8.12 Simple Service Messages

In many service-oriented designs, a one-to-one message exchange will occur between the consumer and producer of a service. That is, the consumer will make a request of the producer of the product or service, and the producer will, in turn, reply with a response. Many such message exchanges are possible using a complementary pair of Request and Response type messages. The pair of Simple Service Messages is a general mechanism for the invocation of a service from one component to another. Software components wishing to expose or make certain services available to other components can utilize this general mechanism for that purpose.

The Simple Service Messages are similar in nature and function to the Directive Messages. Where the Directive Request Message will use a keyword or text string to request some functionality of a component, the Simple Service Request Message will make a service request by name, number, and/or operation. It will also pass in any parameters that are required. The Simple Service Messages do not provide for files to be included in the messages though a block of data can be returned in the Simple Service Response message. Also, it is expected that the number of services offered by a component will be small enough that name or number can easily identify them.

The intention of the Simple Service Messages is to allow a component to offer a small number of simple services to other components and the system in general. They are not meant to provide a comprehensive service framework. Services are expected to be provided locally and not across domains, thereby allowing (or requiring) for all provided services to be uniquely named within the immediate service area. Thus, any component could request any service, by name, offered by another component, if authorized, within the local domain. Because a service name may be used as a subject name element (*ME1*) it must follow the same syntax as elements in a message subject.

8.12.1 Simple Service Request Message

A Simple Service Request Message is a service request that is issued to one application from another. A service request may also be input from a user through a GUI or command line, or as part of the internal logic of a component. Services could also be grouped together in a procedure (or proc), an executable schedule, or other such orchestration techniques. As components become less coupled, they may tend to offer or provide more services and thus enable more rapid software development with orchestrated modules. The Simple Service Request Message will request the invocation of a single service from a single component.

8.12.1.1 Simple Service Request Message Subjects

In most request/response message exchange patterns, the *ME1* element is used to identify the requestor and responder of the request. With the Simple Service messages, the *ME1* element may also be used to identify the service. That is, the unique name of the service (following the

syntax of the message subject name elements) is inserted into element *ME1*. When the name of the service is inserted into the *ME1* element, the message subject elements TYPE, SUBTYPE, and *ME1* would appear as follows:

… REQ.SERV.[SERVICENAME]
and
… RESP.SERV.[SERVICENAME] …

The above syntax shows the message subject for a service request message and service response message. In service terminology, the requestor is known as the consumer of the service and the responder is known as the producer or provider. This message subject syntax allows the consumer to request a service without knowledge of the name of the component providing the service. The provider of the service should respond in kind using the service name in the *ME1* element. As should be obvious, the producer must subscribe not only to messages subjects using *ME1* as a component name, but also to message subjects using *ME1* as a service name. *If a single producer provides a large number of services, it will require an equally large number of message subject subscriptions to manage, and this convention may not be desirable.*

Services are expected to be provided locally and not across domains, thereby allowing for all provided services to be uniquely named within the immediate service area. If the service cannot be uniquely named within the service area, then the *ME1* and *ME2* elements can be employed together to uniquely identify all services, similar to the mission-satellite and message type-subtype pairings. The *ME2* element will serve as the general subject matter or group, and the *ME1* element will identify the service, uniquely named, within that group.

Since a service name may be used as a subject name element it must follow the same syntax as elements in a message subject. See **Error! Reference source not found. Error! Reference source not found.**.

In order to distinguish service names from component names, naming conventions may be established. For example, all services could be named as “S_NNNN”, and/or all components could be named “C_NNNN”.

Table 8-157. Simple Service Request Message Subject Naming

	Subject Standard	Domain Elements		Mission Elements			Message Elements		*Miscellaneous Elements*					
Subject Element	**Speci-fication**	**DOMAIN1**	**DOMAIN2**	**MISSION**	**CONST**	**SAT**	**TYP**	**SUBTYP**	***ME1***	***ME2***	***ME3***	***ME4***	***ME5***	***ME6***
Subject Content	C2MS	[domain 1 – system specific]	[domain 2 – system specific]	[mission]	[constellation]	[sat]	REQ	SERV	DESTINATION-COMPONENT or [service name]	[service group]	[operation]			
Example for Publisher / Sender	C2MS	DOM1	DOM2	MSSN	CNS1	SAT1	REQ	SERV	FD	DAY304	FD			
Example for Publisher / Sender	C2MS	DOM1	DOM2	MSSN	CNS1	SAT1	REQ	SERV	FD	MAN55	FD			
Example for Subscriber / Receiver	C2MS	DOM1	DOM2	*	*	SAT1	REQ	SERV	FD	*	FD			

Table Error! No text of specified style in document.**-158. Properties of the Miscellaneous Elements for the Simple Service Request Message**

Miscellaneous Element	Required / Optional	Description	Field in Msg, if applicable
ME1	Required	Component name of producer, or name of the service	DESTINATION-COMPONENT from header; or SERVICE-NAME
ME2	Optional	Functional arena, subject matter, or group the service belongs to	N/A
ME3	Optional	Name of the operation within the service	OPERATION-NAME
ME4 …	Not used		

The *ME2* and *ME3* elements serve as a two-element pair to uniquely identify all services, similar to the mission-satellite and message type-subtype pair. Should the name of the service not be unique, the *ME2* element can be utilized to avoid ambiguity.

Examples:
Two components, APP4 and APP1, interact with the Simple Service Request Message. APP4 sends a Simple Service Request Message to APP1.

APP4 subject to send a Simple Service Request for a particular service to APP1:

`C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.APP1`

`C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.SERVICEA` (using the service name convention in *ME1*)
`C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.SERVICEA.GROUPB` (using the optional service name convention of *ME1* and *ME2*)
`C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.SERVICEA.GROUPB.OPERATION1` (using all the *ME* elements)

APP1 message subject subscription to receive a request for a particular service using the service name convention in *ME1 (*assumes all services are uniquely named):

`C2MS.*.*.*.*.*.*.*.SERVICEA.>`

APP1 message subject subscription to receive a request for a particular service when service names are not unique, using the *ME1* and *ME2* elements.

`C2MS.*.*.*.*.*.*.*.SERVICEA.GROUPB.>`

APP1 message subject subscription to receive a request for a particular operation within a service by using all the subject elements.

`C2MS.*.*.*.*.*.*.*.SERVICEA.GROUPB.OPERATION1`

APP1 message subject subscription to receive requests for any of its provided services when the service naming convention is not used:

```
C2MS.*.*.*.*.*.REQ.SERV.APP1
```

APP1 subjects to receive all requests for any of its services:

```
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.APP1.>
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.SERVICEA.>
C2MS.*.*.*.*.*.REQ.SERV.SERVICEB.>
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.REQ.SERV.SERVICEC.>
```

and so on for each service.

8.12.1.2 Simple Service Request Message Header

The abbreviated Table 8-159 below shows the required values of the MESSAGE-TYPE and MESSAGE-SUBTYPE fields for the Simple Service Request Message header.

Table Error! No text of specified style in document.**-159. Simple Service Request Message Header**

Field Name	Value	Notes
HEADER-VERSION	2019	Version Number for this message description
MESSAGE-TYPE	REQ	Message type identifier: REQ, RESP, or MSG
MESSAGE-SUBTYPE	SERV	Unique message identifier, fixed for C2MS Standard Messages
More … Please refer to **Error! Reference source not found. Error! Reference source not found.** for a complete definition.	…	…

8.12.1.3 Simple Service Request Message Contents

Figure 8-33 below shows a UML object diagram of the Simple Service Request Message with its required and optional fields.

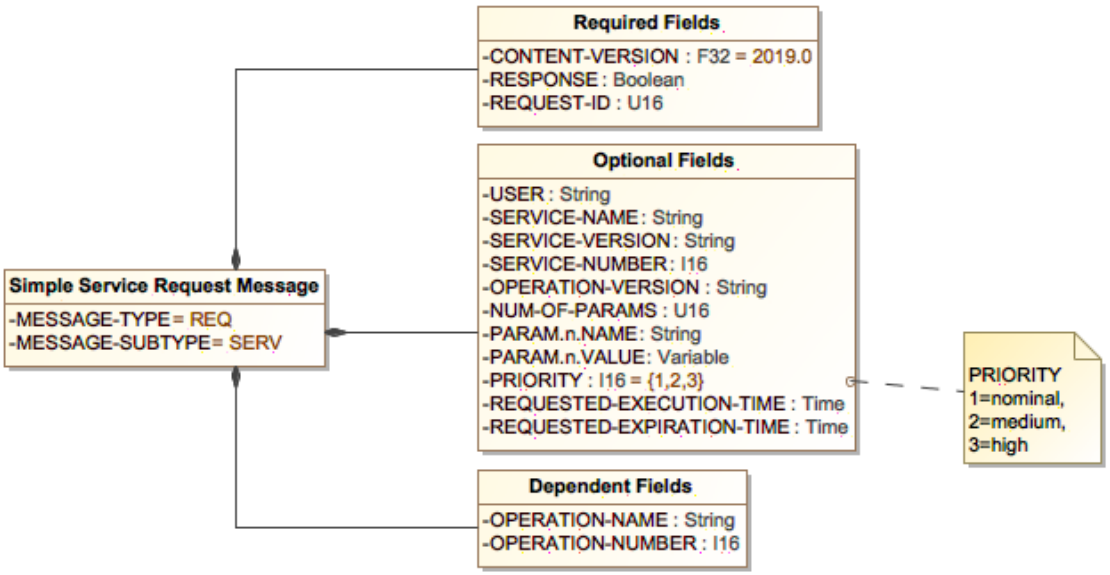

Figure Error! No text of specified style in document.**-33. Simple Service Request Message Diagram**

Table 8-160 below describes additional field names, values, and notes for the Simple Service Request Message.

Table Error! No text of specified style in document.**-160. Simple Service Request Message Additional Information**

Field Name	Value/Description		Notes
CONTENT-VERSION	2019		Version Number for this message content description
RESPONSE	**Value**	**Description**	Indicates if a response is required
	0	No / False	
	1	Yes / True	
REQUEST-ID			ID to identify the request message – if different request messages have the same value, the request is a replacement; else, it is a new request
USER			Which user/workposition/proc/schedule the message is coming from
SERVICE-NAME			Name of the service offered by a component
SERVICE-VERSION			Version of the service
OPERATION-NAME			Name of the operation within the specified service

Field Name	Value/Description		Notes
OPERATION-NUMBER			Number of the operation within the specified service
OPERATION-VERSION			Version of the operation within the specified service
NUM-OF-PARAMS			Number of parameters included within the service request
PARAM.n.NAME			Name of the parameter
PARAM.n.VALUE			Value of the parameter; Component must ascertain the data type before accessing the value (e.g., with a function call)
PRIORITY	**Value**	**Description**	Indicates processing priority, if applicable
	1	Nominal	
	2	Medium	
	3	High	
REQUESTED-EXECUTION-TIME			Absolute or relative time can apply
REQUESTED-EXPIRATION-TIME			Absolute or relative time can apply

Services may evolve over time and services may offer a number of variations or operations for a particular service. A service may be identified by name or number. The requestor may choose either option. If there are a number of operations for that service, then the requestor must specify which operation, by name or number is being requested. When service and operation parameters are not specified, the default will be the first service and first operation within that service as determined by the provider.

The Simple Service Request Message can be used to:

1. Request a service
2. Provide an unsolicited service

To request a service and receive a response via the Simple Service Response Message, the requestor must mark the RESPONSE field as true.

The Simple Service Request Message can also be used to provide an unsolicited service. That is, an application can provide a set of data to other applications automatically, without them having to issue a request. The service provider simply populates the "Parameters" fields of the Simple Service Request Message with the data to be published. The application can also identify the service in the "Service Information" fields. Additionally, the provider of the unsolicited service will publish the message with the name of the service in the *ME1* element of the message subject. Applications wanting to avail themselves of the service will subscribe to the message subject for that service.

As an example, at the conclusion of a pass, an application will automatically provide a description of the pass that includes the start and stop times, the collection point, and the satellite. This data can be inserted into the "Parameter" fields along with the service name, say, 'PASSDESC", in the "SERVICE-NAME" field. Also, the "RESPONSE" field is set to false.

Then the message is published with 'PASSDESC" in the *ME1* element of the message subject. Applications wishing to receive this pass description data automatically can easily subscribe to this message and receive all pass descriptions whenever they are published.

Note that for unsolicited services, the Simple Service Request Message will be sent with a "Publish" message exchange pattern.

8.12.2 Simple Service Response Message

A Simple Service Response Message is sent by an application in response to a Simple Service Request Message. The Simple Service Response Message will provide acknowledgment of the Simple Service Request Message, a status of the action completed, and any data to be returned. A series of Simple Service Response Messages may be required in the case where the processing of the action is lengthy. An example of this would be a complex mathematical calculation using a large volume of data. In this event, an interim or interactive "working" type message would be issued to let the original application know that the action is still being processed. Please see Section **Error! Reference source not found. Error! Reference source not found.** for a general discussion on these types of messages.

8.12.2.1 Simple Service Response Message Subjects

Table 8-161. Simple Service Response Message Subject Naming

	Subject Standard	Domain Elements		Mission Elements			Message Elements		*Miscellaneous Elements*					
Subject Element	**Speci-fication**	**DOMAIN1**	**DOMAIN2**	**MISSION**	**CONST**	**SAT**	**TYP**	**SUBTYP**	***ME1***	***ME2***	***ME3***	***ME4***	***ME5***	***ME6***
Subject Content	C2MS	[domain 1 – system specific]	[domain 2 – system specific]	[mission]	[constellation]	[sat]	RESP	SERV	[DESTINATION-COMPONENT or service name]	[RESPONSE-STATUS]				
Example for Publisher / Sender	C2MS	DOM1	DOM2	MSSN	CNS1	SAT1	RESP	SERV	FD	DAY304				
Example for Publisher / Sender	C2MS	DOM1	DOM2	MSSN	CNS1	SAT1	RESP	SERV	FD	MAN55				
Example for Subscriber / Receiver	C2MS	DOM1	DOM2	*	*	SAT1	RESP	SERV	FD	*				

Table Error! No text of specified style in document.**-162. Properties of the *Miscellaneous Elements* for the Simple Service Response Message**

Miscellaneous Element	Required / Optional	Description	Field Origination in Msg, if applicable
ME1	Required	Component name of consumer or name of service	DESTINATION-COMPONENT from header, or "SERVICE-NAME" from content of Request msg
ME2	Required	Status type supplied by Responder	"RESPONSE-STATUS" from content of Response message

Examples:
Two components, APP4 and APP1, interact with the Simple Service Response message. APP1 sends a Simple Service Response Message to APP4.

APP1 subject to send the Simple Service Response to APP4:

```
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.RESP.SERV.APP4.1      or
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.RESP.SERV.SERVICEA.1
```

APP4 subscribes to receive its own Simple Service Response Messages:

```
C2MS.DOM1.DOM2.MSSN.CNS1.SAT1.RESP.SERV.APP4.>      or
C2MS.*.*.*.*.*.RESP.SERV.APP4.>
```

8.12.2.2 Simple Service Response Message Header

The abbreviated Table 8-163 below shows the required values of the MESSAGE-TYPE and MESSAGE-SUBTYPE fields for the Simple Service Response Message header.

Table Error! No text of specified style in document.**-163. Simple Service Response Message Header**

Field Name	Value	Notes
HEADER-VERSION	2019	Version Number for this message description
MESSAGE-TYPE	RESP	Message type identifier: REQ, RESP, or MSG
MESSAGE-SUBTYPE	SERV	Unique message identifier, fixed for C2MS Standard Messages
More … Please refer to **Error! Reference source not found. Error! Reference source not found.** for a complete definition.	…	…

8.12.2.3 Simple Service Response Message Contents

Table 8-34 below shows a UML object diagram of the Simple Service Response Message with its required and optional fields.

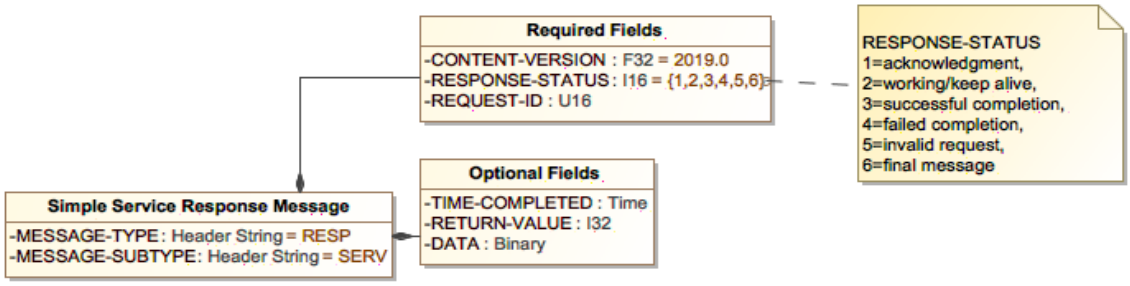

Figure Error! No text of specified style in document.**-34. Simple Service Response Message Diagram**

Table 8-164 below describes additional field names, values, and notes for the Simple Service Response Message.

Table Error! No text of specified style in document.**-164. Simple Service Response Message Additional Information**

Field Name	Value/Description		Notes
CONTENT-VERSION	2019		Version Number for this message content description
RESPONSE-STATUS	**Value**	**Description**	Identifies the status of the Simple Service being processed
	1	Acknowledgement	
	2	Working/keep alive	
	3	Successful completion	
	4	Failed completion	
	5	Invalid Request	
	6	Final Message	
REQUEST-ID			This field's value is to be the same as the REQUEST-ID in the associated REQ message.
TIME-COMPLETED			Time application completed processing the Simple Service
RETURN-VALUE			Return value or status based on the RESPONSE-STATUS. Provides additional status information particular to the request.
DATA			Additional data that may accompany the response