

C2SIM Server Message Flow

Main Message Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIMServer	postC2SIM	Receive XML MSG and parameters from client
		Create C2SIMTransaction
		checkVersion
		Check if MSG has already been handled (Multiserver Environment)
		Log MSG to Replay Log
C2SIMServerProcess	processMessage	Check processing options
	processManually	Remove header from C2SIM/CWIX Messages
		Identify message type
		Log MSG and Type to Debug Log
		Pass message through Cyber. Drop if directed
	checkMessageState	Check if this type message permitted in this state
	processManually	Call specific Class.process() depending on message type if translation is to be performed: C2SIM_C2SIM.process C2SIM_IBML09.process C2SIM_CBML.process C2SIM_CWIX.process C2SIM_MSDL.process
		Log message and identification to debugLogger

C2SIM Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIM_C2SIM	process	Determine type of message and call appropriate method. If message type not identified publish message as is.
C2SIM_C2SIM	process_C2SIM_Initialization	Locate Initialization elements. Save each element (and children) in appropriate internal list: InitializationDataFile ObjectDefinitions/AbstractObject/ForceSide ObjectDefinitions/Action ObjectDefinitions/Entity ... /MilitaryOrganization ObjectDefinitions/PlanPhaseReference SystemEntityList ScenarioSetting
C2SIM_C2SIM	process_C2SIM_Initialization	Build cross reference table for Name and UUID
	c2SIM_Translate_11To9	Translate C2SIM Version 11 to Version 9
	c2SIM_Translate_9To11	Translate C2SIM Version 9 to Version 11
C2SIM_C2SIM	process_C2SIM_Order	
C2SIM_Server_STOMP	publishMessage	publish C2SIM Order as received
C2SIM_Mapping	mapMessage	Map C2SIM Order to Core Order
C2SIM_C2SIM	translateC2SIMLocationToCore	Translate location information to core
C2SIM_IBML09	translateCoreToIBML09	Translate order to IBML09 and publish
C2SIM_CWIX	translateCoreToCWIX	Translate order to CWIX and publish
C2SIM_CBML	translateCoreToCBML	Translate order to CBML and publish
C2SIM_C2SIM	process_C2SIM_Report	
C2SIM_Server_STOMP	publishMessage	publish C2SIM Report as received
C2SIM_C2SIM	Process_C2SIM_Report	Loop through message processing one report at a time. Don't translate AutonomousSystemPositionReport.
C2SIM_Mapping	mapMessage	Map Report to Core Report (Unit)
	Process_C2SIM_Report	Add EntityHealthStatus and OperationalStatusCode
C2SIM_C2SIM	UpdatePosition	Update position in table from report
C2SIM_IBML09	translateCoreToIBML09_GSR	Translate report to IBML09 and publish
C2SIM_CBML	translateCoreToCBML_GSR	Translate report to CBML and publish

IBML09 Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIM_IBML09	process	Check messageDescriptor to determine processing
	processIBML09_Order	
C2SIM_Server_STOMP	publishMessage	Publish IBML09 Order as received
C2SIM_Mapping	mapMessage	mapMessage - Map IBML09 Order to Core Order
C2SIM_IBML09	process	taskerWho -> UUID
		Loop through Tasks
		taskerWho -> UUID and affectedWho -> UUID
	convertOrderDates	Convert dates in task to ISO format
	translateIBML09LocationToCore	Translate location to core
C2SIM_C2SIM	translateCoreToC2SIM_Order	Translate the core order to C2SIM and publish
C2SIM_CWIX	translateCoreToCWIX_Order	Translate the core order to CWIX and publish
C2SIM_CBML	translateCoreToCBML_Order	Translate the core order to CBML and publish
	processIBML09_GSR	
C2SIM_Server_STOMP	publishMessage	Publish IBML09 Report as received
	processIBML09_GSR	Loop through reports in message
C2SIM_Mapping	mapMessage	Map GSR to Unit (Core position report)
		ID in message -> UUID
		ReporterWho Name -> UUID
	convertReportDate	MIP Dates -> ISO Dates
C2SIM_C2SIM	updatePosition	Update Unit Position
C2SIM_CWIX	translateCoreToCWIX_PR	Translate report to CWIX and publish
C2SIM_C2SIM	translateCoreToC2SIM_PR	Translate report to C2SIM and publish
C2SIM_CBML	translateCoreToCBML_GSR	Translate report to CBML and publish

CBML Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIM_CWIX	Process	Check messageDescriptor to determine processing
	processCBML_Order	
C2SIM_Server_STOMP	publishMessage	Publish CBML Order as received
C2SIM_Mapping	mapMessage	mapMessage - Map CWIX Order to Core Order
	process CBML_Order	Loop through Tasks
	convertOrderDates	Convert dates in task to MIP format
	translate CBML LocationToCore	Translate location to core
C2SIM_C2SIM	translateCoreToC2SIM_Order	Translate the core order to C2SIM and publish
C2SIM_CWIX	translateCoreToCWIX_Order	Translate the core order to CWIX and publish
C2SIM_IBML09	translateCoreToIBML09_Order	Translate the core order to IBML09 and publish
	processCBML_GSR	
C2SIM_Server_STOMP	publishMessage	Publish CBML Report as received
		Loop through reports in message
C2SIM_Mapping	mapMessage	Map GSR to Unit (Core position report)
C2SIM_IBML09	convertReportDate	ISO Date -> MIP Date
C2SIM_C2SIM	updatePosition	Update Unit Position
C2CIM_CWIX	processCWIX_GSR	PositionReport -> GeneralStatusReport
C2SIM_CWIX	translateCoreToCWIX_PR	Translate report to CWIX and publish
C2SIM_C2SIM	translateCoreToC2SIM_PR	Translate report to C2SIM and publish
C2SIM_IBML09	translateCoreToIBML09_GSR	Translate report to IBML09 and publish

MSDL Translation Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIM_MSDL	process	Translate MSDL to C2SIM. Add to accumulated C2SIM Initialization Data
	addMSDLToC2SIM	Translate MSDL to C2SIM. Add to accumulated C2SIM Initialization Data
	convertC2SIMToMSDL	Translate accumulated Initialization Data to MSDL

Command Processing

Class	Method	Function
C2SIM_Server	postCommand	Receive Command and parameters from client
		Create C2SIM_Transaction
		Log Command to replayLogger
		Log command information to debugLogger
C2SIM_Command	unitCommandProcess	switch Statement to execute command
		All commands verifyPassword; checkCommandState
		START: RUNNING -> sessionState
		INITIALIZE: Publish "SubmitInitialization"; INITIALIZING -> sessionState
		STOP: Publish "StopScenario"; INITIALIZED -> sessionState
		RESET: Clear Initialization Tables; Publish "ResetScenario" UNINITIALIZED -> sessionState
		PAUSE: Publish "PauseScenario"; PAUSED -> sessionState
		SHARE:
C2SIM_C2SIM	shareC2SIM	Build C2SIMInitializationBody from Initialization tables and publish
		Translate C2SIMInitializationBody to MSDL and publish
		Publish "InitializationComplete
		INITIALIZED -> sessionState
		STATUS: Return Status message
		LOAD: Load external file and process as if received as ObjectInitialization message

STOMP Publishing Processing

<u>Class</u>	<u>Method</u>	<u>Function</u>
C2SIM_Server_STOMP	publishMessage	
		Build STOMP headers from contents of C2SIM_Transaction
		If C2SIM build headers with C2SIM header information
		Log message to replayLogger
		Log message to debugLogger
C2SIMClientSTOMP_Lib	publish	Compute length and create content-length header
	sendFrame	Send frame on STOMP socket