

Draft Recommendation for  
Space Data System Standards

|  |
| --- |
| Cross Support Transfer Service— Service ConTROL |

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Issue: | , |  |
|  | Date: | November, 2024 |  |
|  | Location: | Not Applicable |  |
|  |  |  |  |

**(WHEN THIS RECOMMENDED STANDARD 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 e-mail address below.

This document is published and maintained by:

CCSDS Secretariat

National Aeronautics and Space Administration

Washington, DC, USA

E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

**(WHEN THIS RECOMMENDED STANDARD 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 **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.

o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:

-- The **standard** itself.

-- The anticipated date of initial operational capability.

-- The anticipated duration of operational service.

o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** 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 Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard 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 e-mail 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 Federal Science Policy Office (BFSPO)/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.
* Chinese 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.
* 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.
* 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.

1. United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its ‘Red 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 922.4-W-0.1 | | Service Control Cross Support Transfer Service | October 2022 | | First draft. | |
| CCSDS 922.4-W-0.2 | | Service Control Cross Support Transfer Service | March 2023 | | - Incorporates review comments Toulouse meeting  Rename Throw Event  Adding  organized annex | |
| CCSDS 922.4.-W-0.3 | | Service Control Cross Support Transfer Service | May 2023 | |  | |
| CCSDS 922.4-W-0.4 | | Service Control Cross Support Transfer Service | July 2023 | | Incorporated review comments from Spring Meeting | |
| CCSDS 922.4-W-0.5 | | Service Control Cross Support Transfer Service | April 2024 | | Updated Figure 2-1, 2-2. Incorporated review comments from telecon. Update references and Section 4. | |
| CCSDS 922.4-W-0.6 | | Service Control Cross Support Transfer Service | June 2024 | | Updated fig 2-1,2-2,2-3, added fig 4, incorporated additional comments from Spring meeting. | |
| CCSDS 922.4-W-0.7 | | Service Control Cross Support Transfer Service | November 2024 | | Updated Section 2.5 Operational Scenario and Appendix H | |
| CCSDS 922.4-W-0.8 | | Service Control Cross Support Transfer Service | November 2024 | | Updated Section 2.5 Operational Scenario. | |
|  |  | | |  | |  | |
|  |  | | |  | |  | |

CONTENTS

Section Page

[1 introduction 1-1](#_Toc181886922)

[1.1 Purpose of thIS REcommended Standard 1-1](#_Toc181886923)

[1.2 SCOPE 1-1](#_Toc181886924)

[1.3 APPLICABILITY 1-1](#_Toc181886925)

[1.4 rationale 1-1](#_Toc181886926)

[1.5 document strUcture 1-1](#_Toc181886927)

[1.6 definitions, nomenclature, and conventions 1-1](#_Toc181886928)

[1.7 references 1-1](#_Toc181886929)

[2 Overview OF THE Service Control Cross Support Transfer SERVICE 2-1](#_Toc181886930)

[2.1 Service Summary 2-1](#_Toc181886931)

[2.2 Functional Description 2-1](#_Toc181886932)

[2.3 Service ManagEment 2-1](#_Toc181886933)

[2.4 Cross Support View 2-1](#_Toc181886934)

[2.5 Operational scenario 2-1](#_Toc181886935)

[3 Service Control Cross Support Transfer Service Composition 3-1](#_Toc181886936)

[3.1 DISCUSSION 3-1](#_Toc181886937)

[3.2 Procedures of the Service Control Cross Support Transfer Service 3-1](#_Toc181886938)

[3.3 Service Control Cross Support Transfer Service State Machine 3-1](#_Toc181886939)

[4 MANAGED INFORMATION and FUNCTIONAL RESOURCE PARAMETERS AND EVENTS 4-1](#_Toc181886940)

[4.1 General 4-1](#_Toc181886941)

[4.2 common CSTS manageed information 4-1](#_Toc181886942)

[4.3 Association Control Procedure Managed Information 4-1](#_Toc181886943)

[4.4 Throw Event procedure managed information 4-1](#_Toc181886944)

[4.5 Published DIRECTIVE 4-1](#_Toc181886945)

[4.6 INVOKED DIRECTIVE reconfigurable Parameters 4-1](#_Toc181886946)

[ANNEX A Implementation Conformance Statement Proforma (Normative) A-1](#_Toc181885753)

[ANNEX B Service Object Identifiers Module (Normative) B-1](#_Toc181885754)

[ANNEX C Service CONTROL Procedure Parameters, Events, and Directives (Normative) C-1](#_Toc181885755)

[ANNEX D Security, SANA, and Patent Considerations (Informative) D-1](#_Toc181885756)

[ANNEX E Example Functional Resource Type Object Identifier Registry (INformative) E-1](#_Toc181885757)

[ANNEX F Informative references (INformative) F-1](#_Toc181885758)

[ANNEX G Acronyms (INFormative) G-1](#_Toc181885759)

Figure

[1-1 Cross Support Services Documentation 1-1](#_Toc181535031)

Table

[Table A‑1: Identification of PICS A-1](#_Toc178855628)

[Table A‑2: Identification of Implementation Under Test A-1](#_Toc178855629)

[Table A‑3: Identification of Supplier A-1](#_Toc178855630)

[Table A‑4: Identification of Specification A-1](#_Toc178855631)

[Table A‑5: Required Procedures A-1](#_Toc178855632)

[Table A‑6: Required PDUs A-1](#_Toc178855633)

[Table A‑7: BIND Invocation Parameters A-1](#_Toc178855634)

[Table A‑8: BIND Return Parameters A-1](#_Toc178855635)

[Table A‑9: PEER-ABORT Invocation Parameters A-1](#_Toc178855636)

[Table A‑10: UNBIND Invocation Parameters A-1](#_Toc178855637)

[Table A‑11: UNBIND Return Parameters A-1](#_Toc178855638)

[Table A‑12: EXECUTE-DIRECTIVE Invocation Parameters A-1](#_Toc178855639)

[Table A‑13: EXECUTE-DIRECTIVE Acknowledgement Parameters A-1](#_Toc178855640)

[Table A‑14: EXECUTE-DIRECTIVE Return Parameters A-1](#_Toc178855641)

# introduction

## Purpose of thIS REcommended Standard

This Recommended Standard defines the Service Control Cross Support Transfer Service (SC-CSTS), in conformance with the Cross Support Transfer Services Specification Framework Recommended Standard (reference [1]). The SC-CSTS is a transfer service that allows a Mission Data Operations System (MDOS) to request that actions be invoked on, or parameter values be reconfigured for, cross support services being provided by a Cross Support Complex. The types of Service Control accessible by means of the SC-CSTS are identified in 0.

NOTE – The term ‘Cross Support Complex’, as used throughout this document, corresponds to the Earth Space Link Terminal (ESLT) defined in the Space Communications Cross Support Architecture Description Document (reference[[8](#R_901x1m1SpaceCommunicationsCSADD)].)

## SCOPE

This Recommended Standard defines the Service Control service using procedures and operations defined in the *Cross Support Transfer Service Specification Framework* (reference [1]) and in accordance with the *Guidelines for the Specifications of Cross Support Transfer Services* (reference [[7](#R_901x1m1GuidelinesCSTS)]).

### This Recommended Standard defines the SC-CSTS in terms of:

1. the CSTS Specification Framework (SFW) procedures that comprise the Service Control service. The SFW procedures used in any implementation of this issue of this Recommended Standard must conform to the selected procedures as specified in issue-2 of the SFW as specified in reference [1]. Future Issues of this Recommended Standard (if any) will reference and conform to the issue of the SFW that is in effect at that date;
2. the relationships among the procedures that constitute the service: and
3. the requirements on Service Control service to enable the proper operation of the SC-CSTS.

### This Recommended Standard does not specify:

1. individual implementations or products;
2. the implementation of entities or interfaces within real systems;
3. the methods or technologies required to measure the values of service control parameters and to detect the occurrence of events of interests;
4. the methods or technologies required for communication;
5. the management activities necessary to schedule, configure, and control the SC-CSTS.
6. the specific parameters and events that are to be notified by the SC-CSTS.
7. SC-CSTS does not extend or refine any CSTS SFW procedures to provide the transfer service;
8. SC-CSTS does not extend or refined any of the standard CSTS operations associated with each of the procedures;

## APPLICABILITY

The applicability and limits of applicability of Cross Support Transfer Services in general, as described in reference [1], pertain to the Service Control service.

This Recommended Standard is applicable to the implementation of real systems that monitor provision and production of space communication Cross Support Services for the purposes of generating cyclic status reports, generating notifications of changes in status in real time, and responding to queries of current values of operational parameters.

## rationale

The goal of this Recommended Standard is to create a standard for interoperability for the exchange of cross support service-related status information between the cross support elements of various space Agencies and the users of the Cross Support Services that they provide.

## document strUcture

### document organization

Section 2 describes the Service Control Cross Support Transfer service in terms of:

* the role of Service Management with respect to the SC-CSTS;
* the allocation of production and provision of the SC-CSTS to Functional Resources;
* the cross support view of the SC-CSTS;
* the functional description of the production and provision of the service; and
* an operational scenario that illustrates some of the more significant aspects of the service.

Section 3 specifies the top-level composition of the SC-CSTS. The service type identifier is declared, the procedures that make up the service are identified, and the CSTS state machine that applies to the SC-CSTS is specified. Because the SC-CSTS is composed of procedures that are directly adopted from the CSTS Framework without extension, no further specification of the MD-CSTS is required.

The SC-CSTS incorporates a procedure (Throw Event) is adopted from the Throw Event procedure from the CSTS Framework.

Section 4 specifies the managed information that is exchanged via service management in order to configure the SC-CSTS.

Annex A documents the Implementation Conformance Statement (ICS) Proforma for the SC-CSTS.

Annex B formally specifies the Object Identifiers (OIDs) for the Service Control transfer service.

Annex C formally specifies the PROCEDURE Parameter, Events, and Directives for the Service Control service.

Annex D addresses the security, Space Assigned Numbers Authority (SANA), and patent considerations associated with the SC-CSTS.

Annex E lists the example functional resource type object identifier registry in this Recommended Standard.

Annex F lists the informative references cited in this Recommended Standard.

Annex G lists the acronyms used in this Recommended Standard.

Annex H lists the cross references to Cross Support Transfer Service Specific Framework in this Recommended Standard.

### cross support Transfer services documentation

The basic organization of the CSTS documentation and the relationship to CSTS documentation is shown in Figure 1‑1.



Figure ‑ : Cross Support Services Documentation

The Cross Support Architecture is documented in

1. *Space Communications Cross Support—Architecture Description Document* ([F8](#R_901x1m1SpaceCommunicationsCSADD)): An Informational Report describing an architecture in terms of CCSDS-recommended configurations for secure space communications cross support. This architecture is intended to be used as a common framework when CCSDS Agencies 1) provide and use space communications Cross Support Services and 2) develop systems that provide interoperable space communications cross support.
2. *Space Communications Cross Support—Architecture Requirements Document* (F6): A Recommended Practice defining a set of requirements for CCSDS-recommended configurations for secure space communications cross support architectures.

Common to all Cross Support Services

1. *Space Communication Cross Support Service Management suite* (references [F6]). Data format Recommended Standards that specify the Service Management Information Entities that are used to configure and schedule cross support services, which include transfer services.

Common to the Transfer Services, that is, SLE Transfer Services and CSTSes

1. *Space Link Extension—Internet Protocol for Transfer Services* (reference [F2]): A Recommended Standard that defines a protocol for transfer of Protocol Data Units (PDUs) defined in the CSTSes. This Recommended Standard was originally developed to support SLE transfer services (hence the title), but it is also applicable to CSTSes.

The concept, the reference model, and the SLE Transfer Services are documented in

1. *Cross Support Concept—Part 1: Space Link Extension Services* (reference [F1]): A report introducing the concepts of cross support and the SLE services. Many of the concepts for the SLE transfer services have been adopted for the CSTSes (see k) below).
2. *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [2]): A Recommended Standard that defines the framework and terminology for the specification of SLE services. Much of the framework and terminology of this reference model has been adopted or adapted for CSTSes (see 1.6.1.3 and 2.2 of reference [1]).
3. The *SLE Transfer Services suite*: The SLE Transfer Services are a suite of Cross Support Services that are used to transfer specific TC and TM PDUs. The SLE Transfer Services are closely related to the CSTS suite in that they collectively define the set of operations that are the basis for the CSTS SFW. However, because of their history (the SLE Transfer Services were already specified and implemented prior to development of the CSTS SFW), the SLE Transfer Services are separated from CSTSes.

The documents specific to CSTSes are

1. *Cross Support Transfer Services Specification Framework* (reference [1]): A Recommended Standard that defines the specification of the CSTS procedures;
2. *Guidelines for Specification of Cross Support Transfer Services* (reference[7]): A Recommended Practice that defines the guidelines for construction of a CSTS based on the CSTS SFW;
3. *Cross Support Transfer Services Concept* (reference [F3]): A Report that provides tutorial material on the objectives and concepts of the CSTS specification; and
4. *Cross Support Transfer Services Suite*: The set of specifications for actual CSTSes built from the procedures in the CSTS SFW and in accordance with the CSTS Guidelines. This Recommended Standard is a member of this suite.

## definitions, nomenclature, and conventions

### Terms

#### Terms Defined in the Cross Support Transfer Services Specification Framework Recommended Standard (reference [1])

1. Association Control procedure;
2. Cross Support Complex;
3. Cross Support Service production;
4. Cross Support Transfer Service;
5. Functional Resource instance;
6. Functional Resource Instance Number;
7. Functional Resource Name;
8. Functional Resource Type;
9. non-blocking (operation);
10. prime procedure instance;
11. procedure configuration parameter;
12. procedure type;
13. procedure instance identifier;
14. Published Identifier;
15. qualified parameter;
16. secondary procedure instance;
17. service management parameter;
18. service-user-responding-timer;
19. subscription.
20. Directive Identifier
21. Directive Name

#### Terms Defined in the Cross Support Reference Model (reference[2])

1. Complex Management (CM) (called *SLE Complex Management* in reference [2]);
2. Mission User Entity (MUE);
3. service agreement (called *SLE* *Service Agreement* in reference [2]);
4. service package (called *SLE* *Service Package* in reference [2]);
5. space link session;
6. transfer service production;
7. transfer service provision;
8. Utilization Management (UM);
9. utilization phase (called SLE Service Package Utilization phase in reference [2]).

### NOMENCLATURE

#### Normative Text

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

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 set off from the normative specifications either in notes or under one of the following subsection headings:

* Overview;
* Background;
* Rationale;
* Discussion.

### conventions

#### Overview

The conventions defined in the CSTS Specification Framework Recommended Standard (reference [1]) are applicable to this Service Control service specification, with the exception of the representation of Object Identifiers. The conventions for the representation of Object Identifiers in this Recommended Standard are described in 1.6.3.2.

#### Object Identifier Representation

##### General

The SC-CSTS service involves extensive use of Functional Resource Types, procedure types, Functional Resource Names, procedure instance identifiers, Parameter Names, Parameter Identifiers, Event Names, and Event Labels. As specified in reference [1], all of these names are based on Published Identifiers, which are International Organization for Standardization (ISO) Object Identifiers (OIDs). OIDs have the syntax of strings of integers. For purposes of readability, rather than using actual OIDs in the descriptions and examples in this Recommended Standard, the OIDs in these names and identifiers are represented using the following textual notation to represent the OIDs.

##### Functional Resource Type

As specified in reference [1], a Functional Resource Type is A logical function or related set of functions that characterizes a unique instance of service provider or production capability. A Functional Resource Type is defined as a Published Identifier.

##### Procedure Type

As specified in reference [1], a procedure type is an ISO OID that is assigned to a CSTS procedure. In the descriptions and examples in this Recommended Standard, a procedure type is represented using the notation *{object identifier classifier}*, which is a textual description of the Object Identifier. Thus {Throw Event} represents the Object Identifier for the Throw Event procedure type.

##### Functional Resource Name

As specified in reference [1], a Functional Resource Name is the unique identifier of an instance of a Functional Resource within the scope of a given service package. A Functional Resource Name is made of a Functional Resource Type and a Functional Resource Instance Number. A Functional Resource Name is represented using the notation *[{functional resource type published identifier classifier}: FRIN]*.[{Antenna}: 1] represents the name of the Antenna Functional Resource Type that is assigned FRIN = 1 in the scheduled service package.

##### Directive Identifier

As specified in reference [1], a Directive Identifier is the unique identifier of a directive defined for service provision or service production. A Directive Identifier is defined as a Published Identifier.

##### Directive Name

As specified in reference [1], a Directive Name is A data structure consisting of a Directive Identifier that represents an individual directive type, and a Functional Resource Name that represents the Functional Resource with which the specific instance of that directive type is associated.

##### Procedure Instance Identifier

As specified in reference [1], a procedure instance identifier identifies the specific instance of a procedure in a CSTS. It is composed of a procedure type Published Identifier and a procedure role, which can have one of three values: ‘association control’ if it is the Association Control procedure of the CSTS; ‘prime procedure instance’ if it is the prime instance of the procedure of the CSTS; or a positive integer Secondary Procedure Instance Number (SPIN) if it is a secondary procedure instance of the CSTS.

In the descriptions and examples in this Recommended Standard, a procedure instance identifier for an Association Control procedure is represented using the notation *[{procedure type object identifier classifier}: ‘association control’]*.[{Association Control}: ‘association control’] represents the Procedure Instance Identifier of the Association Control instance of a CSTS.

NOTE – In the case of the Framework Association Control procedure, the Procedure Role appears to be redundant with the procedure type. However, it is possible for a CSTS to extend the Association Control procedure, in which case it would get a separate procedure type, in which case the Procedure Role explicitly identifies the OID as that belonging to an Association Control procedure.

A procedure instance identifier for a prime procedure instance is represented using the notation *[{procedure type object identifier classifier}: ‘prime procedure instance’]*.[{Cyclic Report}: ‘prime procedure instance’] represents the procedure instance identifier of the Cyclic Report instance that serves as the prime procedure instance of a CSTS.

A procedure instance identifier for a secondary procedure instance is represented using the notation *[{procedure type object identifier classifier}: SPIN]*.[{Notification}: 1] represents the procedure instance identifier of the Notification instance that serves as the first Notification secondary procedure instance of a CSTS.

## 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.

[] *Cross Support Transfer Service—Specification Framework*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 921.1-B-2. Washington, D.C.: CCSDS, February 2021.

[] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 910.4-B-2. Washington, D.C.: CCSDS, October 2005.

[] *Service Control Message*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 503.0-B-1. Washington, D.C.: CCSDS, November 2007.

[] *Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. 5th ed. International Standard, ISO/IEC 8825-1:2015. Geneva: ISO, 2015.

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

[] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

[7] *Guidelines for the Specification of Cross Support Transfer Services*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 921.2‑M‑1. Washington, D.C.: CCSDS, February 2021.

[8] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.

# Overview OF THE Service Control Cross Support Transfer SERVICE

## Service Summary

The Service Control CSTS is a CCSDS Cross Support Transfer Service that provides a user with the capability to:

a) Request that a control action be performed on a space communication cross support service during the utilization phase of a cross support service package;

b) Request the change of values of one or more configuration managed parameters of a space communication cross support service during the utilization phase of a cross support service package.

An instance of the Service Control service is realized through the following activities:

a) The user binds to the provider to establish a service association;

b) The user requests that a real-time control action be performed, or the user requests that one or more configuration parameters be given new values (as specified in the request);

c) The service provider validates the request and acknowledges its validity to the user (or rejects the request if it is invalid);

e) The service provider attempts to perform the action or reconfigure the parameter values.

1) If a control action is requested, the service provider reports whether the requested action has been successfully performed or has failed.

2) If multiple parameters are to be reconfigured, the service provider changes the parameter values if and only if all requested parameter value changes can be made successfully. The service provider reports whether the requested parameter changes have been successfully performed or failed.

f) The user unbinds from the provider to release the association.

## Functional Description

### General

As defined in the Cross Support Reference Model: Part 1 [[2](#R_910x4b2CrossSupportReferenceModelPart1)], related cross support services are bundled into Service Packages for the purpose of ensuring that the required relationships among those cross support services are consistent duri their production and provision. For example, multiple cross support transfer services may be related to the operation of the same RF link, and the return RF link may be related to the forward RF link - all of those transfer services, as well as the RF links themselves, comprise a single Service Package for the purposes of scheduling. The content and structure of Service Packages are fully defined in reference [[4](#R_902x9wCrossSupportSmSmUtilizationReque)].

A Service Package may contain a single instance of the Service Control service. The single instance of the Service Control service in a Service Package is capable of invoking all control actions and real-time reconfigurations of all reconfigurable parameters for all Functional Resource Instances representing production processes for that Service Package. For the example of a Service Package comprising a forward RF link, a return RF link, multiple SLE transfer service instances, and an instance of the Service Control service, the Service Control service instance has access to the control actions and reconfigurable parameters for those forward and return RF links.

In accordance with reference [3], the functionality associated with a transfer service is partitioned into production and provision of the service. The following subsections describe the production and provision of the Service Control service, respectively.

### Service Production

The production of the Service Control service associated with a given Service Package consists of:

* instances of *controllable functional resources* that realize that Service Package; and
* a Control Distribution function that translates the control actions and parameter reconfigurations contained in the requests into the equipment-specific commands of the equipment represented by the functional resource identifiers associated with the control actions and parameter reconfigurations. For multiple parameter reconfigurations, the Control Distribution function also determines whether all parameters can be successfully reconfigured before reconfiguring any of them.

NOTE - The Control Distribution function may be part of Complex Management, but it is described here as a separate production function to emphasize the required functionality,

Figure 2-1 is a notional representation of the relationship between the functional resources of a Service Package and an instance of Service Control service that reports on that Service Package.



Figure 2‑1: Production and Provision of Service Control Services (Notional)

A *functional resource* (FR) refers to a logical function or related set of functions that provides space communication capability (see [[5](#R_FunctionalResourcesSpaceAssignedNumber)]), and a *controllable* functional resource is a functional resource for which reconfigurable parameters and/or control actions are specified. A functional resource parameter of a CSTS is a parameter that is accessible through the functional resource that represents that CSTS. A functional resource parameter is named using the Functional Resource Type and functional resource parameter registered in the SANA Functional Resource Registry (reference [[5]](#R_FunctionalResourcesSpaceAssignedNumber)).

Functional Resources are not the physical resources (e.g., transmitters and receivers) that comprise real systems. Rather, they are abstract representations of the functions or capabilities that are provided by those physical resources. A Functional Resource may be realized by several physical entities that work cooperatively to perform that function. Alternatively, for some types of functional resources, a single physical resource may be designed such that it instantiates several functional resources.

Fundamental to the concept of Functional Resources is that each one represents a cohesive, atomic set of space communication functionality with which can be associated to single instances of management parameters, monitored parameters, real-time control parameters, and event notifications.

The execution of a typical Service Package is realized through the operation of instances of different types of functional resources (e.g., forward space link, return space link subcarrier). Conceptually, instrumentation of the functional resources provide real-time access to some or all of the configuration parameters of those resources, and allow control actions (e.g., sweep the forward link carrier) to be invoked upon them.

The SC-CSTS can reconfigure parameter values and invoke control actions for any set of functional resources as long as the new parameter values and control actions can be expressed in accordance of this specification.

The Control Distribution function of SC-CSTS production makes all reconfigurable parameters and control actions accessible available to the instance of SC-CSTS for the Service Package in conformance with the naming, syntax, type, and units defined in the specification for those parameters and actions. In cases where the real implementations of controllable functional resources do not present their reconfigurable parameters and control inputs in conformance with the naming, syntax, and/or units required for transfer via the SC-CSTS, the Control Distribution function performs the appropriate format conversion. Also, in cases where real implementations of functional resources do not directly implement control actions of functional resource types, the Control Distribution function may synthesize control actions from lower-level control actions and/or manipulation of configuration parameter values.

Upon validation of a control directive from the SC-CSTS service instance, the Control Distribution function checks and enforces any guard conditions appropriate to that directive prior to attempting to perform the operation. These guard conditions ensure that the directed action is appropriate to the current state of the affected controllable functional resources. The Control Distribution function also confirms that all parameters listed in a ‘reconfigure parameters’ directive are capable of being reconfigured to the directed values *before any* parameter is reconfigured. If no guard conditions are violated (and if all parameters directed for reconfiguration are determined to be reconfigurable), the Control Distribution function performs the directive and confirms successful performance to the SC-CSTS service instance. Otherwise, the Control Distribution function does not perform the directive and indicates a negative result to the SC-CSTS service instance.

NOTE - For the purposes of this description, the enforcement of the guard conditions and pre-checking of all parameter reconfigurations are ascribed to the Control Distribution function. Real service providers (Complexes) may distribute these capabilities differently within the overall production of the Service Package.

### Service Provision

A single instance of the SC-CSTS may exist in a Service Package, with that instance having access to all reconfigurable parameters and control actions published for that mission.

The SC-CSTS allows a single instance of the Throw Event procedure to be activated during the service instance provision period (see reference [[1](#R_921x1b1CstsSpecificationFramework)]) of the SC-CSTS instance.

The instance of the Service Control service allows the user to specify the directive qualifier that carries the configuration parameters (up to a maximum number that is controlled via Service Management) or a single control action in each EXECUTE-DIRECTIVE invocation. For each parameter that is to be reconfigured, the fully-qualified parameter name and *typed value* (see[1.6.3.2.6](#OLE_LINK3)) are specified in the EXECUTE-DIRECTIVE invocation. The directive qualifiers contain the configuration parameter which is a sequence of parameter identifier and parameter value pairs.

In performing the EXECUTE-DIRECTIVE operation of the SC-CSTS, the service instance validates the invocation, which includes confirming that the fully-qualified parameter names or action name contained in the invocation is in the set of published parameter names or published action names, respectively; that an action has the qualifiers that are required for the action type, and that the typed values of the parameters or action qualifiers are as specified for their respective parameters or action qualifiers. Upon detection of a condition that invalidates the invocation, the SC-CSTS service instance returns a negative acknowledgement that identifies the reason (diagnostic) for the invalid condition. If the reason is an unknown or an incorrectly-specified reconfiguration parameter, the service instance also lists the fully-qualified names of all reconfiguration parameters that are invalid for that same reason.

If a directed action is invalid, the service instance returns a negative acknowledgement listing the reason (diagnostic) that the invocation is not valid.

Following validation of the execute-directive invocation, the service instance forwards the directive to the control distribution production to attempt to reconfigure the parameters or perform the action. Upon indication from service production that the directive has been successfully performed, the service instance returns a positive-result return. Otherwise, the service instance returns a negative-result return with appropriate diagnostic information service management

As defined in the Cross Support Reference Model: Part 1 [[3](#R_910x4b2CrossSupportReferenceModelPart1)], related cross support services are bundled into Service Packages for the purposes of scheduling. Cross support service management both establishes the constraints on the Service Packages to which a given spaceflight mission must conform (e.g., data rate and frequency ranges, types and numbers of cross support transfer service instances) and provides the mechanisms for instantiating conformant Service Packages (e.g., via scheduling).

Regarding the production and provision of Service Control service instances, cross support service management:

a) Establishes the set of configuration parameters that may be reconfigured during the utilization phase of a service package. The set of reconfigurable parameters that a given service provider makes available is referred to as the set of *published* reconfigurable parameters. The published reconfigurable parameters are logically a part of the Service Agreement between the spaceflight mission and the Complex; and

b) Establishes the set of control actions that may be invoked during the utilization phase of a service package. The set of control actions that a given service provider makes available is referred to as the set of published actions. The published actions are logically a part of the Service Agreement between the spaceflight mission and the Complex.

The means by which service management develops these published sets is outside the scope of this Recommendation.

## Service ManagEment

Cross support service management both establishes the constraints on the Service Packages to which a given spaceflight mission conforms (e.g., data rate and frequency ranges, types and numbers of cross support transfer service instances) and provides the mechanisms for instantiating conformant Service Packages (e.g., via scheduling).

Regarding the production of Service Control and provision of Service Control CSTS instances, cross support service management:

1. schedules the Service Packages that specify the tracking activities that are to be performed and the SC-CSTS instance that transfers the Service Control to the users of those service instances; and
2. establishes the types of Service Control that can be exercised by the SC-CSTS instance during the execution and reporting of a Service Package.

A Service Package identifies the various space communication and radiometric functions that are to be performed by a Cross Support Complex during a specified period of time. Typically, the Service Package corresponds to the functions performed at a single ground station for one pass/contact/track. More sophisticated cases involving more than one antenna can also be supported. One possibility is that the TDM can contain inputs from different ground stations.

The Service Package also defines – indirectly, through reference to configuration profiles – the configuration parameters that specify the initial configurations of the space communication and radiometric functions, and the interrelationships among them (e.g., the frame length on each return link symbol stream). With respect to the Service Control Service, the Service Package identifies which resources perform the functions that produce the radiometric measurements that are reported by the Service Control service, and pairs those resources to the TDM keyword identifiers (refer to reference [3]) used in the generation of the atomic segments.

The Service Management will schedule the time period when real-time directives are permitted by the SC service instance. This SM provision will ensure SC service be coordinated with scheduled directives.

The means by which service management schedules the Service Packages is outside the scope of this Recommendation. The CCSDS SM (references [F7] and [F8]) defines a standard set of service management information entities used in the scheduling of Service Packages.

## Cross Support View

Figure 2-2 shows an example configuration of a Cross Support Complex providing a Service Control provider instance to a Service Control user. In this example, several production functional resource instances are configured to support the Service Package (and are available for control via the SC-CSTS). The associated production functional resource instances are not *part of* the Service Control service, but they are *controlled through* the Service Control service.

As shown in Figure 2-2, a single Service Package may include a single instance of the Service Control service, which is in turn configured to access all or a subset reconfigurable parameters and control actions of functional resource instances associated to the Service Package.



Figure 2‑2: Example of the Servi­ce Control Service Instance for a Service Package

NOTE - According to the Cross Support Reference Model [[3](#R_910x4b2CrossSupportReferenceModelPart1)], the responsibility for controlling the execution of a Service Package is associated with the Utilization Management (UM) role of the MDOS, which is a Service Management role. In this case, a Service Management function is being performed, but it is being performed using a CSTS rather than using a management service built upon the SCCS-SM infrastructure [4]. Although figure 2-2 shows the Mission User Entity (MUE) for the SC-CSTS as being separate from the UM role, it should be understood that in the nominal case the use of the SC-CSTS will be in support of a Service Management responsibility.

## Operational scenario

As part of the Service Control activities that establish the relationship between the mission user and the cross support complex, Utilization Management (UM) for the user and Complex Management (CM) negotiate the set of cross support services that will be available to the mission user within the context of the Service Agreement and each Service Package.

During the execution of a Service Package, the SC-CSTS user establishes an association with the SC-CSTS Provider by means of the BIND operation. Within an established association the SC-CSTS user can update reconfigurable parameters using EXECUTE-DIRECTIVE operation. The Functional Resource Model (FRM) specifies for each Functional Resource type the available configuration parameters. More precisely, the FRM specifies the parameter OID and the type definition of the value. The FRM and the contained Functional resources is defined in the SANA Functional Registry (see SANA [[5]](#R_FunctionalResourcesSpaceAssignedNumber)). The following example illustrates how a SC-CSTS user establishes an association and sets the symbol rate for the forward link to a new value. Once the execution of the Service Package is completed, the SC-CSTS user releases the association to the SC-CSTS Provider with an UNBIND operation.

Fig 2-3 shows the high level overview of the above SC-CSTS scenario, Figure 2-4 details the scenario further and illustrates how an reconfigurable parameter value is updated to a new value.



Figure 2- 3 Example of the Servi­ce Control Service Instance for a Service Package

The following steps illustrate a typical sequence of operations between the SC-CSTS User and the SC-CSTS Provider.

1. The user invokes the BIND operation of the Association Control procedure to establish an association with the provider.
2. The user invokes EXECUTE-DIRECTIVE operations and monitors the acknowledgement and return operations.
3. Repeat b) until the service is completed.
4. The user invokes the UNBIND operation at the end of the service.

To illustrate how to change a concrete parameter, we assume the ground station provider is providing a forward service and the SC-CSTS User wants to change the symbol rate on the forward link by means of the SC-CSTS Provider.



Figure 2-4 Sample Operational Scenario to change the symbol rate on the forward link to 80 ksps

Figure 2-4 shows the relevant ExecuteDirectiveInvocation operation and the corresponding acknowledgement and return operations. The change of the symbol rate is achieved by setting ccsds401CarrierXmitSymbolStreamModType parameter of the Functional Resource Instance Ccsds401SpaceLinkCarrierXmit. As shown in Figure 2-4, ccsds401CarrierXmitSymbolStreamModType is a complex parameter, the change of the symbol rate is achieved by setting the below element to the desired value:

ccsds401CarrierXmitSymbolStreamModType

symbolRateDopplerCompensation

modType

bpsk

bpskSymbolRateAndPcmFormat

ccsds

**symbolRate**

When encoding a complex parameter for the EXECUTE\_DIRECTIVE operation, all elements of the complex parameter must be provided. It is not sufficient to encode only the changed value.

# Service Control Cross Support Transfer Service Composition

## DISCUSSION

The service-level OIDs for the Service Control CSTS are specified in ANNEX C.

## Procedures of the Service Control Cross Support Transfer Service

The Service Control CSTS service shall be composed of the procedures identified in Table 3-1.

There shall be one and only one instance of the Association Control procedure.

The Association Control procedure shall be adopted directly from reference [[1](#R_921x1b1CstsSpecificationFramework)].

The version number of the Association Control procedure shall be ‘2’, indicating that the Association Control procedure specified in reference [[1](#R_921x1b1CstsSpecificationFramework)] shall be used without substitution of any other version of the Association Control procedure.

The Throw Event procedure shall be adopted directly from reference [[1](#R_921x1b1CstsSpecificationFramework" \o "Cross Support Transfer Service—Specification Framework)].

NOTE - The elements of table 3-1 are defined in the Procedures Identification subsection of the CSTS Guidelines [2].

Table 3-1: Service Control Transfer Service Procedures

|  |  |  |
| --- | --- | --- |
| Procedure | Association Control | Throw Event [P] |
| Version | SFW 2 | 2 |
| Number of Instances | 1..1 | 1.. \* |
| Specification Approach | adopted | adopted |
| Source | CSTSFW [[1](#R_921x1b1CstsSpecificationFramework)]: Association Control | CSTSFW [[1](#R_921x1b1CstsSpecificationFramework)]: Throw Event |

The Service Control Cross Support Transfer Service uses the Association Control procedure of reference [1] without derivation or refinement.

The Service Control Cross Support Transfer Service uses the Throw Event procedure, which is adopted and extended from the Throw Event procedure of reference [[1](#R_921x1b1CstsSpecificationFramework)].

## Service Control Cross Support Transfer Service State Machine

The Service Control Cross Support Transfer Service state machine conforms to the state machine for a CSTS with a stateful prime procedure instance, as defined in Annex G3 of reference [1].

# MANAGED INFORMATION and FUNCTIONAL RESOURCE PARAMETERS AND EVENTS

## General

The CSTS Specification Framework (reference [1]) specifies that certain parameters and events are specified for the functional resource that represents each CSTS Provider. The configuration and operation of an instance of the Service Control service requires that the managed information specified in this section be established at the Service Provider prior to the start of the service instance provision period.

## common CSTS manageed information

The managed information SC-CSTS shall include the managed information common to all CSTSes identified in the Interactions with Management annex of reference [[1](#R_921x1b1CstsSpecificationFramework)].

## Association Control Procedure Managed Information

The managed information for the Association Control procedure of the SC-CSTS shall consist of the managed information specified for the Framework Association Control procedure in reference [[1](#R_921x1b1CstsSpecificationFramework)].

NOTE - The managed information specified for the Framework Association Control procedure consists of the service-user-responding-timer.

## Throw Event procedure managed information

The managed information for the Throw Event procedure shall include the guard conditions (if any) that apply to the reconfiguration of the parameters contained in the set of published reconfigurable parameters (see [4.6](#_Published_reconfigurable_Parameters) ).

The managed information for the Throw Event procedure shall include the guard conditions (if any) that apply to the performance of actions contained in the set of published actions (see [4.5](#_Published_Actions)).

NOTE - The Framework Throw Event procedure specified in reference [[1](#R_921x1b1CstsSpecificationFramework)] states that the guard conditions are specified by the service using the procedure. The Service Control service defers the specification of the guard conditions to the Provider that is performing the service.

## Published DIRECTIVE

SC-CSTS service shall establish the set of invoke directive execution that are to be made available to be requested by an instance of the Throw Event procedure of the Service Control service.

The set of published directive executions shall include:

1. invoke-id, directive OID, qualifiers, and result
2. (list of invoked directives previously)

## INVOKED DIRECTIVE reconfigurable Parameters

SC-CSTS service management shall establish the set of invoked directive reconfigurable parameters that are to be made available to be requested by an instance of the Throw Event procedure of the Service Control service.

The set of invoked directive reconfigurable parameters shall include:

1. The name of the parameter; and
2. The type and range of the parameter value.

Each invoked directive reconfigurable parameter name shall be a fully qualified name consisting of two components: (1) the *functional resource identifier* and (2) the *parameter type*.

1. Implementation Conformance Statement Proforma  
     
   (Normative)
   1. INTRODUCTION
      1. OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of the *Cross Support Transfer Services – Service Control Service*, CCSDS 922.4-W-0.7. CCSDS 922.4-W-0.7 specifies the requirements on the provider of the Service Control Cross Support Transfer Service.

The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements referenced in the RL.

The RL support column in this annex is blank. An implementation’s completed RL is called the PICS. The PICS states which capabilities and options have been implemented. The following can use the PICS:

1. the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
2. a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
3. a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);
4. a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.
   * 1. ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

Item Column

The item column contains a prefix identifying the element the given table is referring to and sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means ‘Is this feature supported by the implementation?’

Status Column

The status column uses the following notations:

|  |  |  |
| --- | --- | --- |
|  | M | mandatory; |
|  | O | optional; |
|  | O.<n> | optional, but support of at least one of the group of options labeled by the same numeral <n> is required; |
|  | C<n> | conditional as defined in corresponding expression below the table; |
|  | X | prohibited; |
|  | N/A | not applicable. |

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

1. Y Yes, supported by the implementation;
2. N No, not supported by the implementation;
3. N/A Not applicable.

The support column should also be used, when appropriate, to enter values supported for a given capability.

Allowed Values Column

All PDU parameter types are specified in annex F of reference [1] or in this Recommended Standard using ASN.1. The ASN.1 data type specification constrains among others the permissible value range and therefore such constraints are not repeated in the Allowed Values column in the tables contained in this ICS annex. However, if a parameter is constrained for all instances of the given PDU to a subset of the range or set specified for that parameter type, then the subset is identified in the tables that contain PDU parameters.

Allowed Values Column Symbols

If the allowed values are too large to fit in the Allowed Values cell, the Allowed Values column uses the notation ‘AV<n>’ as an indication that the allowed values are specified below the table.

Supported Values Column

The Supported Values column is to be used by the implementer to state whether the specified range or set of values for the parameter is supported by entering Y or SV<n>, indicating:

|  |  |  |
| --- | --- | --- |
|  | Y | Yes, the range/set defined in the Recommended Standard is fully supported by the implementation; |
|  | SV<n> | The range/set defined in the Recommended Standard is not fully supported by the implementation. The supported subset is documented below the table. |

* + 1. INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called a PICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

* 1. PICS PROFORMA FOR the Service Control CSTS PROTOCOL ()
     1. GENERAL INFORMATION

The PICS for a SC-CSTS implementation shall encompass the filled in tables A‑1 to A‑4.

Table A‑ : Identification of PICS

|  |  |
| --- | --- |
| Date of Statement (DD/MM/YYYY) |  |
| PICS serial number |  |
| System Conformance statement cross-reference |  |

Table A‑ : Identification of Implementation Under Test

|  |  |
| --- | --- |
| Implementation name |  |
| Implementation version |  |
| Special Configuration |  |
| Other Information |  |

Table A‑ : Identification of Supplier

|  |  |
| --- | --- |
| Supplier |  |
| Contact Point for Queries |  |
| Implementation Name(s) and Versions |  |
| Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems;  System Name(s) |  |

Table A‑ : Identification of Specification

|  |  |
| --- | --- |
| CCSDS 922x4-w-0.5 | |
| Have any exceptions been required?  NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is nonconforming. | Yes [ ] No [ ] |

* + 1. REQUIREMENTS LIST

This subsection provides the Requirement Lists for the elements specified in this Recommended Standard.

Table A‑ : Required Procedures

| Procedures | | | | |
| --- | --- | --- | --- | --- |
| Item | Description | Reference | Status | Support |
|  | Association Control | Subsection 4.3 of reference [1] | M |  |
|  | Throw Event | Subsection 4.12 of reference [1] | M |  |

The Throw Event procedure is mandatory.

Table A‑ : Required PDUs

| Item | PDU | Reference | Service-Provider-System | | Service-User-System | |
| --- | --- | --- | --- | --- | --- | --- |
| Status | Support | Status | Support |
|  | BindInvocation | F3.5 of reference [1] | M |  | M |  |
|  | BindReturn | F3.5 of reference [1] | M |  | M |  |
|  | PeerAbortInvocation | F3.5 of reference [1] | M |  | M |  |
|  | UnbindInvocation | F3.5 of reference [1] | M |  | M |  |
|  | UnbindReturn | F3.5 of reference [1] | M |  | M |  |
|  | ExecuteDirectiveAcknowledge | F3.4 of reference [1] | C1 |  | C1 |  |
|  | ExecuteDirectiveInvocation | F3.4 of reference [1] | C1 |  | C1 |  |
|  | ExecuteDirectiveReturn | F3.4 of reference [1] | C1 |  | C1 |  |

|  |  |
| --- | --- |
|  | IF proc-9 OR proc-11 THEN M ELSE N/A |
|  | IF proc-5 THEN M ELSE N/A |
|  | IF proc-9 THEN M ELSE N/A |
|  | IF proc-6 THEN M ELSE N/A |
|  | IF proc-10 THEN M ELSE N/A |

Table A‑ : BIND Invocation Parameters

| Parameters of the BindInvocation PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Reference | Status | Support | Values | |
| Allowed | Supported |
|  | invokerCredentials | F3.3 of reference [1] | M |  |  |  |
|  | invokeId | F3.3 of reference [1] | M |  |  |  |
|  | procedureInstanceId | F3.3 of reference [1] | M |  | AV1 |  |
|  | initiatorIdentifier | F3.5 of reference [1] | M |  |  |  |
|  | responderPortIdentifier | F3.5 of reference [1] | M |  |  |  |
|  | serviceType | F3.5 of reference [1] | M |  |  |  |
|  | versionNumber | F3.5 of reference [1] | M |  |  |  |
|  | serviceInstanceIdentifier | F3.5 of reference [1] | M |  |  |  |
|  | bindInvocationExtension | F3.5 of reference [1] | M |  | ‘notUsed’ |  |

|  |  |
| --- | --- |
|  | For the BIND invocation the procedureRole element of the parameter bindInv-3 must be set to ‘associationControl’. |

The parameters bindInv-1, bindInv-2, and bindInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.5 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. This parameter is of the type StandardInvocationHeader that is specified in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)].

Table A‑ : BIND Return Parameters

| Parameters of the BindReturn PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Reference | Status | Support | Values | |
| Allowed | Supported |
|  | performerCredentials | F3.3 of reference [1] | M |  |  |  |
|  | invokeId | F3.3 of reference [1] | M |  |  |  |
|  | result | F3.3 of reference [1] | M |  |  |  |
|  | positive | F3.3 of reference [1] | C6 |  | ‘notUsed’ |  |
|  | diagnostics | F3.3 of reference [1] | C7 |  | AV2 |  |
|  | negExtension | F3.3 of reference [1] | C7 |  | ‘notUsed’ |  |
|  | responderIdentifier | F3.5 of reference [1] | M |  |  |  |

|  |  |
| --- | --- |
|  | IF bindRet-3 = ‘positive’ THEN M ELSE X |
|  | IF bindRet-3 = ‘negative’ THEN M ELSE X |

|  |  |
| --- | --- |
|  | For the negative BIND return the parameter bindRet-5 is extended by the type AssocBindDiagnosticExt defined in F3.5 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. Therefore the parameter bindRet-5 may have (a) any value defined for the Diagnostic type in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘diagnosticExtension’; or  (b) any value defined by ‘diagnosticExtension’: ‘acBindDiagExt’: ‘AssocBindDiagnosticExt’ defined in F4.5 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘assocBindDiagnosticExtExtension’. |

All parameters of the BIND return PDU except bindRet-7 are contained the complex parameter of the type StandardReturnHeader that is specified in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. Specific extensions are, however, specified in F3.5 of that document.

Table A‑ : PEER-ABORT Invocation Parameters

| Parameters of the PeerAbortInvocation PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Reference | Status | Support | Values | |
| Allowed | Supported |
|  | diagnostic | F3.5 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  | 40..126 |  |

Table A‑ : UNBIND Invocation Parameters

| Parameters of the UnbindInvocation PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Reference | Status | Support | Values | |
| Allowed | Supported |
|  | invokerCredentials | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | invokeId | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | procedureInstanceId | F3.3 of reference [[1]](#R_921x1b1CstsSpecificationFramework) | M |  | AV3 |  |
|  | unbindInvocationExtension | F3.5 of reference [[1]](#R_921x1b1CstsSpecificationFramework) | M |  | ‘notUsed’ |  |

|  |  |
| --- | --- |
|  | For the UNBIND invocation the procedureRole element of the parameter unbindInv-3 must be set to ‘associationControl’. |

The parameters unbindInv-1, unbindInv-2 and unbindInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.5 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. This parameter is of the type StandardInvocationHeader that is specified in F3.3 of that document.

Table A‑ : UNBIND Return Parameters

| Parameters of the UnbindReturn PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Reference | Status | Support | Values | |
| Allowed | Supported |
|  | performerCredentials | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | invokeId | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | result | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  | AV4 |  |

|  |  |
| --- | --- |
|  | The value of the parameter unbindRet-3 shall always be set to ‘positive’: ‘notUsed’; i.e., the result is always positive and not extended. |

All parameters of the UNBIND return PDU are contained the complex parameter of the type StandardReturnHeader that is specified in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)].

Table A‑12 : EXECUTE-DIRECTIVE Invocation Parameters

| Parameters of the ExecuteDirectiveInvocation PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Ref. | Status | Support | Values | |
| Allowed | Supported |
|  | invokerCredentials | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | invokeId | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | procedureName | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  | AV5 |  |
|  | directiveIdentifier | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  | AV6 |  |
|  | localProcDirQualifier | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C8 |  | AV7 |  |
|  | targetProcedureName | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | X |  |  |  |
|  | serviceProcDirQualifierValues | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | X |  |  |  |
|  | functionalResourceInstanceNumber | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C9 |  |  |  |
|  | functionalResourceQualifiers | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C9 |  | AV8 |  |
|  | directiveQualifierExtension | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | X |  |  |  |
|  | executeDirectiveInvocationExtension | F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  | ‘notUsed’ |  |

|  |  |
| --- | --- |
|  | IF execDirInv-4 is set to the Published Identifier of the ‘reset’ directive (pSCDPresetDirective as defined in Subsection F4.16 of reference [[1](#R_921x1b1CstsSpecificationFramework)]) THEN M ELSE X |
|  | IF execDirInv-4 is set to the Published Identifier of a directive that is registered under a procedure type that is associated with the type of service invoking the EXECUTE-DIRECTIVE operation, but different from the procedure type that shall perform the EXECUTE-DIRECTIVE operation THEN M ELSE X |
|  | IF execDirInv-4 is set to the Published Identifier of a directive that is registered under a Functional Resource Type THEN M ELSE X |

|  |  |
| --- | --- |
|  | If the procedureType element of the parameter execDirInv-3 has the value ‘sequenceControlledFrameDataProcessing’, the value of the procedureRole element of the parameter must be set to ‘prime procedure’; otherwise the the value of the procedureRole element of the parameter execDirInv-3 must be set to ‘secondary procedure’. |
|  | The Published Identifier specified in the execDirInv-4 parameter must identify a registered directive. |
|  | The parameter execDirInv-5 will be: ‘directiveQualifier’: ‘localProcDirQualifier’: ‘DirectiveQualifierValues’: ‘parameterlessValues’ : 'TypeAndValueComplexQualified': 'typeAndValue': 'TypeAndValue': 'intUnsigned': 'SEQUENCE OF IntUnsigned' where this SEQUENCE has the length 1 |
|  | The parameter execDirInv-9 will be one of the following: (a) ‘directiveQualifier’: ‘functResourceDirQualifier’: ‘functionalResourceQualifiers’: ‘DirectiveQualifierValues’: ‘sequenceOfParamIdsAndValues’, (b) ‘directiveQualifier’: ‘functResourceDirQualifier’: ‘functionalResourceQualifiers’: ‘DirectiveQualifierValues’: ‘parameterlessValues’, or (c) ‘directiveQualifier’: ‘functResourceDirQualifier’: ‘functionalResourceQualifiers’: ‘DirectiveQualifierValues’: ‘noQualifierValues’ |

The parameters execDirInv-1, execDirInv-2, and execDir-3 are contained in the complex parameter standardInvocationHeader in the ExecuteDirectiveInvocation type shown in Subsection F4.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. This parameter is of the type StandardInvocationHeader that is specified in Subsection F4.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)].

Table A‑13 : EXECUTE-DIRECTIVE Acknowledgement Parameters

| Parameters of the ExecuteDirectiveAcknowledge PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Ref. | Status | Support | Values | |
| Allowed | Supported |
|  | performerCredentials | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | invokeId | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | result | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | positive | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C11 |  | ‘notUsed’ |  |
|  | diagnostic | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C12 |  | AV9 |  |
|  | negExtension | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C12 |  | ‘notUsed’ |  |

|  |  |
| --- | --- |
|  | IF execDirAck-3 = ‘positive’ THEN M ELSE X |
|  | IF execDirAck-3 = ‘negative’ THEN M ELSE X |

|  |  |
| --- | --- |
|  | For the EXECUTE-DIRECTIVE acknowledgement the parameter execDirAck-5 is extended by the type ExecDirNegAckDiagnosticExt defined in F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. Therefore the parameter execDirAck-5 may have  (a) any value defined for the Diagnostic type in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘diagnosticExtension’; or  (b) any value defined by the extension ‘diagnosticExtension’: ‘execDirAckDiagExt’: ‘ExecDirNegAckDiagnosticExt’ in F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘execDirNegAckDiagnosticExtExtension’. |

All parameters of the EXECUTE-DIRECTIVE acknowledgement PDU are contained the complex parameter of the type StandardReturnHeader that is specified in subsection F4.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. Specific extensions are, however, specified in subsection F4.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)].

Table A‑14 : EXECUTE-DIRECTIVE Return Parameters

| Parameters of the ExecuteDirectiveReturn PDU | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Parameter | Ref. | Status | Support | Values | |
| Allowed | Supported |
|  | performerCredentials | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | invokeId | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | result | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | M |  |  |  |
|  | positive | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C13 |  | ‘notUsed’ |  |
|  | diagnostic | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C14 |  | AV10 |  |
|  | negExtension | F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] | C14 |  | ‘notUsed’ |  |

|  |  |
| --- | --- |
|  | IF execDirRet-3 = ‘positive’ THEN M ELSE X |
|  | IF execDirRet-3 = ‘negative’ THEN M ELSE X |

|  |  |
| --- | --- |
|  | For the negative EXECUTE-DIRECTIVE return PDU the parameter execDirRet-5 is extended by the type ExecDirNegReturnDiagnosticExt defined in F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. Therefore the parameter execDirRet-5 may have  (a) any standard value defined for the Diagnostic type in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘diagnosticExtension’; or  (b) any value defined by the extension ‘diagnosticExtension’: ‘execDirNegReturnDiagnosticExt’: ‘ExecDirNegReturnDiagnosticExt’ defined in F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘execDirNegReturnDiagnosticExtExtension’. Additional values can be introduced by the further extension ‘diagnosticExtension’: ‘execDirNegReturnDiagnosticExt’: ‘ExecDirNegReturnDiagnosticExt’: ‘execDirNegReturnDiagnosticExtExtension’.  If the EXECUTE-DIRECTIVE return PDU is used by the Master Throw Event procedure, that is, the procedureType element of the parameter execDirInv-3 of the associated EXECUTE-DIRECTIVE invocation has the value ‘masterThrowEvent’, additional values are introduced by the further extension ‘diagnosticExtension’: ‘execDirNegReturnDiagnosticExt’: ‘ExecDirNegReturnDiagnosticExt’: ‘execDirNegReturnDiagnosticExtExtension’: ‘teExecDirDiagExt’: ‘TeExecDirNegReturnDiagnosticExt’ (the type TeExecDirNegReturnDiagnosticExt is specified in F3.14 reference [[1](#R_921x1b1CstsSpecificationFramework)].). Therefore the parameter execDirRet-5 may have, in this case,  (a) any standard value defined for the Diagnostic type in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)]. except ‘diagnosticExtension’;  (b) any value defined by the extension ‘diagnosticExtension’: ‘execDirNegReturnDiagnosticExt’: ‘ExecDirNegReturnDiagnosticExt’ defined in F3.4 of reference [[1](#R_921x1b1CstsSpecificationFramework)] except ‘execDirNegReturnDiagnosticExtExtension’; or  (c) any value defined by the extension ‘diagnosticExtension’: ‘execDirNegReturnDiagnosticExt’: ‘ExecDirNegReturnDiagnosticExt’: ‘execDirNegReturnDiagnosticExtExtension’: ‘teExecDirDiagExt’: ‘TeExecDirNegReturnDiagnosticExt’ (defined in F3.14 [[1](#R_921x1b1CstsSpecificationFramework)]) except ‘teExecDirNegReturnDiagnosticExtExtension’. |

All parameters of the EXECUTE-DIRECTIVE return PDU are contained the complex parameter of the type StandardReturnHeader that is specified in F3.3 of reference [[1](#R_921x1b1CstsSpecificationFramework)] .. Specific extensions are, however, specified in F3.4 and F3.14 of reference [[1](#R_921x1b1CstsSpecificationFramework)].

1. Service Object Identifiers Module  
     
   (Normative)
   1. SerVICE CONTROL Object Identifiers module

CCSDS-FORWARD-FRAME-OBJECT-IDENTIFIERS

{ iso(1) identified-organization(3) standards-producing-organization(112) ccsds(4) css(4) csts(1) services(2) ServiceControl(4) ServiceControlServiceModules(4) object-identifiers(1)

}

DEFINITIONS

IMPLICIT TAGS

::= BEGIN

EXPORTS ;

IMPORTS services

, crossSupportFunctionalities

FROM CCSDS-CSTS-OBJECT-IDENTIFIERS

;

-- The CCSDS-CSTS-OBJECT-IDENTIFIERS and CCSDS-CSTS-COMMON-TYPES modules

-- are defined in F4.1 and F4.3 of reference [1].

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Root Object Identifiers of the Service

END

1. Service CONTROL Procedure Parameters, Events, and Directives  
     
   (Normative)

CSTS-FUNCTIONAL-RESOURCE-TYPES

{ iso(1) identified-organization(3) standards-producing-organization(112)

ccsds(4) css(4) crossSuppportResources(2)

}

DEFINITIONS

IMPLICIT TAGS

::= BEGIN

EXPORTS

scCstsProviderFrOid

;

scCstsProviderFrOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 }

-- This parameter reports the production status of the Service Control service instance.

-- This parameter can take on one of four values:

-- - 'configured': the SC service

-- is configured to access the directives of the controlled functional resources of

-- the Service Package;

-- - ‘operational’: the SC service is able to access the directives

-- of at least one of the controlled functional resources of the Service Package;

--

-- - ‘interrupted’: the ability of the SC service to access the directives of the controlled

-- functional resources of the Service Package has been interrupted;

-- - ‘halted’:

-- the ability of the SC service to access the directives of controlled functional

-- resources of the Service Package has been stopped by management action.

scProdStatParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 1 1 1 }

ScProdStat ::= ProdStat

-- This parameter configures and reports the identifier of the given service instance.

--

scServiceInstanceIdParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 2 1 1 }

ScServiceInstanceId ::= CstsSvcInstanceId

-- This enumerated parameter reports the status of the given instance of the Service

-- Control service. It can take on the following values:

-- - 'unbound': all resources

-- required to enable the provision of the service have been allocated, and all objects

-- required to provide the service have been instantiated. However, no association

-- yet exists between the user and the provider, i.e., the transfer service provider

-- port is not bound;

-- - 'boundReady': an association has been established between

-- the user and the provider, and they may interact by means of the service operations.

-- The user may activate the prime instance of the Throw Event procedure, which causes

-- the provider to transition to the state 'active';

-- - 'boundActive': this state resembles

-- state ‘boundReady’, except that in addition the user is able to invoke control directives

-- by initating the EXECUTE-DIRECTIVE operation of the Throw Event procedure. The service

-- instance remains in the 'boundActive' state until the user either invokes (a) the

-- STOP operation of the prime Throw Event procedure instance to cause the provider

-- to transition back to state 'boundReady' or (b) the PEER-ABORT invocation to cause

-- the service to transition back to the 'unbound' state.

scSvcInstanceStateParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 3 1 1 }

ScSvcInstanceState ::= StatefulCstsInstanceState

-- This parameter reports the identifier of the peer application, i.e., the authority

-- on whose behalf the CSTS application entity is initiating an association with the

-- Monitored Data service provider. The provider performs access control based on this

-- parameter. It may also serve as key to further security relevant information such

-- as the authentication level and method and the related password.

--

-- Given the security

-- relevance of this parameter, specific mechanisms for setting this parameter when

-- the FR instance is created and for the transfer of associated security relevant

-- data might be specified in a bilateral agreement between service user and service

-- provider.

scInitiatorIdParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 4 1 1 }

ScInitiatorId ::= AuthorityIdentifier

-- This parameter reports the identifier of the Service Control service application.

-- The user performs access conrol based on this parameter. It may also serve as key

-- to further security relevant information such as the authentication level and method

-- and the related password.

--

-- Given the security relevance of this parameter, specific

-- mechanisms for setting this parameter when the FR instance is created and for the

-- transfer of associated security relevant data might be specified in a bilateral

-- agreement between service user and service provider.

scResponderIdParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 5 1 1 }

ScResponderId ::= AuthorityIdentifier

-- This parameter configures and reports the port identifier to be used by the user

-- to connect to the service provider. However, the parameter value is only a logical

-- name that needs to be translated into the technology-specific addressing information

-- required to establish a connection with the specific port of the responding CSTS

-- application entity. As such this parameter is irrelevant for the service provider,

-- but it may be needed for certain kind of gateways between service user and service

-- provider application.

scResponderPortIdParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 6 1 1 }

ScResponderPortId ::= ResponderPortId

-- This parameter configures and reports the setting of the maximum time period in seconds

-- permitted from when a confirmed SC-CSTS operation is invoked until the return is

-- received by the invoker.

--

-- If a response is not received within that time period,

-- the invoker may invoke the PEER-ABORT operation.

--

scResponseTimeoutParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 7 1 1 }

ScResponseTimeout ::= SvcResponseTimeout

-- This parameter reports on the invoked directives. Invoked directives are identified

-- by the invocation ID (invoke-id as part of the standard operation header) and the

-- OID of the invoked directive. In addition the result of the directive invokation

-- is reported. For negative results a diagnostic is reported.

--

-- The covered execution

-- history is limited by the time this functional resource instance is instantiated.

scExecutedDirectivesParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 8 1 1 }

ScExecutedDirectives ::= SEQUENCE OF InvokedDirective

-- This parameter reports on the cancelled directives. Directives are identified by

-- the invocation ID (invoke-id as part of the standard operation header) and the OID

-- of the invoked directive. In addition the result of the directive invokation is

-- reported. For negative results a diagnostic is reported.

--

-- The covered execution

-- history is limited by the time this functional resource instance is instantiated.

scCancelledDirectivesParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 9 1 1 }

ScCancelledDirectives ::= SEQUENCE OF InvokedDirective

-- This parameter reports on the directives pending execution. Directives are identified

-- by the invocation ID (invoke-id as part of the standard operation header) and the

-- OID of the invoked directive. In addition the result of the directive invokation

-- is reported. For negative results a diagnostic is reported.

--

-- The covered execution

-- history is limited by the time this functional resource instance is instantiated.

scPendingDirectivesParamOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 1 10 1 1 }

ScPendingDirectives ::= SEQUENCE OF InvokedDirective

-- This directive allows to cancel directives which are pending for future execution.

-- Directives are identified by the invocation ID (invoke-id as part of the standard

-- operation header) of the EXECUTE-DIRECTIVE operation and the OID identifying the

-- directive.

scCancelPendingDirectiveDirectiveOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 3 1 1 }

-- This qualifier identifies the directive to cancel

scDirectiveIdentifierQualifierOid OBJECT IDENTIFIER ::= { 1 3 112 4 4 2 1 90200 3 1 1 1 }

ScDirectiveIdentifier ::= SEQUENCE OF SEQUENCE

{

invokeId INTEGER

, directiveOid OBJECT IDENTIFIER

}

END

1. Security, SANA, and Patent Considerations   
     
   (Informative)
   1. Security Considerations
      1. Introduction

This subsection describes security aspects of the Service Control service.

The CSTS Specification Framework Recommended Standard (reference [1]) explicitly provides authentication and access control for CSTSes. As one of the suite of CSTSes, the Monitored Data service inherits the authentication and access control capabilities defined in the CSTS Specification Framework Recommended Standard. The Service Control service provides no service-specific security capabilities. As specified in the CSTS Specification Framework, additional security capabilities, if required, are levied on the underlying communications services that support the SC-CSTS. Specification of the various underlying communications technologies, and in particular their associated security provisions, are outside the scope of this Recommended Standard.

* + 1. Security Concerns with Respect to the Service Control Service

The Statements of Security Concerns subsection (refer to H1 of reference [1]) identifies the support for capabilities that respond to security concerns in the areas of data privacy (also known as confidentiality), data integrity, authentication, access control, availability of resources, and auditing.

* + 1. Potential Threats and Attack Scenarios

As a member of the suite of CSTSes, the Service Control service depends on unspecified mechanisms operating in the underlying communications service, or on privacy-ensuring capabilities in the service-specific application processes that interoperate through the procedures defined in reference [1], to ensure data privacy (confidentiality). If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could read the data contained in the SC-CSTS protocol data units as they traverse the WAN between service user and service provider.

Reference [1] constrains the ability of a third party to seize control of an active CSTS instance, but it does not specify mechanisms that would prevent an attacker from intercepting the protocol data units. Interception of Service Control could assist an attacker in establishing the orbit/trajectory of the Mission spacecraft, which could assist the attacker in subsequent attempts to acquire return link data or jam the forward link. The prevention of such interception attacks depends on unspecified mechanisms in the underlying communications service. If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could intercept data transferred between the service user and the service provider without detection.

If the CSTS authentication capability is not used and if authentication is not ensured by the underlying communications service, attackers could somehow obtain valid initiator-identifier values and use them to initiate SC-CSTS instances by which they could gain access to the Service Control transferred via the service.

The SC-CSTS depends on unspecified mechanisms operating in the underlying communications service to ensure that the supporting network has sufficient resources to provide sufficient support to legitimate service users. If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could prevent legitimate service users from using the SC-CSTS.

If the service provider of the SC-CSTS provides no security auditing capabilities, or if a service user chooses not to employ auditing capabilities that do exist, then attackers may delay or escape detection while stealing data exchanged via the service.

* + 1. Consequences of not Applying Security to the Technology

The consequences of not applying security to the SC-CSTS are possible degradation and loss of ability to use the service, or the interception of Service Control that could aid in the determination of the orbit/trajectory of the spacecraft, acquisition of the space link, and/or jamming of the space link. The ability to actually acquire the space link (that is, either transmit to the spacecraft or receive and interpret data from the spacecraft) depends on the security supplied by the protocols used on that space link.

* 1. SANA Considerations

The SC-CSTS relies on the SANA Functional Resource Registry (reference [5]) to provide the identification and definition of Functional Resource parameters and events.

As described in this Recommended Standard, the SC-CSTS reports parameters and events that are named in the context of Functional Resources. Functional Resource Types are registered under the

{ iso(1) identified-organization(3) standards-producing-organization(112) ccsds(4) css(4) crossSupportResources(2)

}

node of the OID registration tree.

There are two subnodes under the crossSupportResources node: crossSupportFunctionalities and agencyFunctionalities, used to register CCSDS-standard Functional Resource Types and agency-unique Functional Resource Types, respectively. Agency-unique Functional Resource types are not relevant in the SC-CSTS context. Under each Functional Resource Type OID, the parameters, events, and directives are registered under dedicated subnodes.

Maintenance of the SANA registry of the Functional Resource Types, parameters, events, and directives under the crossSupportFunctionalities subnode is under the purview of the CCSDS Cross Support Services Area in accordance with the process and procedures identified in the CSTS Specification Framework (reference [1]).

The positions in the OID tree where the OIDs of the Functional Resources associated with the SC-CSTS are registered are specified in [Annex [B](#_Service_Object_Identifiers)] and Annex [C](#_Service_CONTROL_Procedure). The OID values are defined in Annex [C](#_Service_CONTROL_Procedure).

* 1. Patent Considerations

There are no patents that are known to apply to the technology used in the Service Control service.

1. Example Functional Resource Type   
   Object Identifier Registry  
     
   (INformative)

Introduction

This annex provides an example registry of Object Identifiers for the Functional Resource Types that are used in the operational scenario for the SC-CSTS (see reference [5](#R_FunctionalResourcesSpaceAssignedNumber)). This example set of Object Identifiers is a subset of the information found in the SANA Functional Resource Registry (reference [5]) as of the publication of this Recommended Standard.

These examples are included in this informative annex to provide example Object Identifiers for the Functional Resource Types, parameters, and events that are cited abstractly in the operational scenario.

The SANA Functional Resource Registry is the official repository of all object identification assignments for Functional Resource Types and the parameters and notifiable events that belong to those Types. The example Object Identifiers and parameter and event definitions used in this annex may at some time in the future diverge from the official definitions in the SANA registry. Any real implementation of the SC-CSTS shall always use only the Functional Resource Types, parameters, and notifiable events that are specified in the SANA registry.

The Functional Resource Types used in the operational scenario section of this Recommended Standard are:

* –  CCSDS 401 Space Link Carrier Transmission;

As specified in D3 of reference [[1](#R_921x1b1CstsSpecificationFramework)], all Functional Resource Types are registered under one of two subnodes under crossSupportResources node of the OID registration tree:

{ iso(1) identified-organization(3) standards-producing-organization(112)

ccsds(4) css(4) crossSuppportResources(2)

}

1. Informative references   
     
   (INformative)

[F1] *Cross Support Concept—Part 1: Space Link Extension Services*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 910.3-G-3. Washington, D.C.: CCSDS, March 2006.

[F2] *Space Link Extension—Internet Protocol for Transfer Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 913.1-B-2. Washington, D.C.: CCSDS, September 2015.

[F3] *Cross Support Transfer Service Specification Framework Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 920.0-G-1. Washington, D.C.: CCSDS, forthcoming.

[F4] *Space Link Extension—Return All Frames Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.1-B-4. Washington, D.C.: CCSDS, August 2016.

[F5] *Space Link Extension—Forward CLTU Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-4. Washington, D.C.: CCSDS, August 2016.

[F6] *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.

[F7] *Cross Support Service Management—Service Management Utilization Request Formats* CCSDS 902.9, proposed.

[F8] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 2. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 902.1-R-2. Washington, D.C.: CCSDS, May 2017.

[] *Navigation Data—Definitions and Conventions*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 500.0-G-3. Washington, D.C.: CCSDS, May 2010.

[F10] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1.Washington, D.C.: CCSDS, November 2013.

1. Acronyms  
      
   (INFormative)

|  |  |
| --- | --- |
|  |  |
| CCSDS | Consultative Committee for Space Data Systems |
| CLTU | Communication Link Transmission Unit |
| CM | Complex Management |
| CSTS | Cross Support Transfer Service |
| DOR | Differential One-way Ranging |
| F-CLTU | Forward CLTU |
| FR | Functional Resource |
| ISO | International Organization for Standardization |
| MDOS | Mission Data Operation System |
| OID | Object Identifier |
| PDU | Protocol Data Unit |
| RAF | Return All Frames |
| RF | Radio Frequency |
| SANA | Space Assigned Numbers Authority |
| SCCS-SM | Space Communication Cross Support Service Management |
| SM | Service Management |
| SLE | Space Link Extension |
| SC-CSTS | Service Control Cross Support Transfer Service |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

**ANNEX H**

CROSS REFERENCES to CSTS SPECIFICATION FRAMEWORK AND TO SERVICE CONTROL

**Table** H-1 lists the specific sections and subsections of the CSTS Specification Framework (reference [[1])](#R_921x1b1CstsSpecificationFramework) that are referenced by this Recommended Standard, and identifies the sections and subsections of this Recommended Standard that make specific reference to each of those sections/subsections of reference [[1](#R_921x1b1CstsSpecificationFramework)].

| **Reference [**[**1**](#R_921x1b1CstsSpecificationFramework)**] Section/Subsection** | **Referencing Sections/ Subsection**  **of SC-CSTS** |
| --- | --- |
| 1.6.1.3 | 1.5.2 f) |
| 2.2 | 1.5.2 f) |
| 4.3 | 3.2.5 |
| 4.12 | 3.2.5 |
| 4.3 | Table 3-1, A2.2 Table A-5 |
| 4.12 | Table-3.1, A-2.2 Table A-5 |
| Annex F | A1.2 |
| Annex F3.1 | Annex B |
| Annex F3.3 | Annex B, Annex C, Annex A2.2 Table (A-7,A-8,A-10,A11, A-12, A-13, A-14) |
| Annex F3.4 | Annex A2.2 Table (A-5, A-6, A-7, A-8,A-9, A-10,A-12) |
| Annex F3.5 | Annex A2.2 Table (A-5, A-6, A-7, A-8,A-9, A-10) |
| Annex F3.7 | Annex A2.2 Table (A-6, A-10) |
| Annex F3.14 | Annex B |
| Annex G3 | 3.3 |
| Annex H1 | Annex D1 |