCDATA)>
<!ELEMENT INFRARED_POLARITY (#PCDATA)>
<!ELEMENT INPUT_DATUM (#PCDATA)>
<!ELEMENT INPUT_TYPE (#PCDATA)>
<!ELEMENT KIND (#PCDATA)>
<!ELEMENT LAUNCH_PLATFORM_TYPE (#PCDATA)>
<!ELEMENT MAGNETIC_HEADING_CALCULATION_OPTION (#PCDATA)>
<!ELEMENT PATH_TYPE (#PCDATA)>
<!ELEMENT RECEIVER_ID (#PCDATA)>
<!ELEMENT RECOVERY_PLATFORM_TYPE (#PCDATA)>
<!ELEMENT RENDEZVOUS_TYPE (#PCDATA)>
<!ELEMENT ROUTE_SOURCE (#PCDATA)>
<!ELEMENT ROUTE_TYPE (#PCDATA)>
<!ELEMENT RUN_IN_OPTION (#PCDATA)>
<!ELEMENT SOURCE (#PCDATA)>
<!ELEMENT SURFACE (#PCDATA)>
<!ELEMENT TEMPERATURE_TYPE (#PCDATA)>
<!ELEMENT TRACK_TYPE (#PCDATA)>
<!ELEMENT TRANSFER_METHOD (#PCDATA)>
<!ELEMENT TRANSITION_POINT_TURN_TYPE (#PCDATA)>
<!ELEMENT TURN_SETTING_TYPE (#PCDATA)>
<!ELEMENT TURN_TYPE (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT ACTUAL_MSL_ALTITUDE (#PCDATA)>
<!ELEMENT ACTUAL_TEMPERATURE (#PCDATA)>
<!ELEMENT ALTERNATE_MSL_ALTITUDE (#PCDATA)>
<!ELEMENT AUTO_DELAY_FLAG (#PCDATA)>
<!ELEMENT AUTO_TIME_FLAG (#PCDATA)>
<!ELEMENT BANK_ANGLE (#PCDATA)>
<!ELEMENT BASE64 (#PCDATA)>
<!ELEMENT BE_NUMBER (#PCDATA)>
<!ELEMENT BEARING (#PCDATA)>
<!ELEMENT BINGO_FUEL (#PCDATA)>
<!ELEMENT CALCULATED_ACTUAL_FUEL_CHANGE (#PCDATA)>
<!ELEMENT CALCULATED_DRAG (#PCDATA)>
<!ELEMENT CALCULATED_GROSS_WEIGHT (#PCDATA)>
<!ELEMENT CALCULATED_TRUE_AIRSPEED (#PCDATA)>
<!ELEMENT CALCULATION_FLAG (#PCDATA)>
<!ELEMENT CENTER_OF_GRAVITY_X (#PCDATA)>
```

```
<!ELEMENT CENTER_OF_GRAVITY_Y (#PCDATA)>
<!ELEMENT CENTER_OF_GRAVITY_Z (#PCDATA)>
<!ELEMENT CLASSIFICATION_LABEL (#PCDATA)>
<!ELEMENT CLOCK_TIME (#PCDATA)>
<!ELEMENT CLOUD_CEILING (#PCDATA)>
<!ELEMENT COMMANDED_COMPLETION_TIME (#PCDATA)>
<!ELEMENT COMMANDED_TIME (#PCDATA)>
<!ELEMENT COMMANDED_TIME_ADJUSTMENT (#PCDATA)>
<!ELEMENT CONTINUATION_FUEL (#PCDATA)>
<!ELEMENT CREATION_DATE_TIME (#PCDATA)>
<!ELEMENT DAFIF_DATE_TIME (#PCDATA)>
<!ELEMENT DATABASE_LOOKUP_STRING (#PCDATA)>
<!ELEMENT DELTA_DRAG (#PCDATA)>
<!ELEMENT DELTA_GROSS_WEIGHT (#PCDATA)>
<!ELEMENT DEPARTURE_DATE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT DETECT_RANGE_FLAG (#PCDATA)>
<!ELEMENT DIRECTION (#PCDATA)>
<!ELEMENT DISTANCE (#PCDATA)>
<!ELEMENT DRAG (#PCDATA)>
<!ELEMENT DROP_POINT_TO_RED_LIGHT_DISTANCE (#PCDATA)>
<!ELEMENT DROP_ZONE_TRUE_COURSE (#PCDATA)>
<!ELEMENT DTD_ID (#PCDATA)>
<!ELEMENT EMERGENCY_SAFE_AGL_ALTITUDE (#PCDATA)>
<!ELEMENT EMPTY_WEIGHT (#PCDATA)>
<!ELEMENT END_FUEL (#PCDATA)>
<!ELEMENT EXTERNAL_FUEL_CAPACITY (#PCDATA)>
<!ELEMENT EXTERNAL_FUEL_WEIGHT (#PCDATA)>
<!ELEMENT FIELD_OF_VIEW (#PCDATA)>
<!ELEMENT FORWARD_TRAVEL_DISTANCE (#PCDATA)>
<!ELEMENT FPM_VERSION (#PCDATA)>
<!ELEMENT FUEL (#PCDATA)>
<!ELEMENT FUEL_CHANGE (#PCDATA)>
<!ELEMENT FUEL_CONSUMED (#PCDATA)>
<!ELEMENT FUEL_FLOW (#PCDATA)>
<!ELEMENT FUEL_RATE (#PCDATA)>
<!ELEMENT FUEL_TOPPED_OFF_FLAG (#PCDATA)>
<!ELEMENT FUELFLOW_INCREMENT_PERCENTAGE (#PCDATA)>
<!ELEMENT GROSS_WEIGHT (#PCDATA)>
<!ELEMENT HEIGHT (#PCDATA)>
<!ELEMENT HOLDING_MSL_ALTITUDE (#PCDATA)>
<!ELEMENT HORIZONTAL_ACCURACY (#PCDATA)>
<!ELEMENT HORIZONTAL_DETECTION_RANGE (#PCDATA)>
<!ELEMENT HORIZONTAL_LETHAL_RANGE (#PCDATA)>
<!ELEMENT HORIZONTAL_LOOK_ANGLE (#PCDATA)>
<!ELEMENT HOVER_MODE_ONE (#PCDATA)>
<!ELEMENT HOVER_MODE_TWO (#PCDATA)>
<!ELEMENT HOVER_TIME (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT IMAGE_FILENAME (#PCDATA)>
<!ELEMENT INBOUND_TRUE_COURSE (#PCDATA)>
<!ELEMENT INBOUND_TRUE_HEADING (#PCDATA)>
<!ELEMENT INTERNAL_FUEL_CAPACITY (#PCDATA)>
<!ELEMENT INTERNAL_FUEL_WEIGHT (#PCDATA)>
```

```
<!ELEMENT LATITUDE (#PCDATA)>
<!ELEMENT LEG_DISTANCE (#PCDATA)>
<!ELEMENT LEG_FUELFLOW (#PCDATA)>
<!ELEMENT LENGTH (#PCDATA)>
<!ELEMENT LETHAL_RANGE_FLAG (#PCDATA)>
<!ELEMENT LOADED_FUEL (#PCDATA)>
<!ELEMENT LONGITUDE (#PCDATA)>
<!ELEMENT MAGNETIC_COURSE (#PCDATA)>
<!ELEMENT MAGNETIC_HEADING (#PCDATA)>
<!ELEMENT MAGNETIC_VARIATION (#PCDATA)>
<!ELEMENT MANUAL_CLIMB_RATE (#PCDATA)>
<!ELEMENT MANUAL_DESCENT_RATE (#PCDATA)>
<!ELEMENT MANUAL_FUEL_FLOW (#PCDATA)>
<!ELEMENT MANUAL_FUEL_FLOW_FLAG (#PCDATA)>
<!ELEMENT MANUAL_FUELFLOW (#PCDATA)>
<!ELEMENT MINIMUM_FUEL (#PCDATA)>
<!ELEMENT MISSION_NAME (#PCDATA)>
<!ELEMENT MSL_ALTITUDE (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT NAVAID_RADIAL (#PCDATA)>
<!ELEMENT OSUFFIX (#PCDATA)>
<!ELEMENT OUTBOUND_TRUE_HEADING (#PCDATA)>
<!ELEMENT PIXEL_X (#PCDATA)>
<!ELEMENT PIXEL_Y (#PCDATA)>
<!ELEMENT QUALITY (#PCDATA)>
<!ELEMENT RADAR_EQUIPMENT_CODE (#PCDATA)>
<!ELEMENT RADIUS_OF_TURN (#PCDATA)>
<!ELEMENT RANGE (#PCDATA)>
<!ELEMENT RECEIVER_NAME (#PCDATA)>
<!ELEMENT RECOVERY_FUEL (#PCDATA)>
<!ELEMENT REMARK (#PCDATA)>
<!ELEMENT RESET_HACK_TIME_FLAG (#PCDATA)>
<!ELEMENT RUNWAY_ID (#PCDATA)>
<!ELEMENT S_TURN_INTERCEPT_ANGLE (#PCDATA)>
<!ELEMENT SAM_EQUIPMENT_CODE (#PCDATA)>
<!ELEMENT SENSOR_FIELD_OF_VIEW (#PCDATA)>
<!ELEMENT SLOPE (#PCDATA)>
<!ELEMENT SLOWDOWN_ONE_LEG_DELTA_TEMPERATURE (#PCDATA)>
<!ELEMENT SLOWDOWN_ONE_LEG_FUELFLOW (#PCDATA)>
<!ELEMENT SLOWDOWN_ONE_LEG_PRIOR_DISTANCE (#PCDATA)>
<!ELEMENT SLOWDOWN_ONE_LEG_PRIOR_TIME (#PCDATA)>
<!ELEMENT SLOWDOWN_ONE_LEG_TIME_ADJUSTMENT (#PCDATA)>
<!ELEMENT SLOWDOWN_TWO_LEG_DELTA_TEMPERATURE (#PCDATA)>
<!ELEMENT SLOWDOWN_TWO_LEG_FUELFLOW (#PCDATA)>
<!ELEMENT SLOWDOWN_TWO_LEG_PRIOR_DISTANCE (#PCDATA)>
<!ELEMENT SLOWDOWN_TWO_LEG_PRIOR_TIME (#PCDATA)>
<!ELEMENT SLOWDOWN_TWO_LEG_TIME_ADJUSTMENT (#PCDATA)>
<!ELEMENT SPEED (#PCDATA)>
<!ELEMENT STORE_WEIGHT (#PCDATA)>
<!ELEMENT STORE_WEIGHT_CHANGE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT THREAT_TYPE (#PCDATA)>
<!ELEMENT TIME (#PCDATA)>
<!ELEMENT TIME_ON_TARGET (#PCDATA)>
```

```

<!ELEMENT TIME_ZONE_DESIGNATION (#PCDATA)>
<!ELEMENT TRANSITION_POINT_TO_SLOWDOWN_POINT_DISTANCE (#PCDATA)>
<!ELEMENT TRUE_AIRSPEED (#PCDATA)>
<!ELEMENT TRUE_COURSE (#PCDATA)>
<!ELEMENT TRUE_HEADING (#PCDATA)>
<!ELEMENT TURN_ARC (#PCDATA)>
<!ELEMENT TURN_DELAY (#PCDATA)>
<!ELEMENT TURN_RIGHT (#PCDATA)>
<!ELEMENT USABLE_DROP_ZONE_DISTANCE (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!ELEMENT VERTICAL_ACCURACY (#PCDATA)>
<!ELEMENT VERTICAL_DETECTION_RANGE (#PCDATA)>
<!ELEMENT VERTICAL_LETHAL_RANGE (#PCDATA)>
<!ELEMENT VERTICAL_LOOK_ANGLE (#PCDATA)>
<!ELEMENT WIDTH (#PCDATA)>
<!ELEMENT COMMANDED_ENTRY_ROUTE_POINT_REFERENCE (#PCDATA)>
<!ELEMENT COMMANDED_EXIT_ROUTE_POINT_REFERENCE (#PCDATA)>
<!ELEMENT CONFIGURATION_ITEM_REFERENCE (#PCDATA)>
<!ELEMENT END_STATE_REFERENCE (#PCDATA)>
<!ELEMENT INS_XREF_POINT_REFERENCE (#PCDATA)>
<!ELEMENT INTENT_REFERENCE (#PCDATA)>
<!ELEMENT POINT_REFERENCE (#PCDATA)>
<!ELEMENT ROUTE_POINT_REFERENCE (#PCDATA)>
<!ELEMENT START_STATE_REFERENCE (#PCDATA)>
<!ELEMENT VEHICLE_REFERENCE (#PCDATA)>

```

8.0 Appendix C – Sample CRD File

The following is the sample Common Route Definition (CRD) XML file.

```

<!DOCTYPE CRD SYSTEM "1.1.2">
<CRD>
<SOURCE>PFPS</SOURCE>
<CLASSIFICATION>UNCLASSIFIED</CLASSIFICATION>
<MISSION_LIST><MISSION>
<ID>1</ID>
<DAFIF_DATE_TIME>200103220000</DAFIF_DATE_TIME>
<CREATION_DATE_TIME>20011205132031</CREATION_DATE_TIME>
<MISSION_NAME>sample</MISSION_NAME>
<ROUTE_LIST><ROUTE>
<INTENT_LIST><INTENT>
<TURN_INTENT>
<TURN_SETTING>
<TURN_SETTING_TYPE>TURN_BANK</TURN_SETTING_TYPE>
<VALUE>0</VALUE>
</TURN_SETTING>
<TURN_TYPE>TURN_SQUARE</TURN_TYPE>
</TURN_INTENT>
<COMMANDED_LEG_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>444</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>

```

```
</AIRSPEED>
</COMMANDED_LEG_AIRSPEED>
<CLIMB_DESCENT_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CLIMB_DESCENT_WIND>
<TEMPERATURE>
<VALUE>-28.582</VALUE>
<TEMPERATURE_TYPE>C</TEMPERATURE_TYPE>
</TEMPERATURE>
<ID>1882594750</ID>
<COMMANDED_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>22000</VALUE>
</ALTITUDE>
</COMMANDED_ALTITUDE>
<FUELFLOW_INCREMENT_PERCENTAGE>1</FUELFLOW_INCREMENT_PERCENTAGE>
<RESET_HACK_TIME_FLAG>FALSE</RESET_HACK_TIME_FLAG>
<COMMANDED_EXIT_ROUTE_POINT_REFERENCE>830684835</COMMANDED_EXIT_ROUTE_PO
INT_REFERENCE>
<COMMANDED_ENTRY_ROUTE_POINT_REFERENCE>1207629152</COMMANDED_ENTRY_ROUTE
_POINT_REFERENCE>
<CRUISE_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CRUISE_WIND>
<DELTA_GROSS_WEIGHT>0</DELTA_GROSS_WEIGHT>
<DELTA_DRAG>0</DELTA_DRAG>
<TAKEOFF_INTENT>
<TRUE_HEADING>42.5</TRUE_HEADING>
<TIME_DISTANCE_AND_FUEL>
<FUEL>500</FUEL>
<DISTANCE>0</DISTANCE>
<TIME>0</TIME>
</TIME_DISTANCE_AND_FUEL>
<RUNWAY_INFO>
<MAGNETIC_HEADING>30</MAGNETIC_HEADING>
<LENGTH>9909</LENGTH>
<RUNWAY_ID>03R</RUNWAY_ID>
</RUNWAY_INFO>
</TAKEOFF_INTENT>
<FPM_INTENT>
<CRUISE_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>3059</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
```

```

</CRUISE_FPM_MODE_INTENT>
<MANUAL_DESCENT_RATE>1000</MANUAL_DESCENT_RATE>
<MANUAL_CLIMB_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_CLIMB_AIRSPEED>
<MANUAL_DESCENT_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_DESCENT_AIRSPEED>
<DESCENT_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>4</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>4028</ID>
</FPM_MODE_INTENT>
</DESCENT_FPM_MODE_INTENT>
<CLIMB_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>2013</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>969</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>1083</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CLIMB_FPM_MODE_INTENT>
<MANUAL_CLIMB_RATE>1000</MANUAL_CLIMB_RATE>
</FPM_INTENT>
</INTENT>
<INTENT>
<TURN_INTENT>
<TURN_SETTING>
<TURN_SETTING_TYPE>TURN_BANK</TURN_SETTING_TYPE>
<VALUE>0</VALUE>
</TURN_SETTING>
<TURN_TYPE>TURN_SQUARE</TURN_TYPE>
</TURN_INTENT>
<COMMANDED_LEG_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>444</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</COMMANDED_LEG_AIRSPEED>
<CLIMB_DESCENT_WIND><WIND>

```

```

<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CLIMB_DESCENT_WIND>
<TEMPERATURE>
<VALUE>-28.582</VALUE>
<TEMPERATURE_TYPE>C</TEMPERATURE_TYPE>
</TEMPERATURE>
<ID>1882594751</ID>
<COMMANDED_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>22000</VALUE>
</ALTITUDE>
</COMMANDED_ALTITUDE>
<FUELFLOW_INCREMENT_PERCENTAGE>1</FUELFLOW_INCREMENT_PERCENTAGE>
<RESET_HACK_TIME_FLAG>FALSE</RESET_HACK_TIME_FLAG>
<COMMANDED_EXIT_ROUTE_POINT_REFERENCE>69807834</COMMANDED_EXIT_ROUTE_POI
NT_REFERENCE>
<COMMANDED_ENTRY_ROUTE_POINT_REFERENCE>830684835</COMMANDED_ENTRY_ROUTE_
POINT_REFERENCE>
<CRUISE_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CRUISE_WIND>
<DELTA_GROSS_WEIGHT>0</DELTA_GROSS_WEIGHT>
<DELTA_DRAG>0</DELTA_DRAG>
<FPM_INTENT>
<CRUISE_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>3059</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CRUISE_FPM_MODE_INTENT>
<MANUAL_DESCENT_RATE>1000</MANUAL_DESCENT_RATE>
<MANUAL_CLIMB_AIRSPD><AIRSPD>
<TRUE_AIRSPD>350</TRUE_AIRSPD>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPD>
</MANUAL_CLIMB_AIRSPD>
<MANUAL_DESCENT_AIRSPD><AIRSPD>
<TRUE_AIRSPD>350</TRUE_AIRSPD>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPD>
</MANUAL_DESCENT_AIRSPD>
<DESCENT_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>4</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>4028</ID>

```

```

</FPM_MODE_INTENT>
</DESCENT_FPM_MODE_INTENT>
<CLIMB_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>2013</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>969</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>1083</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CLIMB_FPM_MODE_INTENT>
<MANUAL_CLIMB_RATE>1000</MANUAL_CLIMB_RATE>
</FPM_INTENT>
</INTENT>
<INTENT>
<TURN_INTENT>
<TURN_SETTING>
<TURN_SETTING_TYPE>TURN_BANK</TURN_SETTING_TYPE>
<VALUE>0</VALUE>
</TURN_SETTING>
<TURN_TYPE>TURN_SQUARE</TURN_TYPE>
</TURN_INTENT>
<COMMANDED_LEG_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>444</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</COMMANDED_LEG_AIRSPEED>
<CLIMB_DESCENT_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CLIMB_DESCENT_WIND>
<TEMPERATURE>
<VALUE>-28.582</VALUE>
<TEMPERATURE_TYPE>C</TEMPERATURE_TYPE>
</TEMPERATURE>
<ID>1882594752</ID>
<COMMANDED_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>22000</VALUE>
</ALTITUDE>
</COMMANDED_ALTITUDE>
<FUELFLOW_INCREMENT_PERCENTAGE>1</FUELFLOW_INCREMENT_PERCENTAGE>
<RESET_HACK_TIME_FLAG>FALSE</RESET_HACK_TIME_FLAG>

```

```

<COMMANDED_EXIT_ROUTE_POINT_REFERENCE>1890253365</COMMANDED_EXIT_ROUTE_P
OINT_REFERENCE>
<COMMANDED_ENTRY_ROUTE_POINT_REFERENCE>69807834</COMMANDED_ENTRY_ROUTE_P
OINT_REFERENCE>
<CRUISE_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CRUISE_WIND>
<DELTA_GROSS_WEIGHT>0</DELTA_GROSS_WEIGHT>
<DELTA_DRAG>0</DELTA_DRAG>
<FPM_INTENT>
<CRUISE_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>3059</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CRUISE_FPM_MODE_INTENT>
<MANUAL_DESCENT_RATE>1000</MANUAL_DESCENT_RATE>
<MANUAL_CLIMB_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_CLIMB_AIRSPEED>
<MANUAL_DESCENT_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_DESCENT_AIRSPEED>
<DESCENT_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>4</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>4028</ID>
</FPM_MODE_INTENT>
</DESCENT_FPM_MODE_INTENT>
<CLIMB_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>2013</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>969</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>1083</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>673</ID>

```

```

<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CLIMB_FPM_MODE_INTENT>
<MANUAL_CLIMB_RATE>1000</MANUAL_CLIMB_RATE>
</FPM_INTENT>
</INTENT>
<INTENT>
<TURN_INTENT>
<TURN_SETTING>
<TURN_SETTING_TYPE>TURN_BANK</TURN_SETTING_TYPE>
<VALUE>0</VALUE>
</TURN_SETTING>
<TURN_TYPE>TURN_SQUARE</TURN_TYPE>
</TURN_INTENT>
<COMMANDED_LEG_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>444</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</COMMANDED_LEG_AIRSPEED>
<CLIMB_DESCENT_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CLIMB_DESCENT_WIND>
<TEMPERATURE>
<VALUE>-28.582</VALUE>
<TEMPERATURE_TYPE>C</TEMPERATURE_TYPE>
</TEMPERATURE>
<ID>1882594753</ID>
<COMMANDED_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>22000</VALUE>
</ALTITUDE>
</COMMANDED_ALTITUDE>
<FUELFLOW_INCREMENT_PERCENTAGE>1</FUELFLOW_INCREMENT_PERCENTAGE>
<RESET_HACK_TIME_FLAG>FALSE</RESET_HACK_TIME_FLAG>
<COMMANDED_EXIT_ROUTE_POINT_REFERENCE>121291812</COMMANDED_EXIT_ROUTE_PO
INT_REFERENCE>
<COMMANDED_ENTRY_ROUTE_POINT_REFERENCE>1890253365</COMMANDED_ENTRY_ROUTE
_POINT_REFERENCE>
<CRUISE_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CRUISE_WIND>
<DELTA_GROSS_WEIGHT>0</DELTA_GROSS_WEIGHT>
<DELTA_DRAG>0</DELTA_DRAG>
<FPM_INTENT>
<CRUISE_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>3059</ID>

```

```

<INPUT_LIST><FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CRUISE_FPM_MODE_INTENT>
<MANUAL_DESCENT_RATE>1000</MANUAL_DESCENT_RATE>
<MANUAL_CLIMB_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_CLIMB_AIRSPEED>
<MANUAL_DESCENT_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_DESCENT_AIRSPEED>
<DESCENT_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>4</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>4028</ID>
</FPM_MODE_INTENT>
</DESCENT_FPM_MODE_INTENT>
<CLIMB_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>2013</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>969</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>1083</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CLIMB_FPM_MODE_INTENT>
<MANUAL_CLIMB_RATE>1000</MANUAL_CLIMB_RATE>
</FPM_INTENT>
</INTENT>
<INTENT>
<TURN_INTENT>
<TURN_SETTING>
<TURN_SETTING_TYPE>TURN_BANK</TURN_SETTING_TYPE>
<VALUE>0</VALUE>
</TURN_SETTING>
<TURN_TYPE>TURN_SQUARE</TURN_TYPE>
</TURN_INTENT>

```

```

<COMMANDED_LEG_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>444</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</COMMANDED_LEG_AIRSPEED>
<CLIMB_DESCENT_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CLIMB_DESCENT_WIND>
<TEMPERATURE>
<VALUE>-28.582</VALUE>
<TEMPERATURE_TYPE>C</TEMPERATURE_TYPE>
</TEMPERATURE>
<ID>1882594754</ID>
<COMMANDED_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>22000</VALUE>
</ALTITUDE>
</COMMANDED_ALTITUDE>
<FUELFLOW_INCREMENT_PERCENTAGE>1</FUELFLOW_INCREMENT_PERCENTAGE>
<RESET_HACK_TIME_FLAG>FALSE</RESET_HACK_TIME_FLAG>
<COMMANDED_EXIT_ROUTE_POINT_REFERENCE>121291812</COMMANDED_EXIT_ROUTE_PO
INT_REFERENCE>
<COMMANDED_ENTRY_ROUTE_POINT_REFERENCE>121291812</COMMANDED_ENTRY_ROUTE_
POINT_REFERENCE>
<CRUISE_WIND><WIND>
<SPEED>0</SPEED>
<DIRECTION>0</DIRECTION>
</WIND>
</CRUISE_WIND>
<DELTA_GROSS_WEIGHT>0</DELTA_GROSS_WEIGHT>
<DELTA_DRAG>0</DELTA_DRAG>
<FPM_INTENT>
<CRUISE_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>3059</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CRUISE_FPM_MODE_INTENT>
<MANUAL_DESCENT_RATE>1000</MANUAL_DESCENT_RATE>
<MANUAL_CLIMB_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>
</AIRSPEED>
</MANUAL_CLIMB_AIRSPEED>
<MANUAL_DESCENT_AIRSPEED><AIRSPEED>
<TRUE_AIRSPEED>350</TRUE_AIRSPEED>
<INPUT_TYPE>KTAS</INPUT_TYPE>

```

```
</AIRSPEED>
</MANUAL_DESCENT_AIRSPEED>
<DESCENT_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>4</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>4028</ID>
</FPM_MODE_INTENT>
</DESCENT_FPM_MODE_INTENT>
<CLIMB_FPM_MODE_INTENT><FPM_MODE_INTENT>
<FLAG>1</FLAG>
<MANUAL_FUELFLOW>1000</MANUAL_FUELFLOW>
<ID>2013</ID>
<INPUT_LIST><FPM_INPUT_PAIR>
<ID>969</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>1083</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
<FPM_INPUT_PAIR>
<ID>673</ID>
<VALUE>0</VALUE>
</FPM_INPUT_PAIR>
</INPUT_LIST>
</FPM_MODE_INTENT>
</CLIMB_FPM_MODE_INTENT>
<MANUAL_CLIMB_RATE>1000</MANUAL_CLIMB_RATE>
</FPM_INTENT>
</INTENT>
</INTENT_LIST>
<STATE_LIST><STATE>
<ID>1882594750</ID>
<ROUTE_POINT_REFERENCE>1207629152</ROUTE_POINT_REFERENCE>
<STORE_WEIGHT>0</STORE_WEIGHT>
</STATE>
<STATE>
<ID>1882594751</ID>
<ROUTE_POINT_REFERENCE>830684835</ROUTE_POINT_REFERENCE>
</STATE>
<STATE>
<ID>1882594752</ID>
<ROUTE_POINT_REFERENCE>69807834</ROUTE_POINT_REFERENCE>
</STATE>
<STATE>
<ID>1882594753</ID>
<ROUTE_POINT_REFERENCE>1890253365</ROUTE_POINT_REFERENCE>
</STATE>
<STATE>
<ID>1882594754</ID>
<ROUTE_POINT_REFERENCE>121291812</ROUTE_POINT_REFERENCE>
</STATE>
<STATE>
<ID>1882594755</ID>
```

```

<ROUTE_POINT_REFERENCE>121291812</ROUTE_POINT_REFERENCE>
</STATE>
</STATE_LIST>
<POINT_LIST><POINT>
<ID>1882594750</ID>
<MAGNETIC_VARIATION>-12.2</MAGNETIC_VARIATION>
<DESCRIPTION>LUKE AFB</DESCRIPTION>
<DATABASE_LOOKUP_STRING>US16978</DATABASE_LOOKUP_STRING>
<ELEVATION>
<VALUE>1090</VALUE>
<SOURCE>DAFIF</SOURCE>
</ELEVATION>
<WGS84_POSITION>
<LONGITUDE>W1122259.00</LONGITUDE>
<LATITUDE>N33 3206.00</LATITUDE>
</WGS84_POSITION>
<SOURCE>AIRPORT</SOURCE>
<NAME>KLUF/A</NAME>
<INPUT_DATUM>WGS84</INPUT_DATUM>
</POINT>
<POINT>
<ID>1882594751</ID>
<MAGNETIC_VARIATION>-11.5</MAGNETIC_VARIATION>
<DESCRIPTION>TUCSON</DESCRIPTION>
<DATABASE_LOOKUP_STRING>TUS, 1,2,US</DATABASE_LOOKUP_STRING>
<ELEVATION>
<VALUE>2672</VALUE>
<SOURCE>DAFIF</SOURCE>
</ELEVATION>
<WGS84_POSITION>
<LONGITUDE>W1105453.53</LONGITUDE>
<LATITUDE>N32 0542.68</LATITUDE>
</WGS84_POSITION>
<SOURCE>NAVAID</SOURCE>
<NAME>TUS/R</NAME>
<INPUT_DATUM>WGS84</INPUT_DATUM>
</POINT>
<POINT>
<ID>1882594752</ID>
<MAGNETIC_VARIATION>-12.1</MAGNETIC_VARIATION>
<DESCRIPTION>GILA BEND</DESCRIPTION>
<DATABASE_LOOKUP_STRING>GBN, 1,2,US</DATABASE_LOOKUP_STRING>
<ELEVATION>
<VALUE>790</VALUE>
<SOURCE>DAFIF</SOURCE>
</ELEVATION>
<WGS84_POSITION>
<LONGITUDE>W1124027.38</LONGITUDE>
<LATITUDE>N32 5722.53</LATITUDE>
</WGS84_POSITION>
<SOURCE>NAVAID</SOURCE>
<NAME>GBN/R</NAME>
<INPUT_DATUM>WGS84</INPUT_DATUM>
</POINT>

```

```

<POINT>
<ID>1882594753</ID>
<MAGNETIC_VARIATION>-12</MAGNETIC_VARIATION>
<DESCRIPTION>PHOENIX SKY HARBOR INTL</DESCRIPTION>
<DATABASE_LOOKUP_STRING>US08510</DATABASE_LOOKUP_STRING>
<ELEVATION>
<VALUE>1135</VALUE>
<SOURCE>DAFIF</SOURCE>
</ELEVATION>
<WGS84_POSITION>
<LONGITUDE>W1120029.00</LONGITUDE>
<LATITUDE>N33 2603.00</LATITUDE>
</WGS84_POSITION>
<SOURCE>AIRPORT</SOURCE>
<NAME>KPHX/A</NAME>
<INPUT_DATUM>WGS84</INPUT_DATUM>
</POINT>
<POINT>
<ID>1882594754</ID>
<MAGNETIC_VARIATION>-12.2</MAGNETIC_VARIATION>
<DESCRIPTION>LUKE AFB</DESCRIPTION>
<DATABASE_LOOKUP_STRING>US16978</DATABASE_LOOKUP_STRING>
<ELEVATION>
<VALUE>1090</VALUE>
<SOURCE>DAFIF</SOURCE>
</ELEVATION>
<WGS84_POSITION>
<LONGITUDE>W1122259.00</LONGITUDE>
<LATITUDE>N33 3206.00</LATITUDE>
</WGS84_POSITION>
<SOURCE>AIRPORT</SOURCE>
<NAME>KLUF/A</NAME>
<INPUT_DATUM>WGS84</INPUT_DATUM>
</POINT>
</POINT_LIST>
<VEHICLE_REFERENCE>1</VEHICLE_REFERENCE>
<NAME>sample.rte</NAME>
<DEPARTURE_DATE>19981027</DEPARTURE_DATE>
<ID>1882594749</ID>
<RECOVERY_FUEL>0</RECOVERY_FUEL>
<PATH_LIST><PATH>
<TRANSITION_LIST><TRANSITION>
<END_STATE_REFERENCE>1882594751</END_STATE_REFERENCE>
<CALCULATION_FLAG>FALSE</CALCULATION_FLAG>
<MINIMUM_SAFE_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>AGL</ALTITUDE_TYPE>
<VALUE>200</VALUE>
</ALTITUDE>
</MINIMUM_SAFE_ALTITUDE>
<INTENT_REFERENCE>1882594750</INTENT_REFERENCE>
<START_STATE_REFERENCE>1882594750</START_STATE_REFERENCE>
<ID>1882594750</ID>
</TRANSITION>
<TRANSITION>

```

```

<END_STATE_REFERENCE>1882594752</END_STATE_REFERENCE>
<CALCULATION_FLAG>FALSE</CALCULATION_FLAG>
<MINIMUM_SAFE_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>0</VALUE>
</ALTITUDE>
</MINIMUM_SAFE_ALTITUDE>
<INTENT_REFERENCE>1882594751</INTENT_REFERENCE>
<START_STATE_REFERENCE>1882594751</START_STATE_REFERENCE>
<ID>1882594751</ID>
</TRANSITION>
<TRANSITION>
<END_STATE_REFERENCE>1882594753</END_STATE_REFERENCE>
<CALCULATION_FLAG>FALSE</CALCULATION_FLAG>
<MINIMUM_SAFE_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>0</VALUE>
</ALTITUDE>
</MINIMUM_SAFE_ALTITUDE>
<INTENT_REFERENCE>1882594752</INTENT_REFERENCE>
<START_STATE_REFERENCE>1882594752</START_STATE_REFERENCE>
<ID>1882594752</ID>
</TRANSITION>
<TRANSITION>
<END_STATE_REFERENCE>1882594754</END_STATE_REFERENCE>
<CALCULATION_FLAG>FALSE</CALCULATION_FLAG>
<MINIMUM_SAFE_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>0</VALUE>
</ALTITUDE>
</MINIMUM_SAFE_ALTITUDE>
<INTENT_REFERENCE>1882594753</INTENT_REFERENCE>
<START_STATE_REFERENCE>1882594753</START_STATE_REFERENCE>
<ID>1882594753</ID>
</TRANSITION>
<TRANSITION>
<END_STATE_REFERENCE>1882594755</END_STATE_REFERENCE>
<CALCULATION_FLAG>FALSE</CALCULATION_FLAG>
<MINIMUM_SAFE_ALTITUDE><ALTITUDE>
<ALTITUDE_TYPE>MSL</ALTITUDE_TYPE>
<VALUE>0</VALUE>
</ALTITUDE>
</MINIMUM_SAFE_ALTITUDE>
<INTENT_REFERENCE>1882594754</INTENT_REFERENCE>
<START_STATE_REFERENCE>1882594754</START_STATE_REFERENCE>
<ID>1882594754</ID>
</TRANSITION>
</TRANSITION_LIST>
<ID>1882594750</ID>
<HEADING_CALCULATION_OPTION>GREAT_CIRCLE</HEADING_CALCULATION_OPTION>
<MAGNETIC_HEADING_CALCULATION_OPTION>START</MAGNETIC_HEADING_CALCULATION
_OPTION>
<EMERGENCY_SAFE_AGL_ALTITUDE>0</EMERGENCY_SAFE_AGL_ALTITUDE>
<BINGO_FUEL>1000</BINGO_FUEL>

```

```
<PATH_TYPE>STANDARD</PATH_TYPE>
</PATH>
</PATH_LIST>
<ROUTE_POINT_LIST><ROUTE_POINT>
<POINT_REFERENCE>1882594750</POINT_REFERENCE>
<ID>1207629152</ID>
</ROUTE_POINT>
<ROUTE_POINT>
<POINT_REFERENCE>1882594751</POINT_REFERENCE>
<ID>830684835</ID>
</ROUTE_POINT>
<ROUTE_POINT>
<POINT_REFERENCE>1882594752</POINT_REFERENCE>
<ID>69807834</ID>
</ROUTE_POINT>
<ROUTE_POINT>
<POINT_REFERENCE>1882594753</POINT_REFERENCE>
<ID>1890253365</ID>
</ROUTE_POINT>
<ROUTE_POINT>
<POINT_REFERENCE>1882594754</POINT_REFERENCE>
<ID>121291812</ID>
</ROUTE_POINT>
</ROUTE_POINT_LIST>
<TIME_ZONE_DESIGNATION>0</TIME_ZONE_DESIGNATION>
</ROUTE>
</ROUTE_LIST>
<VEHICLE_LIST><VEHICLE>
<CONFIGURATION_ITEM>
<INTERNAL_FUEL_WEIGHT>6700</INTERNAL_FUEL_WEIGHT>
<EMPTY_WEIGHT>19500</EMPTY_WEIGHT>
<NAME>F-16CD (PW 220) Block 32</NAME>
<INTERNAL_FUEL_CAPACITY>6700</INTERNAL_FUEL_CAPACITY>
<DRAG>15</DRAG>
<ID>1</ID>
<EXTERNAL_FUEL_CAPACITY>4000</EXTERNAL_FUEL_CAPACITY>
<EXTERNAL_FUEL_WEIGHT>4000</EXTERNAL_FUEL_WEIGHT>
</CONFIGURATION_ITEM>
<FUEL_TYPE>JP4</FUEL_TYPE>
<FPM_VERSION>4.04</FPM_VERSION>
<MINIMUM_FUEL>800</MINIMUM_FUEL>
<FPM_VEHICLE_ID>5</FPM_VEHICLE_ID>
<ID>1</ID>
</VEHICLE>
</VEHICLE_LIST>
</MISSION>
</MISSION_LIST>
</CRD>
```

