

Draft Recommendation for  
Space Data System Practices

|  |
| --- |
| Cross Support Service Management—Best Practices |

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Issue: | , |  |
|  | Date: |  |  |
|  | 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 ‘White Book’ status indicates that its contents are not stable, and several iterations resulting in substantial technical changes are likely to occur before it is considered to be sufficiently mature to be released for review by the CCSDS Agencies.

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** |
|  | , , |  | Initial draft. |

CONTENTS

Section Page

[1 Introduction 1-1](#_Toc185612109)

[1.1 PURPOSE AND SCOPE 1-1](#_Toc185612110)

[1.2 APPLICABILITY 1-1](#_Toc185612111)

[1.3 RATIONALE 1-2](#_Toc185612112)

[1.4 DOCUMENT STRUCTURE 1-2](#_Toc185612113)

[1.5 DEFINITIONS 1-2](#_Toc185612114)

[1.6 NOMENCLATURE 1-2](#_Toc185612115)

[1.7 CONVENTIONS 1-3](#_Toc185612116)

[1.8 References 1-4](#_Toc185612117)

[2 Overview 2-1](#_Toc185612118)

[2.1 GENERAL 2-1](#_Toc185612119)

[2.2 Data Entities 2-1](#_Toc185612120)

[2.3 MAPPING TO W3C XML SCHEMA 2-1](#_Toc185612121)

[3 Service Management Levels 3-1](#_Toc185612122)

[3.1 Overview 3-1](#_Toc185612123)

[3.2 Service management – Level 0 3-3](#_Toc185612124)

[ANNEX A Security, SANA, And Patent Considerations (Informative) A-1](#_Toc185612125)

[ANNEX B XML Schema Organization And Packaging for THE Common Data Entity CLASSes (InFormative) B-1](#_Toc185612126)

[ANNEX C Abbreviations and Acronyms (INFORMATIVE) C-2](#_Toc185612127)

CONTENTS (continued)

Figure Page

[1-1 Best Practises in the Context of Space Communication Cross Support Service Management 1-1](#_Toc185612128)

[3-1 Service Management Level 0 – Service Management Utilization Request Formats Class Diagram 3-5](#_Toc185612129)

[3-2 Service Management Level 0 – Service Package Data Formats Class Diagram 3-16](#_Toc185612130)

[3-3 Service Management Level 0 – Simple Schedule Data Format Class Diagram 3-20](#_Toc185612131)

Table

[3-1 Class SrvMgtUtilReqHeader Parameters 3-6](#_Toc185612132)

[3-2 Class NewOnlineSrvPkgReq Parameters 3-8](#_Toc185612133)

[3-3 Class ReplaceOnlineSrvPkgReq Parameters 3-9](#_Toc185612134)

[3-4 Class DeleteSrvPkgReq Parameters 3-10](#_Toc185612135)

[3-5 Class ReplaceSrvPkg Parameters 3-11](#_Toc185612136)

[3-6 Class DeleteSrvPkg Parameters 3-11](#_Toc185612137)

[3-7 Class BasicPass Parameters 3-12](#_Toc185612138)

[3-8 Class ApertureSelection Parameters 3-14](#_Toc185612139)

[3-9 Class ApertureReference Parameters 3-14](#_Toc185612140)

[3-10 Class OnlineSrvPkgType1 Parameters 3-15](#_Toc185612141)

[3-11 Class OnlineSrvPkgType1Details Parameters 3-15](#_Toc185612142)

[3-12 Class SrvPkgHeader Parameters 3-17](#_Toc185612143)

[3-13 Class ServicePkgBody Parameters 3-18](#_Toc185612144)

[3-14 Class ServicePkg Parameters 3-19](#_Toc185612145)

[3-15 Class SimpleScheduleHeader Parameters 3-21](#_Toc185612146)

[3-16 Class ScheduledPackage Parameters 3-23](#_Toc185612147)

[3-17 Class ScheduledActivity Parameters 3-23](#_Toc185612148)

[3-18 Class ServiceInfo Parameters 3-26](#_Toc185612149)

[3-19 Class ServicePackageXRef Parameters 3-27](#_Toc185612150)

# Introduction

## PURPOSE AND SCOPE

### PURPOSE

The purpose of this Magenta Book is to describe best practises in the context of Cross Support Service Management.

### SCOPE

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

Figure 1‑1 puts the Cross Support Service Management—Best Practices into context with the various standards that together form the Space Communication Cross Support Service Management.

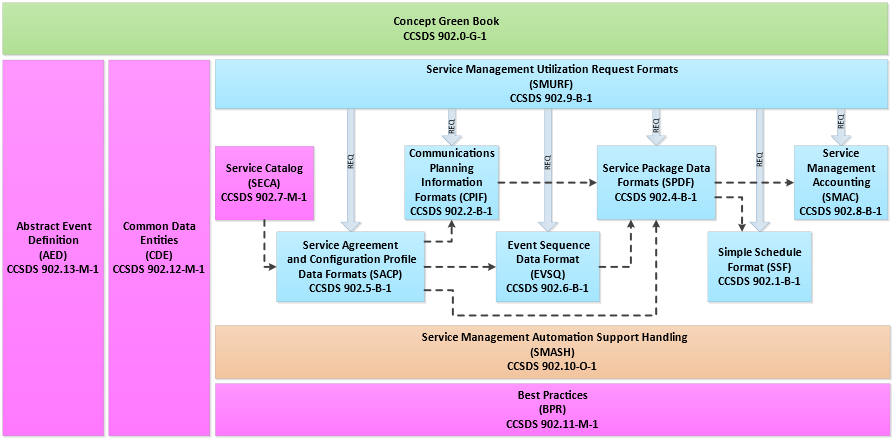


Figure 1‑1 : Best Practises in the Context of Space Communication Cross Support Service Management

## APPLICABILITY

This recommended practise applies across the whole scope of service management. It should however be noted that not all of the Service Management Level outlined in this document are applicable to all Service Management Recommended Standards and Practises.

## RATIONALE

The rationale for this document is to illustrate how the various Service Management Recommended Standards and Practices can be used to support missions of varying degrees of complexity. This is done by defining Service Management “Levels”, where the differing levels deal with different levels of complexity, with Level-0 outlining the simplest way in which service management may be used to support a straightforward mission with no complex scheduling requirements.

## DOCUMENT STRUCTURE

This document is organized as follows:

1. 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.
2. Section 2 provides a brief overview of the common data entities.
3. Section 3 provides details about the various Service Management Levels.
4. Annex A discusses security, Space Assigned Numbers Authority (SANA), and patent considerations.
5. Annex B provides an informative listing of the XML schema for the service management information entity header.
6. Annex C contains a list of abbreviations and acronyms applicable to the Service Management Common Data Entities.

## DEFINITIONS

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

**agency**: A satellite operator or satellite service provider.

## NOMENCLATURE

### Normative Text

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

1. the words ‘shall’ and ‘must’ imply a binding and verifiable specification;
2. the word ‘should’ implies an optional, but desirable, specification;
3. the word ‘may’ implies an optional specification;
4. 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.

### 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.

## CONVENTIONS

### THE UNIFIED MODELING LANGUAGE

The Unified Modelling Language (UML) is a general-purpose, developmental modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. It supports various diagram types, including class diagrams, package diagrams, sequence diagrams, and activity diagrams. In this document notation, semantics, and conventions imposed by the Version 2.4.1 UML specification of the Object Management Group (OMG) (reference [2]) are followed.

Within the document use is made only of class diagrams. A UML class diagram describes the structure of a message, its parts, and how those parts interrelate. A UML class, represented in the diagram as a box, represents a data set. Class diagram conventions include composition, generalization, multiplicity, and constraints. Enumeration notation is also used but only when it is involved in a composition constraint.

### 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 [3]). Wherever an XML Schema datatype is indicated in this Recommended Practice, it is prefixed with the string ‘xsd:’.

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

[4] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.9-B-1. Washington, D.C.: CCSDS, Forthcoming.

[5] *Cross Support Service Management—Service Package Data Formats*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.4-B-1. Washington, D.C.: CCSDS, Forthcoming.

[6] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.1-B-2. Washington, D.C.: CCSDS, Forthcoming

[] *CCSDS Spacecraft Identification Field Code Assignment Control Procedures*. Issue 7. Recommendation for Space Data System Practices (Magenta Book), CCSDS 320.0-M-7. Washington, D.C.: CCSDS, November 2017.

# Overview

## GENERAL

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

## 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.

## 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.

# Service Management Levels

## Overview

Service management is intended to support all types of missions, from complex deep space missions, through constellations, to simple LEO missions. Due to this vast spread in mission complexity service management has to be complex in order to support the diverse requirements.

This complexity is however not required in all cases. In this Best Practises a number of Service Management Levels are defined which highlight the appropriate elements to be used for specific use cases.

In this issue of the document SM – Level 0 is defined, the simplest case. Future versions are expected to address more complex cases.

### Conventions

For clarity in the following sections, *abstract* classes in the class diagrams are highlighted in green. Unlike in the Recommended Standards/Practices where these classes are defined, no discussion of the abstract classes is contained in this document. Instead a description of the parameters contained in an abstract class is given where the abstract class is realized into a concrete class.

If the reader wishes to find full details of the abstract class they are should refer to the Recommended Standard or Practice where they are fully defined.

The class diagrams presented in this document are “cut down” versions of those in the documents where they are originally defined, with only those classes relevant to a particular Service management Level being shown.

It should be noted however that despite the class diagrams being cut down all SM Levels defined in this Recommended Practice are fully compatible with the “complete” version and that any XML file produced from this Recommended Practice will validate against the standard SM Schema files.

#### Colour Coding

In the tables describing parameters of the various classes the following colour coding is used;

|  |
| --- |
| * Red background; indicates that the parameter in the overall context of Service Management is optional but for the particular SM Level under discussion it is not required and shall not be specified. |
| * Yellow background; indicates that the parameter in the overall context of Service Management is optional and that it may be specified for the particular SM Level under discussion. |
| * Green background; indicates that the parameter is either mandatory or, in the overall context of Service Management optional but for the particular SM Level under discussion it is considered mandatory and shall be specified. |

#### Identifiers and References

It is obviously necessary to keep track of the various data entities created during service management activities. By convention Service management adopts the following approach;

1. When a data entity is created it is given an Identifier
2. When it is necessary to refer to an existing data entity this is done by a reference.

For example if a new service request is submitted by means of the SMURF NewOnlineSrvPkgReq class (see e.g. Section 3.2.2.1.3) two IDs are required to be provided by the User CSSS

* requestID – this is the Identifier of the request itself and shall be unique for every SMURF request submitted by the User CSSS
* serviceReqID – this is the Identifier of the service request contained in the NewOnlineSrvPkgReq. This is in effect creating a service package request. The serviceReqID could be of the form (and again shall be unique for every service request submitted by the User CSSS);
  + *XMM-Service-Request-0001*

Any subsequent request to act on the service package request needs to reference the service request Identifier, e.g. if a request is made to modify the service package request by means of the ReplaceOnlineServicePkgReq class (see e.g. Section 3.2.2.1.4) the following would need to be specified;

* requestID – this is the Identifier of the request itself and shall be unique for every SMURF request submitted by the User CSSS
* serviceReqRef – this is the Reference to the Identifier of the service request contained in a previously submitted NewOnlineSrvPkgReq request. In the case where it was desired to modify a service package request created by a NewOnlineSrvPkgReq that had an serviceReqID of *XMM-Service-Request-0001*, then the reference contained in serviceReqRef would be;
  + *XMM-Service-Request-0001*

Similar considerations apply to the Identifiers and References contained in other data entities used in Service Management.

It should also be noted that the Identifier may not always be created during an operation directly supported by Service Management. For example SM Level-0 does not support operations to submit Service Agreements or Configuration Profiles, but it is does support references to the Identifiers of Service Agreements (see e.g. section 3.2.2.1.3) and Configuration Profiles (see e.g. section 3.2.2.1.12). In such cases it is necessary that the participating parties agree on the appropriate Identifiers for the data entities (possibly documenting this in an ICD), so that these Identifiers can then be Referenced as required by the data entities.

## Service management – Level 0

### Overview

#### Assumptions

Service Management – Level 0 is the simplest usage of service management. In defining the use case for this the following assumptions are made;

1. Any required Trajectory information passed between the parties involved is done so outwith the scope of SM – Level 0. The mechanism for this should be documented in an ICD agreed by the involved parties.
2. Any required Service Instance Configuration Files (SICF) passed between the parties involved is done so outwith the scope of SM – Level 0. The mechanism for this should be documented in an ICD agreed by the involved parties.
3. Any required Configuration Profiles passed between the parties involved is done so outwith the scope of SM – Level 0. The mechanism for this should be documented in an ICD agreed by the involved parties.
4. Event Sequences are not required.
5. Delta DOR scan patterns are not required.
6. Only basic constraints are required to specify the required station support, that is
   1. Preferred start time
   2. Optional window around preferred start time
   3. Preferred duration
   4. Optional flexibility around preferred duration
7. Aperture selection can be done by specifying a site reference and an optional aperture reference, i.e. selecting an aperture by specifying its location on a celestial body is not required.

If any of these assumptions are invalid then SM – Level 0 is not the appropriate level of service management to use.

#### Prerequisites

The following are necessary prerequisites for the use of SM – Level 0;

1. A Service Agreement exists between the parties involved.
2. At least 1 Configuration Profile has been defined.
3. The parties involved have agreed the name of the Configuration Profile. If more than one Configuration Profile has been defined each shall have its own unique name agreed between the parties involved.

### Service Management Utilization Request Formats

#### Overview

The following class diagram shows the classes of the Service Management Utilization Request Formats (SMURF) that are relevant to SM Level-0.

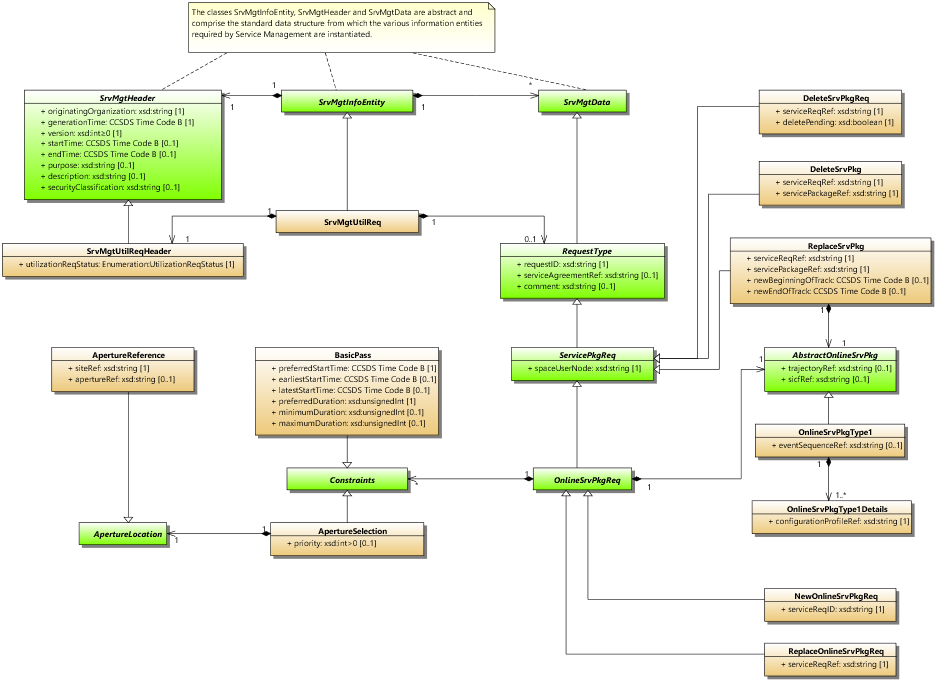


Figure 3‑1 : Service Management Level 0 – Service Management Utilization Request Formats Class Diagram

For SM Level-0 the SMURF supports 5 types of request;

1. Submission of a new service package request (class NewOnlineSrvPkgReq)
2. Replacement of an existing service package request (class ReplaceOnlineSrvPkgReq)
3. Delete an existing service package request (class DeleteSrvPkgReq)
4. Replace an existing service package (class ReplaceSrvPkg)
5. Delete an existing service Package (class DeleteSrvPkg)

These, and the other required concrete classes, are described in the following sections. Full descriptions of all SMURF classes and parameters can be found in Ref. [4].

##### Class SrvMgtUtilReq

This class is effectively the “wrapper” for a Service Management Utilization Request. It does not have any parameters itself but contains the following classes;

* One instance of the SrvMgtUtilreqHeader class (see section 3.2.2.1.2)
* Zero or one instance of one of the following classes;
  + NewOnlineSrvPkgReq (see section 3.2.2.1.3)
  + ReplaceOnlineSrvPkgReq (see section 3.2.2.1.4)
  + DeleteSrvPkgReq (see section 3.2.2.1.5)
  + ReplaceSrvPkg (see section 3.2.2.1.6)
  + DeleteSrvPkg (see section 3.2.2.1.7)

##### Class SrvMgtUtilReqHeader

This class contains the “Header” for the Service Management Utilisation Request. It does not contain any other classes, but contains the parameters as described in table 3‑1.

Table 3‑1 : Class SrvMgtUtilReqHeader Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| originatingOrganization | The organization that generated the information entity. | xsd:string  Permitted values are registered in SANA. | 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.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | 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.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | 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 an ICD between relevant parties. | n/a |
| description | Optional parameter. May be used to describe the contents of the information entity. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| securityClassification | Optional parameter. May be used to specify a security classification. Definitions of security classification are outwith the scope of this recommendation and would be expected to be agreed by the parties involved. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| utilizationReqStatus | The status of the service management utilization request. | Enumeration  The following values are permitted   * TEST indicates that the request has been generated for test purposes only. * OPERATIONAL  indicates that this is an operational request. | n/a |

##### Class NewOnlineSrvPkgReq

This class contains a new Service Package Request, i.e. in effect it creates a Service Package Request. It contains 0 or more Constraint classes, in the context of SM Level-0 the permitted constraint classes are;

* BasicPass (see section 3.2.2.1.8)
* ApertureSelection (see section 3.2.2.1.9)

Additionally, it contains the parameters as described in table 3‑2.

Table 3‑2 : Class NewOnlineSrvPkgReq Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| requestID | A unique request ID assigned by the requestor. | xsd:string | n/a |
| serviceAgreementRef | Optional Parameter.  This can be used to specify a reference to the service agreement under which the requested services are to be provided. | xsd:string | n/a |
| comment | Optional Parameter.  May be used for the provision of ad hoc information. | xsd:string | n/a |
| spaceUserNode | The user of the requested services. These must be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | n/a |
| serviceReqID | This parameter is a identifier that uniquely identifies the request. | xsd:string | n/a |

NOTE – It is the responsibility of the User CSSS to ensure that the serviceReqID is unique for the combination of serviceAgreementRef and spaceUserNode for which the request is being submitted. The provider CSSS can then assume that the combination of serviceAgreementRef, spaceUserNode, and serviceReqID constitutes a unique identifier for a service request.

##### Class ReplaceOnlineSrvPkgReq

This class contains a Replace Service Package Request, i.e. it contains a Service Package Request to replace one that was submitted previously. The identification of the Service package Request that is to be replaced is done by the use of the serviceReqID parameter, which is assigned by the user CSSS when a new service package request is submitted. It contains 0 or more Constraint classes, in the context of SM Level-0 the permitted constraint classes are;

* BasicPass (see section 3.2.2.1.8)
* ApertureSelection (see section 3.2.2.1.9)

Additionally, it contains the parameters as described in table 3‑3.

Table 3‑3 : Class ReplaceOnlineSrvPkgReq Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| requestID | A unique request ID assigned by the requestor. | xsd:string | n/a |
| serviceAgreementRef | Optional Parameter.  This can be used to specify a reference to the service agreement under which the requested services are to be provided. | xsd:string | n/a |
| comment | Optional Parameter.  May be used for the provision of ad hoc information. | xsd:string | n/a |
| spaceUserNode | The user of the requested services. These must be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | n/a |
| serviceReqRef | This parameter is the Identifier of the service request that is to be updated. | xsd:string | n/a |

##### Class DeleteSrvPkgReq

This class contains a Delete Service Package Request, i.e. it contains a Service Package Request to delete one that was submitted preiously. The identification of the Service package Request that is to be deleted is done by the use of the serviceReqID parameter, which is assigned by the user CSSS when a new service package request is submitted. It does not contain any other classes, but contains the parameters as described in table 3‑4.

Table 3‑4 : Class DeleteSrvPkgReq Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| requestID | A unique request ID assigned by the requestor. | xsd:string | n/a |
| serviceAgreementRef | Optional Parameter.  This can be used to specify a reference to the service agreement under which the requested services are to be provided. | xsd:string | n/a |
| comment | Optional Parameter.  May be used for the provision of ad hoc information. | xsd:string | n/a |
| spaceUserNode | The user of the requested services. These must be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | n/a |
| serviceReqRef | This parameter is the Identifier of the service request that is to be deleted. | xsd:string | n/a |
| deletePending | This is a Boolean flag that indicates if service packages that are pending execution are also requested to be deleted.   * TRUE means yes, the request is also to delete those the pending service packages; * FALSE means, no, the request does not apply to pending service packages | xsd:boolean | n/a |

##### Class ReplaceSrvPkg

This class contains a request to a Replace a Service Package, i.e. it contains a Service Package that is to replace one that was previously generated. The identification of the Service package that is to be replaced is done by the use of the servicePackageRef parameter, which is assigned by the provider CSSS when a new service package is generated. It contains exactly 1 of the class OnlineSrvPkgType1 (see section 3.2.2.1.11).

Additionally, it contains the parameters as described in table 3‑5.

Table 3‑5 : Class ReplaceSrvPkg Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| requestID | A unique request ID assigned by the requestor. | xsd:string | n/a |
| serviceAgreementRef | Optional Parameter.  This can be used to specify a reference to the service agreement under which the requested services are to be provided. | xsd:string | n/a |
| comment | Optional Parameter.  May be used for the provision of ad hoc information. | xsd:string | n/a |
| spaceUserNode | The user of the requested services. These must be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | n/a |
| serviceReqRef | In the context of the ReplaceSrvPkg class, this parameter must refer to the identifier of the request that resulted in the generation of the service package that is now to be replaced. | xsd:string | n/a |
| servicePackageRef | This parameter is the Identifier of the request that resulted in the generation of the service package that is now to be replaced. | xsd:string | n/a |
| newBeginningOfTrack | Optional parameter.  Used to specify (in UTC) the new time at which the pass should start. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| newEndOfTrack | Optional parameter.  Used to specify (in UTC) the new time at which the pass should finish. | CCSDS ASCII Time Code B (reference [1]) | UTC |

##### Class DeleteSrvPkg

This class contains a request to Delete a Service Package, i.e. it contains the reference of a Service Package that is to be deleted. The identification of the Service package that is to be deleted is done by the use of the servicePackageRef parameter, which is assigned by the provider CSSS when a new service package is generated. It does not contain any other classes, but contains the parameters as described in table 3‑6.

Table 3‑6 : Class DeleteSrvPkg Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| requestID | A unique request ID assigned by the requestor. | xsd:string | n/a |
| serviceAgreementRef | Optional Parameter.  This can be used to specify a reference to the service agreement under which the requested services are to be provided. | xsd:string | n/a |
| comment | Optional Parameter.  May be used for the provision of ad hoc information. | xsd:string | n/a |
| spaceUserNode | The user of the requested services. These must be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | n/a |
| serviceReqID | This parameter is the Identifier of the service request that resulted in the generation of the service package(s) that is(are) now to be deleted. | xsd:string | n/a |
| servicePackageRef | This is used to specify the reference of the service package that is to be replaced. | xsd:string | n/a |

##### Class BasicPass

This class may be used to specify the preferred start time and duration of a pass along with the acceptable earliest and/or latest start times and minimum and/or maximum acceptable duration. It does not contain any other classes, but contains the parameters as described in table 3‑7.

Table 3‑7 : Class BasicPass Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| PreferredStartTime | Used to specify (in UTC) the preferred time at which a pass starts.  NOTES   1. If an earliestStartTime is specified, then PreferredStartTime ≥ earliestStartTime. 2. If a latestStartTime is specified, then PreferredStartTime ≤ latestStartTime. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| earliestStartTime | Optional parameter.  Used to specify (in UTC) the earliest time at which a pass can start. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| latestStartTime | Optional parameter.  Used to specify (in UTC) the latest time at which a pass can start.  NOTE – If both earliestStartTime and latestStartTime are specified, then latestStartTime ≥ earliestStartTime. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| preferredDuration | This is used to specify the preferred duration of a pass.  NOTES   1. If a minimumDuration is specified then preferredDuration ≥ minimumDuration. 2. If a maximumDuration is specified, then preferredDuration ≤ maximumDuration. | xsd:unsignedInt | Secs |
| minimumDuration | Optional parameter.  This is used to specify the minimum duration required for a pass. | xsd:unsignedInt  Default value if omitted is preferredDuration | Secs |
| maximumDuration | Optional parameter.  This is used to specify the maximum duration required for a pass.  NOTE – If both minimumDuration and maximumDuration are specified, then maximumDuration ≥ minimumDuration. | xsd:unsignedInt  Default value if omitted is preferredDuration | Secs |

##### Class ApertureSelection

This class may be used to specify an aperture with which it is desired that the service is provided. In the context of SM Level-0 it contains exactly one instance of the ApertureReference class (see section 3.2.2.1.10). Additionally, it contains the parameters as described in table 3‑8.

Table 3‑8 : Class ApertureSelection Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| priority | Optional Parameter.  This can be used to specify a preference for which aperture is used if more than one ApertureSelection class is instantiated.  The lower the value specified here the higher the priority, with 1 being the highest priority. | xsd:integer >0  Default value if omitted is 1 | n/a |

##### Class ApertureReference

This 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. It does not contain any other classes, but contains the parameters as described in table 3‑9.

Table 3‑9 : Class ApertureReference Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** | |
| --- | --- | --- | --- | --- |
| siteRef | Name of the site. | xsd:string  Permitted values are registered in SANA. | n/a |
| apertureRef | Optional Parameter.  This may be used to specify the identifier of the aperture. | xsd:string  Permitted values are registered in SANA. | n/a |

##### Class OnlineSrvPkgType1

This class may be used to specify data relevant to an online service package. In the context of SM Level-0 it must contain one or more instances of the class OnlineSrvPkgType1Details (see section 3.2.2.1.12). Additionally, it contains the parameters as described in table Table 3‑10.

Table 3‑10 : Class OnlineSrvPkgType1 Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| sicfRef | Optional parameter: A reference to the existing SICF that is to be used.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | xsd:string | n/a |
| trajectoryRef | Optional parameter. A reference to the existing trajectory data that is to be used.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | xsd:string | n/a |
| eventSequenceRef | Optional parameter. A reference to the existing event sequence that is to be used.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | xsd:string | n/a |

##### Class OnlineSrvPkgType1Details

This class may be used to specify details relevant to an online service package. It does not contain any other classes, but contains the parameters as described in table Table 3‑11.

Table 3‑11 : Class OnlineSrvPkgType1Details Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| configurationProfileRef | A reference to the existing configuration profile that is to be used. | xsd:string | n/a |

### Service Package Data Formats

#### Overview

The following class diagram shows the classes of the Service Package data Formats (SPDF) that are relevant to SM Level-0.

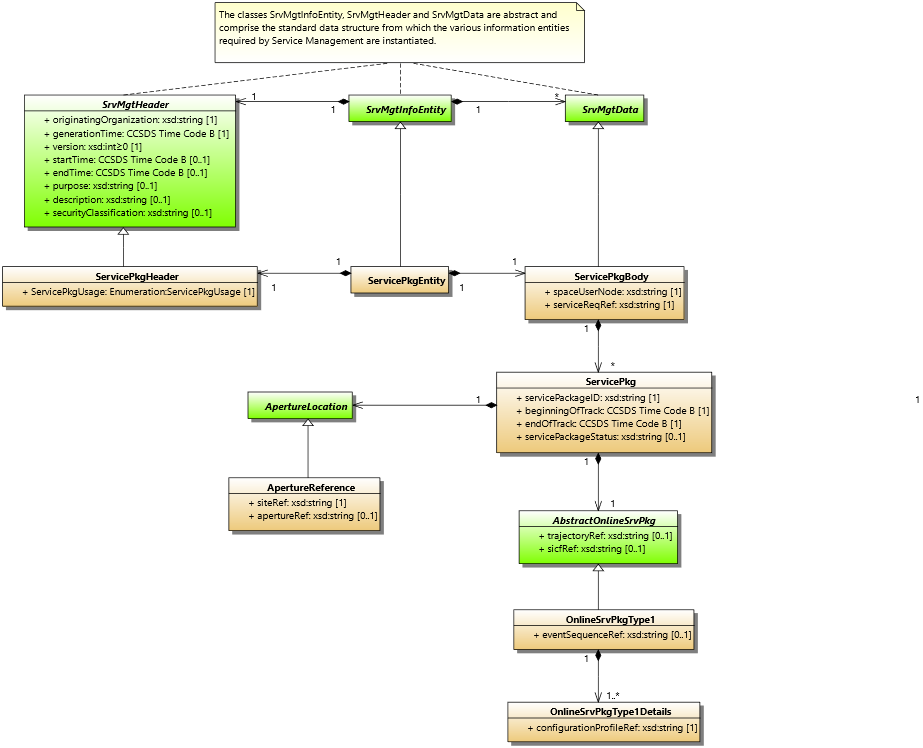


Figure 3‑2 : Service Management Level 0 – Service Package Data Formats Class Diagram

The concrete classes shown in the above figure are described in the following sections. Full descriptions of all SPDF classes and parameters can be found in Ref. [5].

##### Class SrvPkgEntity

This class is effectively the “wrapper” for the Service Package Data. It does not have any parameters itself but contains the following classes;

* One instance of the SrvPkgHeader class (see section 3.2.3.1.2)
* One instance of one of the SrvPkgBody class (see section 3.2.3.1.3)

##### Class SrvPkgHeader

This class contains the “Header” of the Service Package data. It does not contain any other classes, but contains the parameters as described in table 3‑12.

Table 3‑12 : Class SrvPkgHeader Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| originatingOrganization | The organization that generated the information entity. | xsd:string  Permitted values are registered in SANA. | 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.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | 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.  ***For all SM Level-0 Service Management Utilization Requests this parameter is not required and shall not be specified.*** | 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 an ICD between relevant parties. | n/a |
| description | Optional parameter. May be used to describe the contents of the information entity. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| securityClassification | Optional parameter. May be used to specify a security classification. Definitions of security classification are outwith the scope of this recommendation and would be expected to be agreed by the parties involved. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| servicePkgUsage | Mandatory parameter.  The intended usage of the contained set of Service Packages, whether for testing of interfaces, for operational services, or for other uses. | Enumeration  The following values are permitted:   * INTERFACE-TEST indicates that the service packages have been generated for test exchange purposes only. * OPERATIONAL  indicates that the service packages have been generated for operational purposes. * OTHER * indicates any other potential uses for the service packages. | N/A |

##### Class ServicePkgBody

This class contains the “Body” of the Service Package Data. It contains 0 or more instances of the ServicePkg class (see section 3.2.3.1.4). Additionally, it contains the parameters as described in table 3‑13.

Table 3‑13 : Class ServicePkgBody Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| spaceUserNode | The user of the scheduled package. These will typically be spacecraft names as specified in SANA. | xsd:string—Permitted values registered in SANA | N/A |
| serviceReqRef | Reference to service request identifier, in case of usage of the SMURF Request, this is value of the ServiceReqID attribute of the corresponding request. | xsd:string | N/A |

##### Class ServicePkg

This class contains the information for an individual service package included within the ServicePkgBody (see section 3.2.3.1.3). It contains the following classes;

* One instance of the ApertureReference class (see section 3.2.3.1.5).
* One instance of the class OnlineSrvPkgType1 (see section 3.2.3.1.6).

Additionally, it contains the parameters as described in table 3‑14.

Table 3‑14 : Class ServicePkg Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| servicePackageID | Identifier of the service package. | xsd:string | N/A |
| beginningOfTrack | Time at which the Service Package is scheduled to start. | CCSDS ASCII Time Code B (reference [1]) | N/A |
| endOfTrack | The time at which the Service Package is scheduled to stop. | CCSDS ASCII Time Code B (reference [1]) | N/A |
| servicePackageStatus | Optional parameter.  This may be used by the provider to indicate the status of the service package, for instance, within a service package state machine.  NOTE: whilst not mandatory it is recommended that the values used for status are;   * CREATED * SCHEDULED * DELETED * ARCHIVED * EXECUTING * ABORTED | xsd:string | N/A |

##### Class ApertureReference

See section 3.2.2.1.10 above.

##### Class OnlineSrvPkgType1

See section 3.2.2.1.11 above.

##### Class OnlineSrvPkgType1Details

See section 3.2.2.1.12 above.

### Simple Schedule Data Format

#### Overview

The following class diagram shows the classes of the Simple Schedule Format (SSF) that are relevant to SM Level-0.

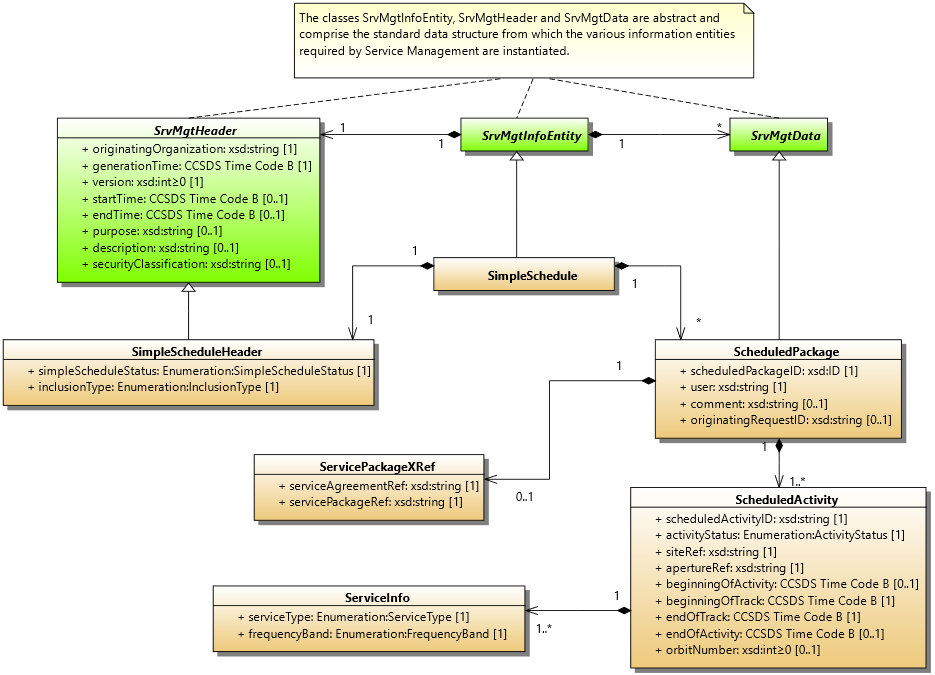


Figure 3‑3 : Service Management Level 0 – Simple Schedule Data Format Class Diagram

The concrete classes shown in the above figure are described in the following sections. Full descriptions of all SSSF classes and parameters can be found in Ref. [6].

##### Class SimpleSchedule

This class is effectively the “wrapper” for the Simple Schedule Format. It does not have any parameters itself but contains the following classes;

* One instance of the SimpleScheduleHeader class (see section 3.2.4.1.2).
* One or more instances of the ScheduledPackage class (see section 3.2.4.1.3).

##### Class SimpleScheduleHeader

This class contains the “Header” for the Simple Schedule Format. It does not contain any other classes, but contains the parameters as described in table 3‑15.

Table 3‑15 : Class SimpleScheduleHeader Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| originatingOrganization | The organization that generated the information entity. | xsd:string  Permitted values are registered in SANA. | 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 | The start time of the simple schedule contained. | xsd:string restricted to holding the time in CCSDS ASCII Time Code B format (reference [1]) | UTC |
| endTime | The end time of the simple schedule contained. | 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 an ICD between relevant parties. | n/a |
| description | Optional parameter. May be used to describe the contents of the information entity. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| securityClassification | Optional parameter. May be used to specify a security classification. Definitions of security classification are outwith the scope of this recommendation and would be expected to be agreed by the parties involved. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| simpleScheduleStatus | The status of the information entity. | Enumeration  The permitted values are:   * TEST—The schedule has been generated for test purposes only. * PROVISIONAL—The schedule contained is provisional and may still be subject to change. * OPERATIONAL—This is an operational schedule. * OTHER—This message has been generated for other purposes. In this case the optional parameter purpose can be used to further specify the purpose of the message. | n/a |
| inclusionType | The inclusion type rule that has been applied to the contained simple schedule. | Enumeration  The permitted values are   * OVERLAP\_INCLUSION  (see Ref. [6] for a description). * START\_INCLUSION  (see Ref. [6] for a description). | n/a |

##### Class ScheduledPackage

This class contains the “Body” of the Simple Schedule data. It contains the following classes;

* One or more instances of the ScheduledActivity class (see section 0).
* Zero or one instances of the class ServicePackageXRef (see section 3.2.4.1.6).

Additionally, it contains the parameters as described in table 3‑16.

Table 3‑16 : Class ScheduledPackage Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| scheduledPackageId | An identifier that is unique for every ScheduledPackage in a schedule.  NOTE – If a schedule is regenerated, then the scheduledPackageID for a particular Scheduled Package may change. | xsd:ID | n/a |
| user | The user of the scheduled package. This will typically be a spacecraft name, as specified in SANA, but may also be ‘PROVIDER-CSSS’ and ‘UNALLOCATED’, with the meanings indicated in the adjacent column. | xsd:string  Permitted values are registered in SANA.  The following strings are also permitted values with the meaning indicated:   * UNALLOCATED—The time is unallocated. * PROVIDER-CSSS—The time is allocated to the Provider CSSS (e.g., for tests, maintenance, upgrades, etc.). | n/a |
| comment | Optional parameter: may be used for the provision of ad-hoc information. | xsd:string  Exact use of this may be specified in an ICD between relevant parties. | n/a |
| originatingRequestID  (serviceReqRef) | Optional parameter: if present must echo the identifier from the request that caused the scheduled package to be generated. | xsd:string | n/a |

##### Class ScheduledActivity

This class contains the activities that are scheduled in a scheduled package. It contains one or more instances of the ServiceInfo class. Additionally, it contains the parameters as described in table 3‑17.

Table 3‑17 : Class ScheduledActivity Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| scheduledActivityID | Identifier that is unique for every ScheduledActivity in a schedule.  It should be noted that if a schedule is regenerated then the scheduledActivityID for a particular Scheduled Activity may change. | xsd:string | n/a |
| activityStatus | Used to indicate how firm the commitment to the activity is. | Enumeration  The permitted values are:   * COMMITTED—barring unforeseen circumstances the activity will be supported. * TENTATIVE—the activity is currently scheduled but may still be subject to change.   In addition to the above, one of the following two values must be used when the ScheduledPackage—user parameter is UNALLOCATED and the ServiceInfo—serviceType is UNUSED.   * AVAILABLE indicates that the aperture could be scheduled during the UNUSED time. * UNAVAILABLE indicates that the aperture is completely unavailable during the UNUSED time. | n/a |
| siteRef | The site to which the scheduled activity applies. | xsd:string  Permitted values are registered in SANA. | n/a |
| apertureRef | Identifier specifying which aperture is to be used. The apertureRef must uniquely identify the aperture at a particular site (i.e., ground station/ relay satellite) that is to be used. | xsd:string  Permitted values are registered in SANA. | n/a |
| beginningOfActivity | Optional parameter: must be present if endOfActivity parameter is present. Time at which the allocation of the site/aperture starts. This is assumed to allow for all necessary setup activities, etc.  Where available this time should be supplied subject to agency policy. If it is not possible to specify the actual value this parameter shall be omitted. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| beginningOfTrack | Time at which tracking the spacecraft is scheduled to start. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| endOfTrack | Time at which tracking the spacecraft is scheduled to end. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| endOfActivity | Optional parameter: must be present if beginningOfActivity parameter is present.  Time at which the allocation of the site/aperture ends. This is assumed to allow for all necessary teardown activities, etc. Where available this time should be supplied subject to agency policy. If it is not possible to specify the actual value this parameter shall be omitted. | CCSDS ASCII Time Code B (reference [1]) | UTC |
| orbitNumber | Optional parameter.  The Orbit Number on which the beginningOfTrack occurs. If this is not relevant (e.g., deep space missions) or not available, this parameter shall be omitted. | xsd:integer≥0 | n/a |

##### Class ServiceInfo

This class specifies the type of activities that are to be carried out during a scheduled activity. It does not contain any other classes, but contains the parameters as described in table 3‑18.

Table 3‑18 : Class ServiceInfo Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| serviceType | The type of service that will be carried out during an activity.  It should be noted that:   * APA-AZ/EL indicates Antenna Pointing Angles, Azimuth/Elevation; * APA-X/Y indicates Antenna Pointing Angles, X/Y; * VLBI indicates Very-Long-Baseline Interferometry. | Enumeration  The permitted values are:   * APA-AZ/EL * APA-X/Y * DELTADOR * DOPPLER * OFFLINE-TM-RECORDING * OFFLINE-TM-PROVISION * RF-ONLY * RANGING * RESERVED * TBD * TELECOMMAND * TELEMETRY * TEST * UNUSED * VLBI | n/a |
| frequencyBand | The frequency band that will be used by the service. (See table 2-1 in reference [7] for a definition of the frequencies each band refers to.) If the frequency band is not relevant the value N/A (not applicable) shall be used. If the frequency band to be used is not defined in table 2-1 in reference [7] then the value OTHER shall be used. For use of the value ALL see the recommendations where use of the ServiceInfo class is used. | Enumeration  The permitted values are as specified in the Frequency Band Designator (FBD) columns of table 2-1 in reference [7] with the addition of the following:   * N/A * OTHER * ALL | n/a |

##### Class ServicePackageXRef

This class is optional and may be used to map the contents of the Simple Schedule to the appropriate SCCS SM service packages (see section 3.2.3). It does not contain any other classes, but contains the parameters as described in table 3‑19.

Table 3‑19 : Class ServicePackageXRef Parameters

| **Parameter** | **Description** | **Data Type** | **Data Units** |
| --- | --- | --- | --- |
| serviceAgreementRef | The Service Agreement to which the following servicePackageRef applies. | xsd:string | n/a |
| servicePackageRef | Cross-reference to the service package, where the aperture allocation is contained in the case that the aperture allocation was obtained via SCCS SM.  NOTE – Several ScheduledActivity instances may have the same servicePackageRef since one service package can contain a number of aperture allocations. | xsd:string | n/a |

### Usage Notes

This section provides notes on how the various aspects of service management should be used for Level-0.

#### Lifecycle

The lifecycle of the various data entities in SM Level-0 is as described below;

1. A SMURF request is submitted by the User CSSS. This needs to be of types NewOnlineSrvPkgReq. This contains the information that the Provider CSSS requires to process the request and try to find a suitable time where a service package can be generated that is compatible with the constraints specified in the request.
2. Once a User CSSS has submitted a NewOnlineSrvPkgReq, and before this has been processed by the Provider CSSS they may submit one of the following types of SMURF request;
   1. ReplaceOnlineSrvPkgReq – this request replaces the one that was previously submitted with a modified request. For SM Level-0 notification of the acceptance or otherwise of changes requested by a ReplaceOnlineSrvPkgReq is not directly supported by Service Management, but the result may appear in a Simple Schedule.
   2. DeleteSrvPkgReq – this request deletes the request that was previously submitted. For SM Level-0 notification of the acceptance or otherwise of changes requested by a DeleteSrvPkgReq is not directly supported by Service Management, but the result may appear in a Simple Schedule.
3. Once the Provider CSSS has processed a request (assuming the received request is syntactically correct) one of two things can happen, it can either be accepted, i.e. a service package can be generated that satisfies the constraints specified in the request or, if the constraints cannot be satisfied, it is rejected. In either case a Service Package data entity is generated;
   1. For the accepted case it is recommended (although not mandatory) that the servicePackageStatus parameter in the ServicePkg is set to “SCHEDULED” and the details of the generated service package are contained in the ServicePkgBody.
   2. For the rejected case what shall be returned is an empty ServicePkyBody with only the two parameters belonging to the ServicePkgBody specified, i.e. spaceUserNode and serviceReqRef.
4. Once a Service Package has been generated the User CSSS may submit additional SMURF requests to act on this. The two possibilities supported are;
   1. ReplaceSrvPkg – this may be used by the User CSSS to modify a Service Package generated by the Provider CSSS. For SM Level-0 it is possible to modify the following aspects of a Service Package by this method;

* Beginning of track
* End of track
* Configuration Profile

It should be noted that accepting these changes is at the discretion of the Provider CSSS and so they may not be implemented. For SM Level-0 notification of the acceptance or otherwise of changes requested by a ReplaceSrvPkg is not directly supported by Service Management, but the result may appear in a Simple Schedule.

* 1. DeleteSrvPkg – this may be used by the User CSSS to delete a Service package generated by the Provider CSSS. For SM Level-0 notification of the acceptance or otherwise of changes requested by a DeleteSrvPkg is not directly supported by Service Management, but the result may appear in a Simple Schedule.

1. The Simple Schedule Format (SSF) specifies a standard format for use in transferring scheduling information related to apertures at ground stations and/or relay satellites between space agencies and commercial or governmental spacecraft operators. It may be generated periodically by the Provider CSSS or on request. As SM Level-0 no mechanism is supported for the submission of a request to generate a Simple Schedule and the mechanism would need to be agreed by the participating entites. The Simple Schedule essentially reports for one or more Space Users Nodes (i.e spacecraft);
   1. Status of an assigned tracking pass.
   2. Aperture assigned at a site.
   3. Beginning of track.
   4. End of track.
   5. Services to be supported during the tracking pass.
   6. Other, optional, information may also be provided;

* Beginning of activity.
* End of activity.
* Orbit number.
* Service agreement covering the activity.
* Reference of the service package containing the activity.

The Simple Schedule may also contain information about unallocated time or time allocated to the Provider CSSS (e.g. for testing or maintenance etc.). Provision of such information is however dependent on the policies of the Provider CSSS.

1. Security, SANA, And Patent Considerations  
     
   (Informative)

As this recommended practise does not specify any new standards but rather describes how to use those specified in other CCSDS books there are no explicit security, SANA or patent considerations, instead the appropriate sections in the following references should be consulted for the relevant considerations;

* Ref . [4] Annex B Security, SANA and Security Considerations
* Ref . [5] Annex B Security, SANA and Security Considerations
* Ref . [6] Annex B Security, SANA and Security Considerations

1. XML Schema Organization And Packaging for THE Common Data Entity CLASSes   
     
   (InFormative)

As this recommended practise does not specify any new XML schemas but rather describes how to use those specified in other CCSDS books the appropriate sections in the following references should be consulted for the relevant information;

* Ref. [4] Annex C XML Schema Organization and Packaging For The Common Data Entity Classes.
* Ref. [5] Annex C XML Schema Organization and Packaging For The Common Data Entity Classes.
* Ref. [6] Annex C XML Schema Organization and Packaging For The Common Data Entity Classes.
  1. Examples

A number of XML examples illustrating the usage of the Service Management Levels described in this Recommended Practise, along with the required schemas, can be found in GitHub at the following URL:

[TBD](https://github.com/CCSDS-CSSM/902Releases/releases/tag/CESG_Review-20230823)

1. 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 |