The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Practices

CROSS SUPPORT SERVICE MANAGEMENT— COMMON DATA ENTITIES

DRAFT RECOMMENDED PRACTICE

CCSDS 902.12-P-1.1

PINK BOOK
September 2022

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Practices

CROSS SUPPORT SERVICE MANAGEMENT— COMMON DATA ENTITIES

DRAFT RECOMMENDED PRACTICE

CCSDS 902.12-P-1.1

PINK BOOK
September 2022

AUTHORITY

Issue:	Pink Book, Issue 1.1
Date:	September 2022
Location:	Not Applicable

(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the email address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not in themselves considered binding on any Agency.

CCSDS Recommendations take two forms: **Recommended Standards** that are prescriptive and are the formal vehicles by which CCSDS Agencies create the standards that specify how elements of their space mission support infrastructure shall operate and interoperate with others; and **Recommended Practices** that are more descriptive in nature and are intended to provide general guidance about how to approach a particular problem associated with space mission support. This **Recommended Practice** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommended Practice** is entirely voluntary and does not imply a commitment by any Agency or organization to implement its recommendations in a prescriptive sense.

No later than five years from its date of issuance, this **Recommended Practice** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Practice** is issued, existing CCSDS-related member Practices and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such Practices or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new Practices and implementations towards the later version of the Recommended Practice.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Practice is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (BELSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Netherlands Space Office (NSO)/The Netherlands.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Practice. Its ‘Pink Book’ status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document’s technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	**Title**	**Date**	**Status**
CCSDS 902.12-M-1	Cross Support Service Management—Common Data Entities, Recommended Practice, Issue 1	February 2021	Original issue
CCSDS 902.12-P-1.1	Cross Support Service Management—Common Data Entities, Draft Recommended Practice, Issue 1.1	September 2022	Current proposed draft update

NOTE – Changes from the original issue are too numerous to permit meaningful markup.

CONTENTS

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

1.1.1 PURPOSE

The purpose of this Magenta Book is to specify the various data entities defined in the scope of service management that are applicable to more than one Service Management Blue Book. These data entities are referred to as 'Common' data entities.

1.1.2 SCOPE

The scope of this book is all of the Service Management Common Data Entities used in the various Service Management Recommended Standards and Practices.

Figure 1-1 puts the Service Management Common Data Entities into context with the various standards that together form the Space Communication Cross Support Service Management.

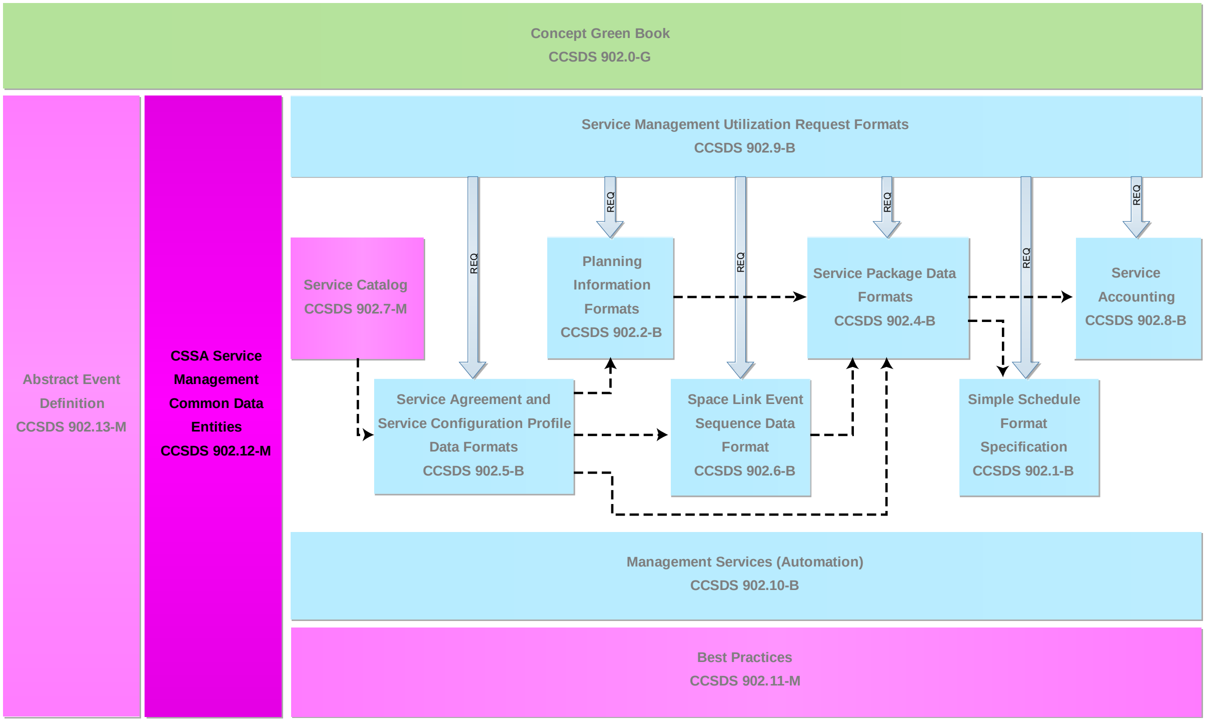
Concept Green Book
CCSDS 902.0-G
Abstract Event Definition CCSDS 902.13-M
CSSA Service Management Common Data Entities CCSDS 902.12-M
Service Management Utilization Request Formats
CCSDS 902.9-B
REQ
Service Catalog CCSDS 902.7-M
Planning Information Formats CCSDS 902.2-B
Service Package Data Formats CCSDS 902.4-B
Service Accounting CCSDS 902.8-B
Service Agreement and Service Configuration Profile Data Formats CCSDS 902.5-B
Space Link Event Sequence Data Format CCSDS 902.6-B
Simple Schedule Format Specification CCSDS 902.1-B
Management Services (Automation)
CCSDS 902.10-B
Best Practices
CCSDS 902.11-M

Figure 1-1: Service Management Common Data Entities in the Context of Space Communication Cross Support Service Management

1.2 APPLICABILITY

These common data entities are defined in the context of service management and are thus implicitly applicable to the various Service Management Recommended Standards and Practices.

1.3 RATIONALE

The rationale for this document is that there is some overlap between the various Service Management Recommended Standards and Practices in terms of commonality of data structures. This document is used to record these common data entities.

1.4 DOCUMENT STRUCTURE

This document is organized as follows:

a) Section 1 provides the purpose, scope, applicability, and rationale of this Recommended Practice and identifies the conventions and references used throughout the document. This section also describes how this document is organized. A brief description is provided for each section and annex so that the reader will have an idea of where information can be found in the document. It also identifies terminology that is used in this document but is defined elsewhere.

b) Section 2 provides a brief overview of the common data entities.

c) Section 3 provides details about the common data entities.

d) Annex A discusses security, Space Assigned Numbers Authority (SANA), and patent considerations.

e) Annex B provides an informative listing of the XML schema for the service management information entity header.

f) Annex C is a list of informative references.

g) Annex D contains a list of abbreviations and acronyms applicable to the Service Management Common Data Entities.

1.5 DEFINITIONS

For the purposes of this document, the following definition applies:

agency: A satellite operator or satellite service provider.

1.6 NOMENCLATURE

1.6.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Practice:

a) the words 'shall' and 'must' imply a binding and verifiable specification;

b) the word 'should' implies an optional, but desirable, specification;

c) the word 'may' implies an optional specification;

d) the words 'is', 'are', and 'will' imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.6.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is offset from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.7 CONVENTIONS

1.7.1 THE UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) diagrams used in the specification (including class diagrams, package diagrams, sequence diagrams, and activity diagrams) follow the notation, semantics, and conventions imposed by the Version 2.4.1 UML specification of the Object Management Group (OMG) (reference [2]).

1.7.2 XML SCHEMA DATATYPES

Many of the datatype definitions in this Recommended Practice are the same as definitions defined by XML Schema, Part 2 Datatypes (see reference [11]). Wherever an XML Schema datatype is indicated in this Recommended Practice, it is prefixed with the string 'xsd:'.

1.8 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.

[2] *Unified Modeling Language (UML)*. Version 2.4.1. Needham, Massachusetts: Object Management Group, August 2011.

[3] *CCSDS SANA Registry Management Policy*. Issue 2. CCSDS Record (Yellow Book), CCSDS 313.1-Y-2. Washington, D.C.: CCSDS, October 2020.

[4] *Space Assigned Numbers Authority (SANA)—Role, Responsibilities, Policies, and Procedures*. Issue 3. CCSDS Record (Yellow Book), CCSDS 313.0-Y-3. Washington, D.C.: CCSDS, October 2020.

[5] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.

[6] *Abstract Event Definition*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 902.13-M-1. Washington, D.C.: CCSDS, February 2021.

[7] “Planet and Satellite Names and Discoverers.” Gazetteer of Planetary Nomenclature. https://planetarynames.wr.usgs.gov/Page/Planets.

[8] “Minor Planet Names: Alphabetical List.” Minor Planet Center. https://minorplanetcenter.net//iau/lists/MPNames.html.

[9] B. A. Archinal, et al. “Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2015.” *Celestial Mechanics and Dynamical Astronomy* 130, no. 22 (March 2018).

[10] “Functional Resources.” Space Assigned Numbers Authority. http://sanaregistry.org/r/functional_resources/.

[11] David Peterson, et al., eds. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. Version 1.1. W3C Recommendation.

[12] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 0. Proposed Draft Recommendation for Space Data System Standards (Proposed Red Book), CCSDS 902.9-R-0. Washington, D.C.: CCSDS, August 2022.

[13] *XML Specification for Navigation Data Messages*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 505.0-B-2. Washington, D.C.: CCSDS, May 2021.

[14] *Delta-Differential One Way Ranging (Delta-DOR) Operations*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 506.0-M-2. Washington, D.C.: CCSDS, February 2018.

[15] "NORAD Two-Line Element Set Format." CelesTrak. https://celestrak.org/NORAD/documentation/tle-fmt.php.

2 OVERVIEW

2.1 GENERAL

This section provides a high-level overview of the common data entities defined within the scope of service management.

2.2 DATA ENTITIES

Within the scope of service management, data entities are defined as XML constructs. Thus the data entities are defined in XML Schema files.

Data is either mandatory, in which case, suitable values must be present, or optional, in which case, values may be present or not. In addition, it is possible to extend the contents of the data entities by defining additional parameters. The content of any additional parameters so defined is outside the scope of this document and should be documented in an ICD agreed upon by the involved parties.

2.3 MAPPING TO W3C XML SCHEMA

This Recommended Practice includes the specification of a mapping to World Wide Web Consortium (W3C) eXtensible Markup Language (XML) schema. The normative mapping of this Recommended Practice to W3C XML schemas is a virtual annex to this Recommended Practice and is contained in a stand-alone set of schema files.

NOTE – The XML schema has been elaborated on the basis of the mapping guidelines described in reference [C1].

3 SERVICE MANAGEMENT COMMON DATA ENTITIES

3.1 SERVICE MANAGEMENT INFORMATION ENTITY CONTENT/STRUCTURE

3.1.1 OVERVIEW

The Service Management Information Entity forms the basis for defining information entities required by service management. Figure 3-1 shows the UML class diagram for the Service Management Information Entity. For clarity, abstract classes are highlighted in green.

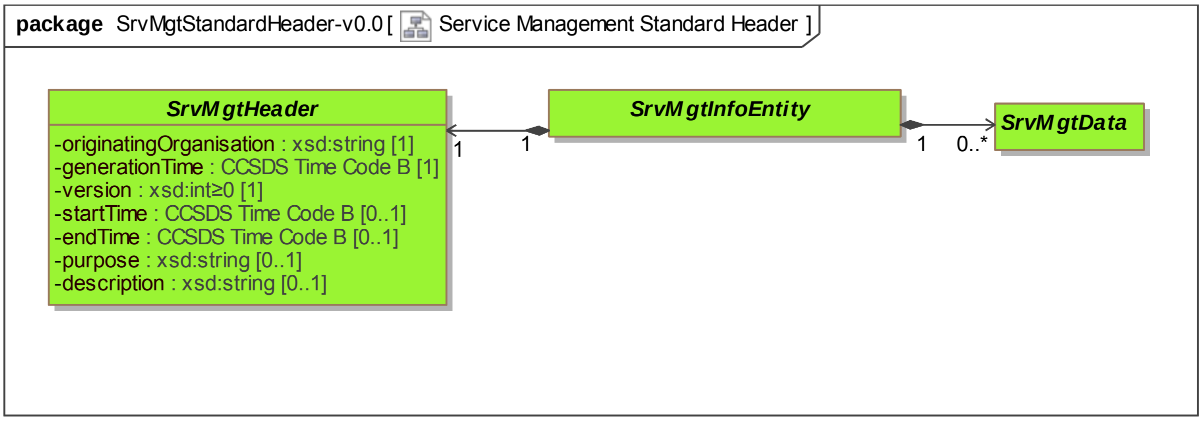
package SrvMgtStandardHeader-v0.0 [Service Management Standard Header]
SrvMgtHeader
-originatingOrganisation : xsd:string [1]
-generationTime : CCSDS Time Code B [1]
-version : xsd:int≥0 [1]
-startTime : CCSDS Time Code B [0..1]
-endTime : CCSDS Time Code B [0..1]
-purpose : xsd:string [0..1]
-description : xsd:string [0..1]
1
1
SrvMgtInfoEntity
1
0..*
SrvMgtData

Figure 3-1: Service Management Information Entity Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.1.2 CLASS SrvMgtInfoEntity (ABSTRACT)

3.1.2.1 The SrvMgtInfoEntity is an abstract class that may be used to instantiate the various classes of information entity required by service management.

3.1.2.2 The SrvMgtInfoEntity class shall contain no parameters.

3.1.3 CLASS SrvMgtHeader (ABSTRACT)

3.1.3.1 SrvMgtHeader is an abstract class that may be used to instantiate the header required by service management.

3.1.3.2 The SrvMgtHeader class parameters specified in table 3-1 shall be used to identify the originating organization, status of the schedule, status, version, time at which the information entity was generated, start time to which the information entity applies, and end time to which it applies.

NOTE – Optional parameters allow for further definition of the purpose of the message and a description of its contents.

Table 3-1: Class SrvMgtHeader Parameters

Parameter	Description	Data Type	Data Units
`originatingOrganization`	The organization that generated the information entity.	xsd:string Permitted values are registered in SANA (see reference [3] for further information).	n/a
`generationTime`	Time at which the information entity was generated.	xsd:string restricted to holding the time in CCSDS ASCII Time Code B format (reference [1])	UTC
`version`	The version of the information entity. This increments every time an information entity of the same concrete type, status, and time range is generated (i.e., has the same `startTime` and `endTime`). NOTE – The version may increment by 1 every time but is not constrained to do so. The only constraint is that each version number is greater than the previous.	xsd:positiveInteger	n/a
`startTime`	Optional parameter: The start time to which the information entity applies. NOTE – For certain Service Management Information Entity types, use of this parameter is mandatory.	xsd:string restricted to holding the time in CCSDS ASCII Time Code B format (reference [1])	UTC
`endTime`	Optional parameter: The end time to which the information entity applies. NOTE – For certain Service Management Information Entity types, use of this parameter is mandatory.	xsd:string restricted to holding the time in CCSDS ASCII Time Code B format (reference [1])	UTC
`purpose`	Optional parameter: May be used to specify the purpose of the information entity.	xsd:string Exact use of this may be specified in other Service Management Recommended Standards and Practices in which the SrvMgtHeader class is used or in an ICD between relevant parties.	n/a
`description`	Optional parameter. May be used to describe the contents of the information entity, for example, any filter criteria used in the generation of the information entity.	xsd:string Exact use of this may be specified in an ICD between relevant parties.	n/a

3.1.4 CLASS SrvMgtData (ABSTRACT)

3.1.4.1 SrvMgtData is an abstract class that may be used to instantiate the various classes of data required by service management.

3.1.4.2 The SrvMgtData class shall contain no parameters.

3.2 SrvMgtAbstractEvent AND SrvMgtEventTime CONTENT/STRUCTURE

3.2.1 OVERVIEW

The SrvMgtAbstractEvent and SrvMgtEventTime define the event and associated time classes required for the definition of events in Service Management. The SrvMgtAbstractEvent is derived from the AbstractEvent class and the SrvMgtEventTime is derived from the AbstractEvent class (see reference [6]).

Figure 3-2 shows the UML class diagram for the SrvMgtAbstractEvent and SrvMgtEventTime classes. It should be noted that, for clarity, abstract classes are highlighted in green. These classes are used to instantiate events needed in the context of Service Management. They are specializations of the AbstractEvent and AbstractEventTime classes described in reference [6].

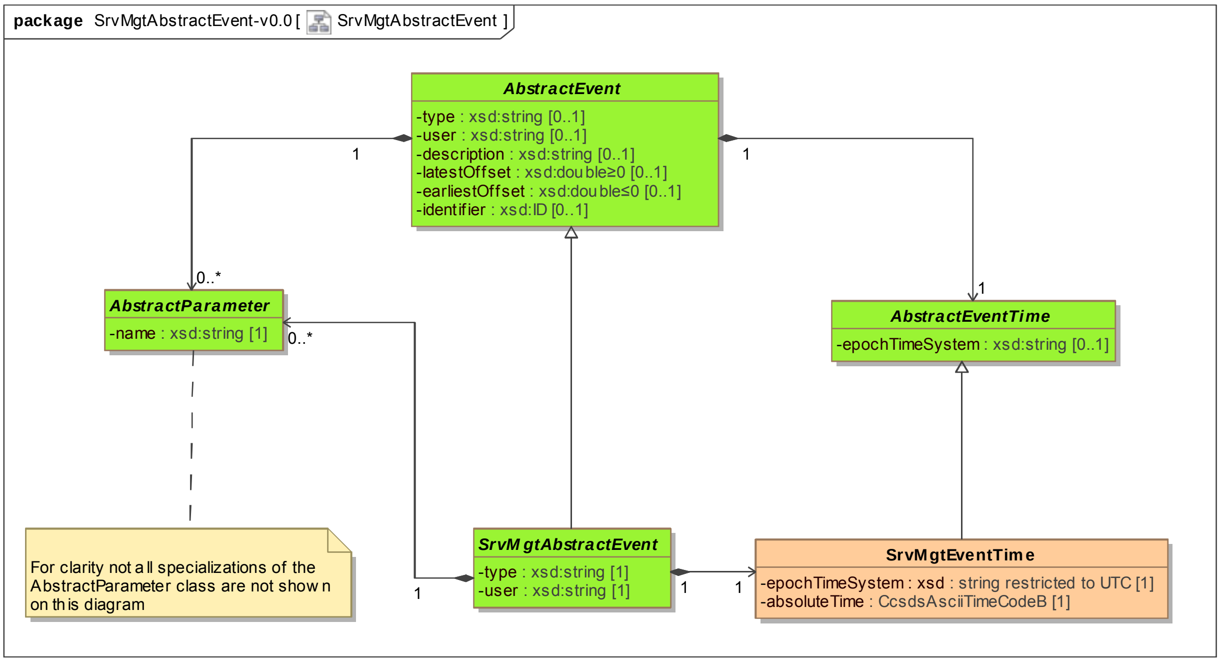
package SrvMgtAbstractEvent-v0.0 [SrvMgtAbstractEvent]
AbstractEvent
-type : xsd:string [0..1]
-user : xsd:string [0..1]
-description : xsd:string [0..1]
-latestOffset : xsd:double≥0 [0..1]
-earliestOffset : xsd:double≤0 [0..1]
-identifier : xsd:ID [0..1]
1
1
0..*
AbstractParameter
-name : xsd:string [1]
0..*
1
AbstractEventTime
-epochTimeSystem : xsd:string [0..1]
For clarity not all specializations of the AbstractParameter class are shown on this diagram
1
SrvMgtAbstractEvent
-type : xsd:string [1]
-user : xsd:string [1]
1
1
SrvMgtEventTime
-epochTimeSystem : xsd : string restricted to UTC [1]
-absoluteTime : CcsdsAsciiTimeCodeB [1]

Figure 3-2: SrvMgtAbstractEvent Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.2.2 CLASS SrvMgtAbstractEvent (ABSTRACT)

3.2.2.1 The SrvMgtAbstractEvent class is an abstract class that may be used to instantiate the various events that are required in service management.

NOTE – The SrvMgtAbstractEvent class is a specialization of class AbstractEvent described in reference [6]), and the generic description of the parameters is given in that document.

3.2.2.2 In the context of the SrvMgtAbstractEvent class, the usage of the AbstractEvent parameters defined described in reference [6] shall be as specified in table 3-2.

Table 3-2: Class SrvMgtAbstractEvent Use of Class AbstractEvent Parameters

Parameter	Description	Data Type	Data Units
`type`	In the context of the SrvMgtAbstractEvent, this parameter is mandatory and is used to specify the type of the event. The various types of event permitted are discussed in reference [5], subsection 5.3. NOTE – The values listed here are registered with SANA. (See reference [6], annex subsection A2.5 for more information re: unregistered event types.)	xsd:string – registered values of – COMMS Predicted communications geometry events (see reference [5], subsection 5.6); – DATARATE Sustainable data rates & volume estimate events (see reference [5], subsection 5.7); – RFI Predicted radio frequency interference events (see reference [5], subsection 5.8); – CONFLICTS Predicted resource conflict events (see reference [5], subsection 5.9).	n/a
`user`	In the context of the SrvMgtAbstractEvent, this parameter is mandatory and is used to specify the user to which the event is relevant. These will typically be spacecraft names as specified in SANA. NOTE – These will typically be spacecraft abbreviations as indicated in the SANA Considerations in annex A. The using recommendation may indicate users other than spacecraft.	Xsd:string Permitted values are registered in SANA.	n/a

3.2.3 CLASS SrvMgtEventTime

3.2.3.1 The SrvMgtEventTime class may be used to specify the time at which an event occurs in the context of service management.

NOTE – The SrvMgtEventTime class is a specialization of class AbstractEventTime described in reference [6], and the generic description of the parameters is given in that document.

3.2.3.2 The SrvMgtEventTime class, in addition to the parameters of class AbstractEventTime described in reference [6], shall contain the parameters as specified in table 3-3.

Table 3-3: Class SrvMgtEventTime Additional Parameters

Parameter	Description	Data Type	Data Units
`absoluteTime`	The absolute time at which the event occurs.	xsd: string restricted to holding the time in CCSDS ASCII Time Code B format (reference [1])	UTC

3.2.3.3 In the context of the SrvMgtEventTime class, the usage of the AbstractEventTime class parameter defined in reference [6] shall be as specified in table 3-4.

Table 3-4: Class SrvMgtEventTime Use of Class AbstractEventTime Parameters

Parameter	Description	Data Type	Data Units
`epochTimeSystem`	In the context of Service Management this parameter is mandatory and is used to specify the Epoch Time System to which the time is relative.	xsd:string In the context of the SrvMgtEventTime, the permitted values are restricted to – UTC.	n/a

3.3 SERVICE MANAGEMENT COMMON CLASSES—Aperture Location

3.3.1 OVERVIEW

The ApertureLocation class is one of the data entities that compose Service Management Common Classes that are used in various parts of data management.

Figure 3-3 shows the UML class diagram for the Service Management ApertureLocation class.

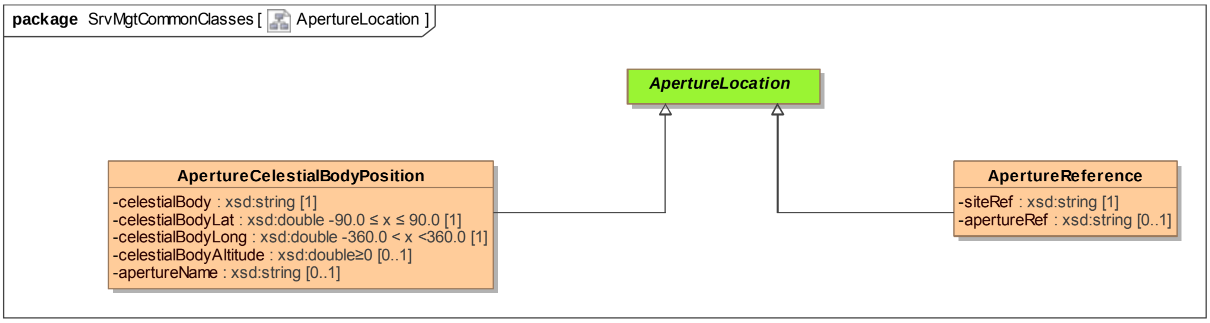
package SrvMgtCommonClasses [ApertureLocation]
ApertureLocation
ApertureCelestialBodyPosition
-celestialBody : xsd:string [1]
-celestialBodyLat : xsd:double -90.0 ≤ x ≤ 90.0 [1]
-celestialBodyLong : xsd:double -360.0 < x <360.0 [1]
-celestialBodyAltitude : xsd:double≥0 [0..1]
-apertureName : xsd:string [0..1]
ApertureReference
-siteRef : xsd:string [1]
-apertureRef : xsd:string [0..1]

Figure 3-3: Service Management ApertureLocation Class Diagram

3.3.2 CLASS ApertureLocation (ABSTRACT)

3.3.2.1 ApertureLocation is an abstract class that may be used to instantiate the various classes of aperture location required by service management.

3.3.2.2 The ApertureLocation class shall contain no parameters.

3.3.3 CLASS ApertureReference

3.3.3.1 The ApertureReference class may be used to specify the location of an aperture by the site name where it is located and the name of the aperture.

NOTE – The ApertureReference class is a specialization of class ApertureLocation described in 3.3.2.

3.3.3.2 The parameters of the ApertureReference class shall be as specified in table 3-5.

Table 3-5: Class ApertureReference Parameters

Parameter	Description	Data Type	Data Units
`siteRef`	Name of the site (ground or spacecraft) to which the event refers.	xsd:string Permitted values are registered in SANA (see A2.4 for further information).	n/a
`apertureRef`	Optional Parameter. This may be used to specify the identifier of the aperture (ground or spacecraft) to which the event refers.	xsd:string Permitted values are registered in SANA (see A2.4 for further information).	n/a

3.3.4 CLASS ApertureCelestialBodyPosition

3.3.4.1 The ApertureCelestialBodyPosition class allows the specification of an arbitrary location on a celestial body by means of latitude and longitude. The conventions adopted for the specification of the latitude and longitude on a particular celestial body should be as specified in reference [9]. If this is not the case, then the particular convention used should be documented in an ICD between the relevant parties.

NOTE – The ApertureCelestialBodyPosition class is a specialization of class ApertureLocation described in 3.3.2.

3.3.4.2 The parameters of the ApertureCelestialBodyPosition class shall be as specified in table 3-6.

Table 3-6: Class ApertureCelestialBodyPosition Parameters

Parameter	Description	Data Type	Data Units
`celestialBody`	Celestial body on which the location is being specified.	xsd:string Permitted values are as follows: – Planet and Satellite Names – as specified in reference [7]; – Minor Planet Names, as specified in reference [8].	n/a
`celestialBodyLat`	The latitude on the celestial body.	xsd:double -90.0 ≤ x ≤ 90.0 The conventions adopted for the specification of the latitude on a particular celestial body shall be as specified in reference [9].	Degrees
`celestialBodylong`	The longitude on the celestial body.	xsd:double -360.0 < x < 360.0 The conventions adopted for the specification of the longitude on a particular celestial body shall be as specified in reference [9].	Degrees
`celestialBodyAltitude`	Optional Parameter. The altitude of the aperture above the centre of the celestial body.	xsd:double x ≥ 0	Meters
`apertureName`	Optional Parameter. This is used to specify the name of the aperture.	xsd:string	n/a

3.3.4.3 Some provider Cross Support Service Systems (CSSSes) make the approximation that an aperture is located at the centre of a celestial body when calculating occultations. The ApertureCelesitalBodyPosition class may also be used to specify the position of an aperture as though it were located at the centre of a celestial body.[1] To do this, the following values must be specified for the following parameters of the ApertureCelesitalBodyPosition class:

– celestialBodyLat	0.0
– celestialBodylong	0.0
– celestialBodyAltitude	0.0
– apertureName	“” (i.e., an empty string)

[1] This is obviously only an approximation; however, it is one that some Provider CSSSes make when calculating occultations from distant celestial bodies. For example, when calculating the times at which the occultation of a spacecraft orbiting Mars starts and ends as seen from Earth, if a Provider CSSS used the centre of the Earth as the reference for its calculation, it would obtain the same occlusion start and end times for all Earth-based apertures. In this case, the ApertureCelestialBodyPosition class would have the following parameters:

– celestialBody	“Earth”
– celestialBodyLat	0.0
– celestialBodylong	0.0
– celestialBodyAltitude	0.0
– apertureName	“”

3.4 SERVICE MANAGEMENT COMMON CLASSES—PlanningInfoTypeList LIST

3.4.1 OVERVIEW

The PlanningInfoTypeList class is one of the data entities that compose Service Management Common classes that are used in various parts of data management.

Figure 3-4 shows the UML class diagram for the Service Management PlanningInfoTypeList class.

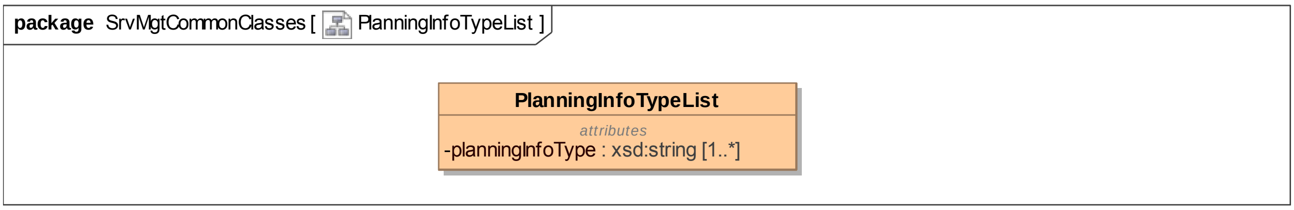
package SrvMgtCommonClasses [PlanningInfoTypeList]
PlanningInfoTypeList
attributes
-planningInfoType : xsd:string [1..*]

Figure 3-4: Service Management PlanningInfoTypeList Class Diagram

3.4.2 CLASS PlanningInfoTypeList

3.4.2.1 The PlanningInfoTypeList class may be used to specify a list of Planning Info Types.

3.4.2.2 The parameters of the PlanningInfoTypeList class shall be as specified in table 3-7.

Table 3-7: Class PlanningInfoTypeList Parameters

Parameter	Description	Data Type	Data Units
`planningInfoType`	This specifies the type of data that is contained in the planning information. The various types of data that can be contained in a planning information are discussed in reference [5], subsection 5.3. Planning information may contain 1 or more of these different planning types.	Array of Xsd:strings Currently supported values are: – COMMS Predicted communications geometry events NOTE – It is expected that in future releases of this Recommended Practice, this list will be extended to include some or all of the following: – DATARATE Sustainable data rates & volume estimate – RFI Predicted radio frequency interference events – CONFLICTS Predicted resource conflict events – COSTS Cost estimates	n/a

3.5 SERVICE MANAGEMENT COMMON CLASSES—ModResParm

3.5.1 OVERVIEW

The ModResParm (short for Modified Resource Parameter) class is one of the data entities that make up the Service Management Common classes that are used in various parts of data management.

Figure 3-5 shows the UML class diagram for the Service Management ModResParm class.

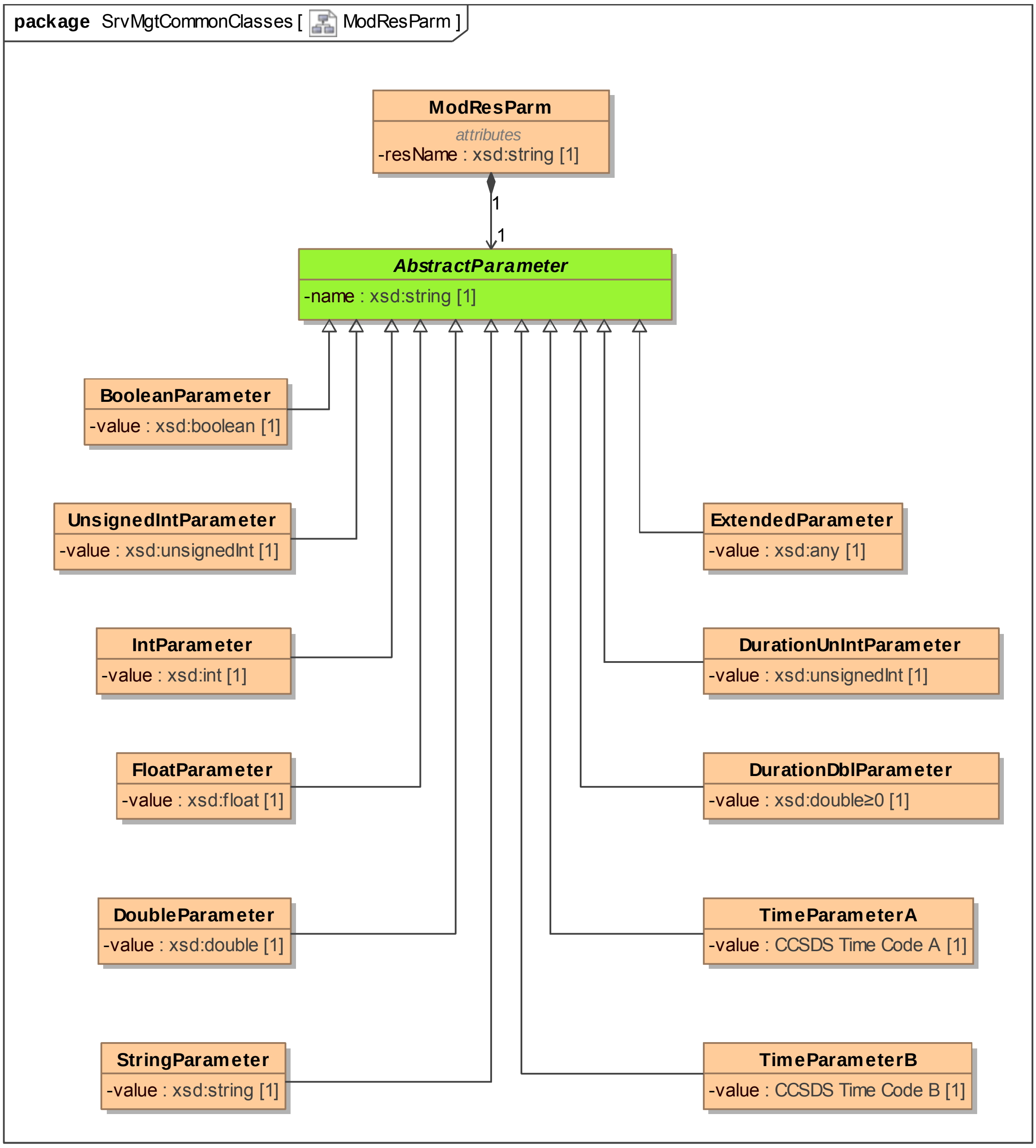

Figure 3-5: Service Management ModResParm Class Diagram

3.5.2 CLASS ModResParm

3.5.2.1 The ModResParm class may be used to assign a value to a resource parameter. The resource to which the value is to be assigned shall be specified by the resName parameter, and the parameter of the Resource that is to be modified shall be specified by the name parameter in classes derived from the AbstractParameter class.

3.5.2.2 Each instance of the ModResParm class must contain one and only one instance of a parameter class derived from the abstract AbstractParameter class described in reference [6].

3.5.2.3 The parameters of the ModResParm class shall be as specified in table 3-8.

Table 3-8: Class ModResParm Parameters

Parameter	Description	Data Type	Data Units
`resName`	The name of the resource for which the modified parameter value is to be specified.	xsd:string	n/a

3.5.2.4 In the event that the ModResParm class may be used to specify modified values of Functional Resources (see reference [10]), the following should be noted:

– The `resName` parameter of the ModResParm must contain the `fRNickname` of the function resource that is to be modified.

– The `name` parameter of AbstractParameter class shall be used to specify the name of the parameter of the functional resource for which the parameter value is to be modified.

3.6 SERVICE MANAGEMENT DATA WRAPPER CLASSES

3.6.1 OVERVIEW

The Service Management Data Wrapper classes are a set of classes that are used within Service Management to provide wrappers for various types of data entity. All the data wrapper classes are instantiated from the abstract AbstractDataWrapper class.

Figure 3-6 shows the UML class diagram for the Service Management Data Wrapper classes.

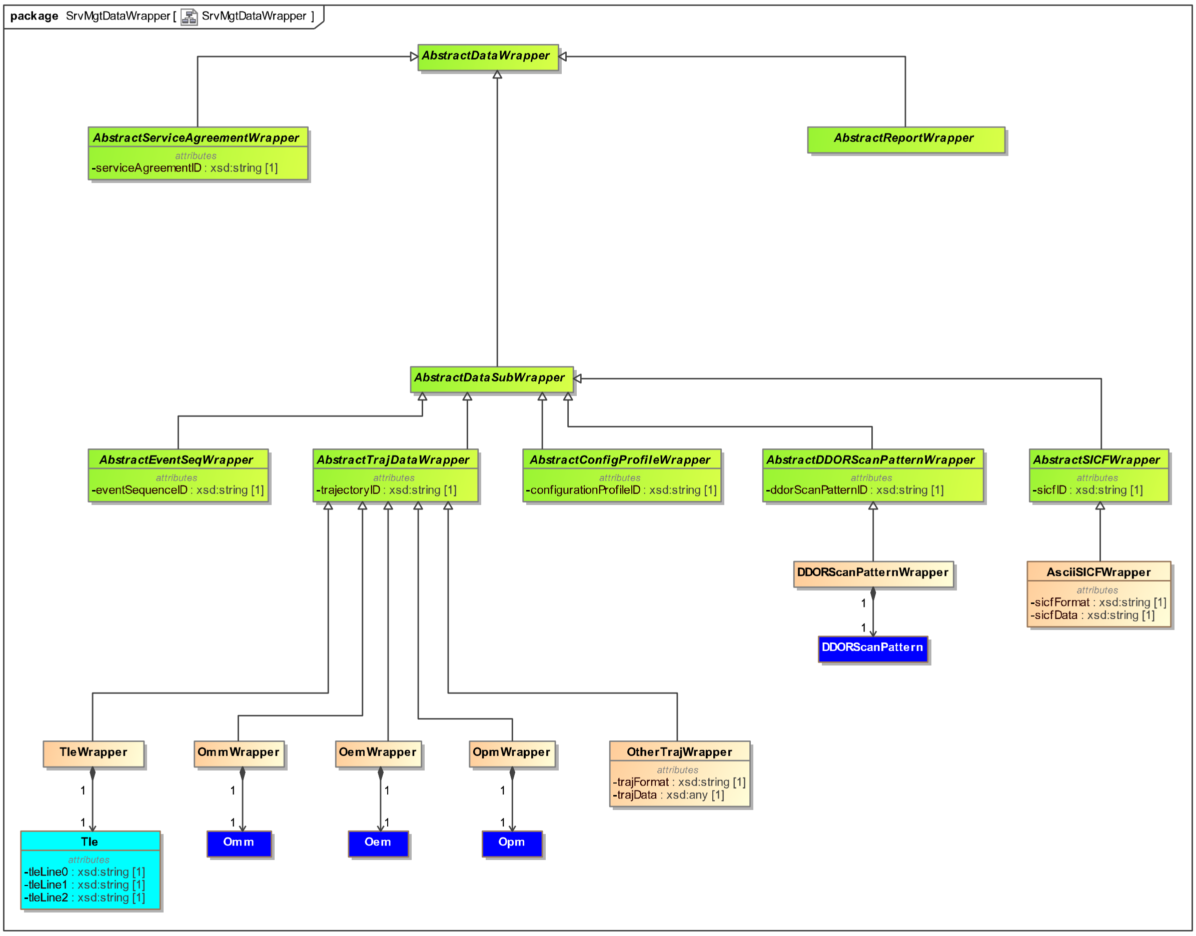
package SrvMgtDataWrapper [SrvMgtDataWrapper]
AbstractDataWrapper
AbstractServiceAgreementWrapper
attributes
-serviceAgreementID : xsd:string [1]
AbstractReportWrapper
AbstractDataSubWrapper
AbstractEventSeqWrapper
attributes
-eventSequenceID : xsd:string [1]
AbstractTrajDataWrapper
attributes
-trajectoryID : xsd:string [1]
AbstractConfigProfileWrapper
attributes
-configurationProfileID : xsd:string [1]
AbstractDDORScanPatternWrapper
attributes
-ddorScanPatternID : xsd:string [1]
AbstractSICFWrapper
attributes
-sicfID : xsd:string [1]
DDORScanPatternWrapper
1
1
DDORScanPattern
AsciiSICFWrapper
attributes
-sicfFormat : xsd:string [1]
-sicfData : xsd:string [1]
TleWrapper
1
1
Tle
attributes
-tleLine0 : xsd:string [1]
-tleLine1 : xsd:string [1]
-tleLine2 : xsd:string [1]
OmmWrapper
1
1
Omm
OemWrapper
1
1
Oem
OpmWrapper
1
1
Opm
OtherTrajWrapper
attributes
-trajFormat : xsd:string [1]
-trajData : xsd:any [1]

Figure 3-6: Service Management Data Wrapper Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.6.2 CLASS AbstractDataWrapper (ABSTRACT)

3.6.2.1 The AbstractDataWrapper is an abstract class that may be used to instantiate the various classes of data wrappers required by service management.

3.6.2.2 The AbstractDataWrapper class shall contain no parameters.

3.6.3 CLASS AbstractServiceAgreementWrapper (ABSTRACT)

3.6.3.1 AbstractServiceAgreementWrapper is an abstract class that may be used to instantiate the wrapper classes required for Service Agreement data entities.

3.6.3.2 If required, an Agency- (or mission-) specific Service Agreement Wrapper class may be derived from this class.

NOTE – The AbstractServiceAgreementWrapper class is a specialization of class AbstractDataWrapper described in 3.6.2, and the generic description of the parameters is given in that subsection.

3.6.3.3 The AbstractServiceAgreementWrapper class parameter specified in table 3-9 shall be used to identify the service agreement encapsulated in the wrapper class.

Table 3-9: Class AbstractServiceAgreementWrapper Parameters

Parameter	Description	Data Type	Data Units
`serviceAgreementID`	A unique Service Agreement ID assigned by the service provider.	xsd:string	n/a

3.6.4 CLASS AbstractReportWrapper (ABSTRACT)

3.6.4.1 AbstractReportWrapper is an abstract class that may be used to instantiate the wrapper classes required to encapsulate report data entities.

3.6.4.2 If required, an Agency- (or mission-) specific Report Wrapper class may be derived from this class.

NOTE – The AbstractReportWrapper class is a specialization of class AbstractDataWrapper described in 3.6.2, and the generic description of the parameters is given in that subsection.

3.6.4.3 The AbstractReportWrapper class shall contain no additional parameters.

3.6.5 CLASS AbstractDataSubWrapper (ABSTRACT)

3.6.5.1 AbstractDataSubWrapper is an abstract class that may be used to instantiate the wrapper classes required to encapsulate data entities that may be used in submission requests (see reference [12]).

NOTE – The AbstractDataSubWrapper class is a specialization of class AbstractDataWrapper described in 3.6.2, and the generic description of the parameters is given in that subsection.

3.6.5.2 The AbstractDataSubWrapper class shall contain no additional parameters.

3.6.6 CLASS AbstractEventSeqWrapper (ABSTRACT)

3.6.6.1 AbstractEventSeqWrapper is an abstract class that may be used to instantiate the wrapper classes required for Event Sequence data entities.

3.6.6.2 If required, an Agency- (or mission-) specific Event Sequence Wrapper class may be derived from this class.

NOTE – The AbstractEventSeqWrapper class is a specialization of class AbstractDataSubWrapper described in 3.6.5, and the generic description of the parameters is given in that subsection.

3.6.6.3 The AbstractEventSeqWrapper class parameter specified in table 3-10 shall be used to identify the event sequence encapsulated in the wrapper class.

Table 3-10: Class AbstractEventSeqWrapper Parameters

Parameter	Description	Data Type	Data Units
`eventSequenceID`	The ID of the event sequence.[2] In a submission request (see reference [12]) the ID of the event sequence is assigned. This can later be used to reference event sequence. If this identifier is a value that has not been used previously, then it is assumed that it refers to a new event sequence. If the identifier is the same as one used previously by the same user the existing event sequence is replaced by that specified in the submission request.	xsd:string	n/a

3.6.7 CLASS AbstractTrajDataWrapper (ABSTRACT)

3.6.7.1 AbstractTrajDataWrapper is an abstract class that may be used to instantiate the wrapper classes required for Trajectory data entities.

3.6.7.2 If required, an Agency- (or mission-) specific Trajectory Data Wrapper class may be derived from this class.

NOTE – The AbstractTrajDataWrapper class is a specialization of class AbstractDataSubWrapper described in 3.6.5, and the generic description of the parameters is given in that subsection.

3.6.7.3 The AbstractTrajDataWrapper class parameter specified in table 3-11 shall be used to identify the trajectory data encapsulated in the wrapper class.

[2] It is the responsibility of the User CSSS to assign the eventSequenceID. The provider CSSS can then assume that the combination of serviceAgreementRef (see reference [12]), spaceUserNode (see reference [12]), and eventSequenceID constitutes a unique identifier for an event sequence.

Table 3-11: Class AbstractTrajDataWrapper Parameters

Parameter	Description	Data Type	Data Units
`trajectoryID`	The ID of the trajectory.[3] In a submission request (see reference [12]) the ID of the trajectory is assigned. This can later be used to reference the trajectory. If this identifier is a value that has not been used previously then it is assumed that it refers to a trajectory. If the identifier is the same as one used previously by the same user, the existing trajectory is replaced by that specified in the submission request.	xsd:string	n/a

3.6.8 CLASS OemWrapper

3.6.8.1 The OemWrapper is a class that may be used to wrap an OEM data entity that conforms to the specification in reference [13].

NOTE – The OemWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.6.7, and the generic description of the parameters is given in that subsection.

3.6.8.2 The OemWrapper class shall contain no additional parameters.

3.6.8.3 Each instance of a OemWrapper class shall contain 1, and only 1, instance of the OEM class specified in reference [13].

3.6.9 CLASS OmmWrapper

3.6.9.1 The OmmWrapper is a class that may be used to wrap an OMM data entity that conforms to the specification in reference [13].

NOTE – The OmmWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.6.7, and the generic description of the parameters is given in that subsection.

3.6.9.2 The OmmWrapper class shall contain no additional parameters.

[3] It is the responsibility of the User CSSS to assign the trajectoryID. The provider CSSS can then assume that the combination of serviceAgreementRef (see reference [12]), spaceUserNode (see reference [12]), and trajectoryID constitutes a unique identifier for an event sequence.

3.6.9.3 Each instance of a OmmWrapper class shall contain 1, and only 1, instance of the OMM class specified in reference [13].

3.6.10 CLASS OpmWrapper

3.6.10.1 The OpmWrapper is a class that may be used to wrap an OPM data entity that conforms to the specification in reference [13].

NOTE – The OpmWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.6.7, and the generic description of the parameters is given in that subsection.

3.6.10.2 The OpmWrapper class shall contain no additional parameters.

3.6.10.3 Each instance of a OpmWrapper class shall contain 1, and only 1, instance of the OPM class specified in reference [13].

3.6.11 CLASS TleWrapper

3.6.11.1 The TleWrapper is a class that may be used to wrap a TLE data entity that conforms to the specification in reference [15].

NOTE – The TleWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.6.7, and the generic description of the parameters is given in that subsection.

3.6.11.2 The TleWrapper class shall contain no additional parameters.

3.6.11.3 Each instance of a OpmWrapper class shall contain 1, and only 1, instance of the TLE class specified in 3.6.12.

3.6.12 CLASS Tle

3.6.12.1 The Tle class may be used to specify trajectory data in the TLE format (see reference [15]).

NOTE – The TLE format is not a CCSDS recommendation. It is therefore strongly recommended that, if at all feasible, one of the OEM, OMM, or OPM formats be used instead.

3.6.12.2 The Tle class parameters specified in table 3-12 shall be used to specify the trajectory data in TLE format as per reference [15].

Table 3-12: Class Tle Parameters

Parameter	Description	Data Type	Data Units
`tleLine0`	Line 0 of the TLE (see 3.6.12.2.1 for a detailed description of the format).	xsd:string*24	n/a
`tleLine1`	Line 0 of the TLE (see 3.6.12.2.2 for a detailed description of the format).	xsd:string*69	n/a
`tleLine2`	Line 0 of the TLE (see 3.6.12.2.3 for a detailed description of the format).	xsd:string*69	n/a

3.6.12.2.1 tleLine0 Format

From reference [15], the format of Line 0 of a TLE is a 24-character name (to be consistent with the name length in the NORAD SATCAT).

3.6.12.2.2 tleLine1 Format

From reference [15], the format of Line 1 of a TLE is a 69-character long string composed of the elements given in table 3-13.

Table 3-13: Class TLE Line 1 Format

Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U = Unclassified)
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (Leading decimal point assumed)
54-61	BSTAR drag term (Leading decimal point assumed)
63	Ephemeris type
65-68	Element number
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Any column not specified in table 3-13 must contain a ‘space’ character.

3.6.12.2.3 tleLine2 Format

From reference [15], the format of Line 2 of a TLE is a 69-character long string composed of the elements given in table 3-14.

Table 3-14: Class TLE Line 2 Format

Column	Description
01	Line Number of Element Data
03-07	Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (Leading decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63	Mean Motion [Revs per day]
64-68	Revolution number at epoch [Revs]
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Any column not specified in table 3-14 must contain a ‘space’ character.

3.6.13 CLASS OtherTrajWrapper

3.6.13.1 The OtherTrajWrapper is a class that may be used to wrap an arbitrary format trajectory data entity. This shall be accomplished by using the type xsd:any to hold the required trajectory data; because of this, an additional schema must be defined to specify the data that is stored in the xsd:any element.

NOTE – The OtherTrajWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.6.7, and the generic description of the parameters is given in that subsection.

3.6.13.2 The OtherTrajWrapper class parameters specified in table 3-15 shall be used to specify the trajectory data.

Table 3-15: Class OtherTrajWrapper Parameters

Parameter	Description	Data Type	Data Units
`trajFormat`	Name of the trajectory format.	xsd:string	n/a
`trajData`	The trajectory data. NOTE – As the type of this element is xsd:any, a schema will need to be defined to specify the data that is stored in this element.	xsd:any	n/a

3.6.14 CLASS AbstractConfigProfileWrapper (ABSTRACT)

3.6.14.1 AbstractConfigProfileWrapper is an abstract class that may be used to instantiate the wrapper classes required for Configuration Profile data entities.

3.6.14.2 If required, an Agency- (or mission-) specific Configuration Profile Wrapper class may be derived from this class.

NOTE – The AbstractConfigProfileWrapper class is a specialization of class AbstractDataSubWrapper described in 3.6.5 and the generic description of the parameters is given in that subsection.

3.6.14.3 The AbstractConfigProfileWrapper class parameter specified in table 3-16 shall be used to identify the configuration profile encapsulated in the wrapper class.

Table 3-16: Class AbstractConfigProfileWrapper Parameters

Parameter	Description	Data Type	Data Units
`configurationProfileID`	The ID of the configuration profile.[4] In a submission request (see reference [12]), the ID of the configuration profile is assigned. This can later be used to reference the configuration profile. If this identifier is a value that has not been used previously, then it is assumed that it refers to a new configuration profile. If the identifier is the same as one used previously by the same user, the existing configuration profile is replaced by that specified in the submission request.	xsd:string	n/a

3.6.15 CLASS AbstractDDORScanPatternWrapper (ABSTRACT)

3.6.15.1 AbstractDDORScanPatternWrapper is an abstract class that may be used to instantiate the wrapper classes required for DDOR Scan Pattern data entities.

3.6.15.2 If required, an Agency- (or mission-) specific DDOR Scan Pattern Wrapper class may be derived from this class.

NOTE – The AbstractDDORScanPatternWrapper class is a specialization of class AbstractDataSubWrapper described in 3.6.5, and the generic description of the parameters is given in that subsection.

3.6.15.3 The AbstractDDORScanPatternWrapper class parameter specified in table 3-17 shall be used to identify the DDOR Scan Pattern encapsulated in the wrapper class.

[4] It is the responsibility of the User CSSS to assign the configurationProfileID. The provider CSSS can then assume that the combination of serviceAgreementRef (see reference [12]), spaceUserNode (see reference [12]), and configurationProfileID constitutes a unique identifier for a configuration profile.

Table 3-17: Class AbstractDDORScanPatternWrapper Parameters

Parameter	Description	Data Type	Data Units
`ddorScanPatternID`	The ID of the DDOR Scan Pattern.[5] In a submission request (see reference [12]) the ID of the DDOR Scan Pattern is assigned. This can later be used to reference the DDOR Scan Pattern. If this identifier is a value that has not been used previously, then it is assumed that it refers to a new DDOR Scan Pattern. If the identifier is the same as one used previously by the same user, the existing DDOR Scan Pattern is replaced by that specified in the submission request.	xsd:string	n/a

3.6.16 CLASS DDORScanPatternWrapper

3.6.16.1 The DDORScanPatternWrapper is a class that may be used to wrap a Configuration Profile data entity that conforms to the specification in reference [14].

NOTE – The DDORScanPatternWrapper class is a specialization of class AbstractDDORScanPatternWrapper described in 3.6.15, and the generic description of the parameters is given in that subsection.

3.6.16.2 The DDORScanPatternWrapper class shall contain no additional parameters.

3.6.16.3 Each instance of a DDORScanPatternWrapper class shall contain 1, and only 1, instance of the DDORScanPattern class specified in reference [14].

3.6.17 CLASS AbstractSICFWrapper (ABSTRACT)

3.6.17.1 AbstractSICFWrapper is an abstract class that may be used to instantiate the wrapper classes required for Service Instance Configuration File (SICF) data entities.

3.6.17.2 As there is no standard specification for SICF, an Agency- (or mission-) specific SICF Wrapper class must be derived from this class.

NOTE – The AbstractSICFWrapper class is a specialization of class AbstractDataSubWrapper described in 3.6.5, and the generic description of the parameters is given in that subsection.

3.6.17.3 The AbstractSICFWrapper class parameter specified in table 3-18 shall be used to identify the SICF encapsulated in the wrapper class.

[5] It is the responsibility of the User CSSS to assign the ddorScanPatternID. The provider CSSS can then assume that the combination of serviceAgreementRef (see reference [12]), spaceUserNode (see reference [12]), and ddorScanPatternID constitutes a unique identifier for a DDOR Scan pattern.

Table 3-18: Class AbstractConfigProfileWrapper Parameters

Parameter	Description	Data Type	Data Units
`sicfID`	The ID of the SICF.[6] In a submission request (see reference [12]), the ID of the SICF is assigned. This can later be used to reference the SICF. If this identifier is a value that has not been used previously, then it is assumed that it refers to a new SICF. If the identifier is the same as one used previously by the same user the existing SICF is replaced by that specified in the submission request.	xsd:string	n/a

3.6.18 CLASS AsciiSICFWrapper

3.6.18.1 Whilst, as noted above, there is no standard format for SICFs, the vast majority of these are at least expressed in ASCII. The AsciiSICFWrapper is a class that may be used to wrap an arbitrary-format SICF that shall be in ASCII format.

NOTE – The AsciiSICFWrapper class is a specialization of class AbstractSICFWrapper described in 3.6.17, and the generic description of the parameters is given in that subsection.

3.6.18.2 The AsciiSICFWrapper class parameters specified in table 3-19 shall be used to specify the ASCII format SICF data.

Table 3-19: Class ASCIISICFWRAPPER Parameters

Parameter	Description	Data Type	Data Units
`sicfFormat`	Name of the SICF format.	xsd:string	n/a
`sicfData`	The ASCII format SICF data.	xsd:string	n/a

[6] It is the responsibility of the User CSSS to assign the sicfID. The provider CSSS can then assume that the combination of serviceAgreementRef (see reference [12]), spaceUserNode (see reference [12]), and sicfID constitutes a unique identifier for a SICF.

3.7 SERVICE MANAGEMENT SERVICE PACKAGE CLASSES

3.7.1 OVERVIEW

The Service Management Service Package Classes are a set of classes that are used within Service Management in both the requesting a service package and returning service packages created as a result of service package requests.

Figure 3-7 shows the UML Class diagram for the Service Management Data Wrapper classes.

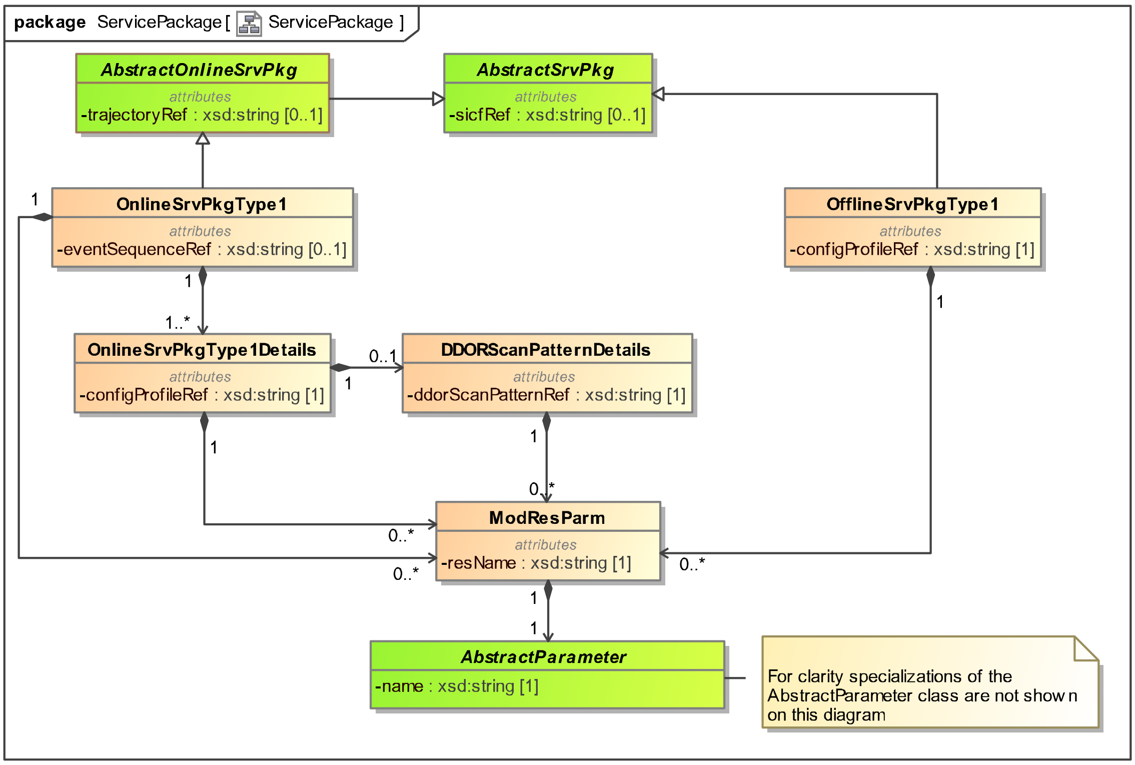
package ServicePackage [ServicePackage]
AbstractOnlineSrvPkg
attributes
-trajectoryRef : xsd:string [0..1]
AbstractSrvPkg
attributes
-sicfRef : xsd:string [0..1]
OnlineSrvPkgType1
attributes
-eventSequenceRef : xsd:string [0..1]
OfflineSrvPkgType1
attributes
-configProfileRef : xsd:string [1]
OnlineSrvPkgType1Details
attributes
-configProfileRef : xsd:string [1]
DDORScanPatternDetails
attributes
-ddorScanPatternRef : xsd:string [1]
ModResParm
attributes
-resName : xsd:string [1]
AbstractParameter
-name : xsd:string [1]
1
1..*
0..1
1
1
1
0..*
0..*
0..*
0..*
1
1
1
For clarity specializations of the AbstractParameter class are not shown on this diagram

Figure 3-7: Service Management Service Package Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.7.2 CLASS AbstractSrvPkg (ABSTRACT)

3.7.2.1 The AbstractSrvPkg is an abstract class that may be used to instantiate the various classes of service packages required by service management.

3.7.2.2 The AbstractSrvPkg class parameter specified in table 3-20 shall be used to provide a reference to the SICF to be used.

Table 3-20: Class AbstractSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`sicfRef`	Optional parameter: A reference to the existing SICF that is to be used.	xsd:string	n/a

3.7.3 CLASS AbstractOnlineSrvPkg (ABSTRACT)

3.7.3.1 AbstractOnlineSrvPkg is an abstract class that may be used to instantiate the various classes of online service packages required by service management.

3.7.3.2 If required, an Agency- (or mission-) specific online service package class may be derived from this class.

NOTE – The AbstractOnlineSrvPkg class is a specialization of class AbstractSrvPkg described in 3.7.2, and the generic description of the parameters is given in that subsection.

3.7.3.3 The AbstractOnlineSrvPkg class parameter specified in table 3-21 shall be used to provide a reference to the trajectory data to be used.

Table 3-21: Class AbstractOnlineSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`trajectoryRef`	Optional parameter. A reference to the existing trajectory data that is to be used.	xsd:string	n/a

3.7.4 CLASS OnlineSrvPkgType1

3.7.4.1 OnlineSrvPkgType1 class may be used to specify data relevant to an online service package.

NOTE – The OnlineSrvPkgType1 class is a specialization of class AbstractOnlineSrvPkg described in 3.7.3, and the generic description of the parameters is given in that subsection.

3.7.4.2 Each instance of the OnlineSrvPkgType1 class must contain 1 or more OnlineSrvPkgType1Details classes (see 3.7.5).

3.7.4.3 Each instance of the OnlineSrvPkgType1 class may contain 0 or more ModResParm classes (see 3.5.2).

3.7.4.4 The OnlineSrvPkgType1 class parameter specified in table 3-22 shall be used to provide a reference to the Event Sequence to be used.

Table 3-22: Class OnlineSrvPkgType1 Parameters

Parameter	Description	Data Type	Data Units
`eventSequenceRef`	Optional parameter. A reference to the existing event sequence that is to be used.	xsd:string	n/a

3.7.5 CLASS OnlineSrvPkgType1Details

3.7.5.1 OnlineSrvPkgType1Details class may be used to specify details relevant to an online service package.

3.7.5.2 Each instance of the OnlineSrvPkgType1Details class may contain 0 or more DDORScanPatternDetails classes (see 3.7.6).

3.7.5.3 Each instance of the OnlineSrvPkgType1Details class may contain 0 or more ModResParm classes (see 3.5.2).

3.7.5.4 The OnlineSrvPkgType1Details class parameter specified in table 3-23 shall be used to provide a reference to the configuration profile to be used.

Table 3-23: Class OnlineSrvPkgType1Details Parameters

Parameter	Description	Data Type	Data Units
`configProfileRef`	A reference to the existing configuration profile that is to be used.	xsd:string	n/a

3.7.6 CLASS DDORScanPatternDetails

3.7.6.1 DDORScanPatternDetails class may be used to specify a reference to an existing DDOR Scan Pattern.

3.7.6.2 Each instance of the DDORScanPatternDetails class may contain 0 or more ModResParm classes (see 3.5.2).

3.7.6.3 The DDORScanPatternDetails class parameter specified in table 3-24 shall be used to provide a reference to the DDOR Scan Pattern to be used.

Table 3-24: Class DDORScanPatternDetails Parameters

Parameter	Description	Data Type	Data Units
`DDORScanPatternRef`	A reference to the existing DDOR Scan Pattern that is to be used.	xsd:string	n/a

3.7.7 CLASS OfflineSrvPkgType1

3.7.7.1 OfflineSrvPkgType1 class may be used to specify data relevant to an offline service package.

NOTE – The OfflineSrvPkgType1 class is a specialization of class AbstractSrvPkg described in 3.7.2, and the generic description of the parameters is given in that subsection.

3.7.7.2 Each instance of the OfflineSrvPkgType1 class may contain 0 or more ModResParm classes (see 3.5.2).

3.7.7.3 The OfflineSrvPkgType1 class parameter specified in table 3-25 shall be used to provide a reference to the Configuration Profile to be used.

Table 3-25: Class OfflineSrvPkgType1 Parameters

Parameter	Description	Data Type	Data Units
`configProfileRef`	A reference to the existing configuration profile that is to be used.	xsd:string	n/a

ANNEX A

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

A1 SECURITY CONSIDERATIONS

Security considerations are not applicable, as these are addressed in the Recommended Standards and Practices that reference the data structures contained in this document.

A2 SANA CONSIDERATIONS

A2.1 GENERAL

The recommendations of this document rely on the SANA registries described below. New assignments in these registries, in conformance with the policies identified, will be available at the SANA registry Web site: https://sanaregistry.org. Therefore the reader shall look at the SANA Web site for all the assignments contained in these registries.

A2.2 REGISTRY FOR `originatingOrganization`

The values for originatingOrganization (see table 3-1) are those listed in the 'Abbreviation' field of the CCSDS Organizations registry. The Organizations registry is located at:

https://sanaregistry.org/r/organizations/

In accordance with reference [4], subsection 3.11 (c), updates to this registry are at the discretion of CCSDS member agencies or registered organizations, via the registered agency or organization representative.

The CCSDS Organizations registry also lists roles that organizations fulfil. The Organization Roles registry is located at:

https://sanaregistry.org/r/organization-roles

The following roles from the organization-roles registry are relevant to Service Management:

– 'Cross Support Service Provider'	This role is assigned to organizations that provide planning info, service packages, and suchlike (i.e., the responses to requests).
– 'Cross Support Service User'	This role is assigned to organizations that submit requests to Cross Support Service Providers.

Organizations may be assigned any combination of the above and additional roles.

A2.3 REGISTRY FOR `user`

The values for the `user` parameter (see table 3-2) are typically those listed in the Spacecraft Abbreviation field of the CCSDS Spacecraft Identifiers registry. The Spacecraft Identifiers registry is located at:

https://sanaregistry.org/r/spacecraft/

In accordance with reference [4], subsection 3.11 (c), updates to this registry are at the discretion of CCSDS member agencies or registered organizations, via the registered agency or organization representative.

For values of the `user` parameter that are not spacecraft, the using recommendation shall identify the appropriate registry for such values.

A2.4 REGISTRY FOR SITEREF AND APERTUREREF

The values for the `siteRef` parameter (see table 3-5) of this Recommended Practice are those of the Site Name Abbreviation field of the Service Sites and Apertures registry. The Service Sites and Apertures registry is located at:

https://sanaregistry.org/r/service_sites_apertures/

Although the registry allows for Site Name records with no Aperture Name field records, for effective use with this Recommended Practice, any referenced Site Name needs to contain at least one Aperture Name field. It may contain as many fields as needed to list as many apertures found at one site as needed. The Aperture Name and Aperture Name Abbreviations are unique with respect to any other Aperture Name or Aperture Name Abbreviations for the particular Site Name.

The values for the apertureRef parameter (see table 3-5) of this Recommended Practice are those of the Aperture Name Abbreviation field.

In accordance with reference [4], subsection 3.11 (c), updates to this registry are at the discretion of CCSDS member agencies or registered organizations, via the registered agency or organization representative.

A2.5 REGISTRY FOR EVENT TYPES

The event types COMMS, DATARATE, RFI, and CONFLICTS (defined in table 3-2) are registered as part of tracking the overall set of CCSDS event types. The registry for event types is located at:

https://sanaregistry.org/r/event_types

The update policy for this registry is identified in [6], subsection A2.2.

A2.6 USE OF UNREGISTERED VALUES

Only values that have been registered should be used for the items indicated in A2.2 through A2.4. Unregistered values may be used. If unregistered values are used, they should be prefixed with the string ‘UNR::’.

NOTES

1 ‘UNR::’ indicates an unregistered value.

2 This helps eliminate potential confusion in a multi-agency cross support context.

3 Use of unregistered values is not recommended and should be avoided if possible.

A3 PATENT CONSIDERATIONS

No patent rights are known to apply to any of the specifications of the Recommended Practice.

ANNEX B

XML SCHEMA ORGANIZATION AND PACKAGING FOR THE COMMON DATA ENTITY CLASSES

(INFORMATIVE)

B1 PURPOSE

This annex provides an informative description of XML schema organization and packaging for the Common Data Entity classes as defined in section 3.

B1.1 SCHEMA HISTORY

The files given in table B-1 constitute the Common Data Entity schemas.

Table B-1: Schema History

Schema File	History
`902x12m1-SmCmnEnt-InfEntHdr.xsd`	Unchanged since initial release with Issue 1 of Service Management Common data Entities Magenta Book (902.12-M-1).
`902x12m1-SmCmnEnt-AbsEvnt.xsd`	Unchanged since initial release with Issue 1 of Service Management Common data Entities Magenta Book (902.12-M-1).
`902x12m1-SmCmnEnt-CmnClss.xsd`	Unchanged since initial release with Issue 1 of Service Management Common data Entities Magenta Book (902.12-M-1).
`902x12m2-SmCmnEnt-TLE.xsd`	Initial release with Issue 2 of Service Management Common data Entities Magenta Book (902.12-M-2).
`902x12m2-SmCmnEnt-DataWrp.xsd`	Initial release with Issue 2 of Service Management Common data Entities Magenta Book (902.12-M-2).
`902x12m2-SmCmnEnt-SrvPkg.xsd`	Initial release with Issue 2 of Service Management Common data Entities Magenta Book (902.12-M-2).

B2 SCHEMA ORGANIZATION AND PACKAGING

B2.1 SERVICE MANAGEMENT INFORMATION ENTITY HEADER SCHEMA

The normative Service Management Information Entity Header schema types and global elements are contained in the file ‘902x12m1-SmCmnEnt-InfEntHdr.xsd’.

The SM Information Entity Header types and global elements are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The Service Management Information Entity Header schema includes the following schema:

– 902x13m1-AbsEvnt-TimCde.xsd

  Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

The source of the following schema files

– 902x12m1-SmCmnEnt-InfEntHdr.xsd

– 902x13m1-AbsEvnt-TimCde.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

B2.2 SERVICE MANAGEMENT ABSTRACT EVENT AND SERVICE MANAGEMENT EVENT TIME SCHEMA

The normative Service Management Abstract Event and Service Management Event Time schema types and global elements are contained in the file '902x12m1-SmCmnEnt-AbsEvnt.xsd'.

The SM Abstract Event and Service Management Event Time schema types and global elements are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

The Service Management Abstract Event and Service Management Event Time schema includes the following schema:

– 902x13m1-AbsEvnt-TimCde.xsd

  Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

The source of the following schema files

– 902x12m1-SmCmnEnt-AbsEvnt.xsd

– 902x13m1-AbsEvnt-AbsParm.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

B2.3 SERVICE MANAGEMENT COMMON CLASSES SCHEMA

The normative Service Management Common Classes schema types and global elements are contained in the file '902x12m1-SmCmnEnt-CmnClss.xsd'.

The SM Common Classes schema types and global elements are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The Service Management Common Classes schema includes the following schema:

– 902x13m1-AbsEvnt-TimCde.xsd

  Types and global elements in this schema are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The source of the following schema files

– 902x12m1-SmCmnEnt-CmnClss.xsd

– 902x13m1-AbsEvnt-TimCde.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

B2.4 SERVICE MANAGEMENT TLE SCHEMA

The normative Service Management TLE schema types and global elements are contained in the file ‘902x12m2-SmCmnEnt-TLE.xsd’.

The SM TLE schema types and global elements are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The source of the following schema file

– 902x12m2-SmCmnEnt-TLE.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

B2.5 SERVICE MANAGEMENT DATA WRAPPER CLASSES SCHEMA

The normative Service Management Data Wrapper Classes schema types and global elements are contained in the file ‘902x12m2-SmCmnEnt-DataWrp.xsd’.

The SM Data Wrapper Classes schema types and global elements are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The Service Management Data Wrapper Classes schema includes the following schemas:

– 902x12m2-SmCmnEnt-TLE.xsd

Types and global elements in this schema are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

- ndmxml-2.0.0-master-2.0.xsd

  Types and global elements in this schema are registered in the ‘urn:ccsds:schema:ndmxml’ name space.

The source of the following schema files

- 902x12m2-SmCmnEnt-DataWrp.xsd
- 902x12m2-SmCmnEnt-TLE.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

The source of the following schema file

- ndmxml-2.0.0-master-2.0.xsd

is the SANA Navigation Data Messages XML Schema (Unqualified) registry:

https://sanaregistry.org/r/ndmxml_qualified

B2.6 SERVICE MANAGEMENT SERVICE PACKAGE CLASSES SCHEMA

The normative Service Management Service Package Classes schema types and global elements are contained in the file ‘902x12m2-SmCmnEnt-SrvPkg.xsd’.

The SM Service Package Classes schema types and global elements are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The Service Management Service Package Classes schema includes the following schemas:

- 902x12m1-SmCmnEnt-CmnClss.xsd

  Types and global elements in this schema are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The source of the following schema files

- 902x12m2-SmCmnEnt-SrvPkg.xsd
- 902x12m1-SmCmnEnt-CmnClss.xsd

is the SANA SCCS-SM Information Entity XML Schemas registry:

https://sanaregistry.org/r/service_management_xml_schemas

ANNEX C

INFORMATIVE REFERENCES

(INFORMATIVE)

[C1] *Space Communication Cross Support—Service Management—Operations Concept.* Issue 1-S. Report Concerning Space Data System Standards (Historical), CCSDS 910.14-G-1-S. Washington, D.C.: CCSDS, (May 2011) June 2017.

ANNEX D

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

Term	Meaning
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CSSM	Cross Support Service Management
CSSS	Cross Support Service System
DDOR	Delta-Differential One-Way Ranging
FB	frequency band
ICD	interface control document
ICS	implementation conformance statement
n/a	not applicable
NORAD	North American Aerospace Defense
OEM	Orbit Ephemeris Message
OMG	Object Management Group
OMM	Orbit Ephemeris Message
OPM	Orbit Parameter Message
SANA	Space Assigned Numbers Authority
SATCAT	Satellite Catelog
SCCS SM	Space Communication Cross Support Service Management
SICF	Service Instance Configuration File
SM	Service Management
TLE	Two-line element
UML	Unified Modeling Language
UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
XML	Extensible Markup Language